

**O‘ZBEKISTON RESPUBLIKASI
OLIV VA O‘RTA MAXSUS TA‘LIM VAZIRLIGI
NIZOMIY NOMIDAGI
TOSHKENT DAVLAT PEDAGOGIKA UNIVERSITETI**

Tursunov S.Q., Fayziyeva M.R.

WEB-DIZAYN

**O‘quv qo‘llanma pedagogika oliy ta‘lim muassasalarining 5110700 – Informatika
o‘qitish metodikasi yo‘nalishi uchun mo‘ljallangan**

TOSHKENT – 2020

Bugun respublikamiz ta'lim tizimida Web texnologiyalardan keng foydalanilmoqda. Ushbu qo'llanma «Web-dizayn» deb nomlanib, unda Web dizayn elementlari, Dreamweaver, Flash dasturlari, shuningdek Web texnologiyalar, HTML, PHP, JavaScript dasturlash tillari va MySQL ma'lumotlar bazasini boshqarish tizimlari imkoniyatlari va ta'lim tizimidagi foydalanishning amaliy jihatlari xususida ma'lumotlar berilgan.

O'quv qo'llanma 5110700 – Informatika o'qitish metodikasi yo'nalishida o'qiladigan «Web-dizayn» o'quv fani mazmuni bo'yicha tuzilgan bo'lib, qo'llanmada talabalar mustaqil o'zlashtirishlari uchun topshiriqlar, ularning yechimlari va dasturiy kodlar berilgan. Qo'llanmadan pedagogika oliy ta'lim muassasalari talabalari va barcha qiziquvchilar foydalanishi mumkin.

O'quv qo'llanma Nizomiy nomidagi TDPU Ilmiy kengashining 2020 yil ___ / ___ dagi ___ - sonli qarori bilan nashrga tavsiya etilgan.

TAQRIZCHILAR:

texnika fanlari doktori, professor

R.H.AYUPOV

texnika fanlari nomzodi, dotsent

D.SH. ZIYADULLAYEV

MUNDARIJA

I BOB. WEB-DIZAYN VA WEB-SAHIFA YARATISH PRINSIPLARI		
1.1-§.	Web-texnologiyalarning asosiy tushunchalari	9
1.2-§.	Saytning texnik topshirig'ini ishlash chiqish va loyihalash bosqichlari	13
1.3-§.	Web-sahifaning tuzilishi va yaratish imkoniyatlari, asosiy tushunchalari	15
II BOB. HTML TILIDA WEB SAHIFA YARATISH		
2.1-§.	HTML tiliga kirish, uning asosiy elementlari. Teg tushunchasi.	18
2.2-§.	HTML tilida matnlar bilan ishlash	27
2.3-§.	HTML tilida jadvallar hosil qilish.	37
2.4-§.	HTML tilida rasmlar hosil qilish teglari	48
2.5-§.	HTML tilida gipermatn joylashtirish teglari.	51
2.6-§.	Freymlar hosil qilish va ularni qayta ishlash.	58
III BOB. DREAMWEAVER DASTURIDA WEB-SAYTLAR YARATISH		
3.1-§.	Dreamweaver dasturning asosiy oynalari, menyusi va komponentlari	64
3.2-§.	Dreamweaver dasturida web-sahifa yaratish	82
3.3-§.	Dreamweaver dasturida web-uzel yaratish	92
3.4-§.	Dreamweaver dasturida freymli tuzilma yaratish	100
3.5-§.	Dreamweaver dasturida elektron qo'llanma yaratish	104
3.6-§.	Dreamweaver dasturida sayt dizayni bilan ishlash	110
IV BOB. WEB-SAYTLAR YARATISHNING QO'SHIMCHA DASTURIY TA'MINOTI		
4.1-§.	Flash dasturida oddiy harakatli animatsiyalar va bannerlar yaratish	122
4.2-§.	Flash dasturida interfaol animatsiyalar yaratish	138
4.3-§.	Flash saytlar yaratish	144
4.4-§.	Java Scriptda dinamik saytlar hosil qilish	152
4.5-§.	PHP tili	201
4.6-§.	PHP tilida o`garuvchilar va konstantalar	206
4.7-§.	PHP tilining amallari	219
4.8-§.	PHP tilining operatorlari	226
4.9-§.	PHP da funktsiyalar	242
4.10-§.	PHP da massivlar	258
4.11-§.	PHP da fayllar bilan ishlash	273
	XULOSA	289
	IZOHLI LUG`AT	290
	Foydalanilgan adabiyotlar ro`yhati	298
	ILOVALAR	303

СОДЕРЖАНИЕ

ГЛАВА I. WEB-ДИЗАЙН И ПРИНЦИПЫ СОЗДАНИЯ WEB-СТРАНИЦ		
1.1-§.	Основные понятие Web-технологии	9
1.2-§.	Этапы разработки, технические задачи и проектирование сайта	13
1.3-§.	Структура и возможности создания Web-страниц, основные понятие	16
ГЛАВА II. СОЗДАНИЕ WEB-СТРАНИЦ НА ЯЗЫКЕ HTML		
2.1-§.	Введение в язык HTML, его основные понятия. Понятие тега.	18
2.2-§.	Работа с текстами на языке HTML	24
2.3-§.	Создание таблиц на языке HTML	31
2.4-§.	Теги вставки рисунков в языке HTML	36
2.5-§.	Теги создания гипертекста в языке HTML	37
2.6-§.	Создание фреймов и их настройка	41
ГЛАВА III. СОЗДАНИЯ WEB-САЙТОВ В ПРОГРАММЕ DREAMWEAVER		
3.1-§.	Основное окно, меню и компоненты программы Dreamweaver	64
3.2-§.	Создание web-страниц в программе Dreamweaver	82
3.3-§.	Создание web-узлов программы Dreamweaver	92
3.4-§.	Создания фреймовой структуры в программе Dreamweaver	100
3.5-§.	Создания электронного пособия в программе Dreamweaver	104
3.6-§.	Работа с дизайном сайта в программе Dreamweaver	110
ГЛАВА IV. ДОПОЛНИТЕЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СОЗДАНИЯ WEB-САЙТОВ		
4.1-§.	Создания простой анимации движения и баннеров в программе Flash	122
4.2-§.	Создание интерактивной анимации в программе Flash	138
4.3-§.	Создание Flash сайтов	144
4.4-§.	Создание динамических сайтов	152
4.5-§.	Язык PHP	201
4.6-§.	Переменные и константы в языке PHP	206
4.7-§.	Операции в языке PHP	219
4.8-§.	Операторы в языке PHP	226
4.9-§.	Функции в языке PHP	242
4.10-§.	Массивы в языке PHP	258
4.11-§.	Работа с файлами в языке PHP	273
	ЗАКЛЮЧЕНИЕ	289
	ГЛОССАРИЙ	290
	Литература	298
	ПРИЛОЖЕНИЕ	303

C O N T E N T

CHAPTER I. WEB-DESIGN AND PRINCIPIS OF CREATING WEB-PAGES		
1.1-§.	Basic concept of Web technology	9
1.2-§.	Stages of development, technical tasks and site design	13
1.3-§.	The structure and capabilities of creating Web pages, the basic concept	16
CHAPTER II. CREATING WEB PAGES IN HTML LANGUAGE		
2.1-§.	Introduction to HTML, its basic concepts. The concept of a tag.	64
2.2-§.	Using text in HTML	82
2.3-§.	Creating tables in HTML	92
2.4-§.	Tags for inserting drawings in HTML	100
2.5-§.	HTML hypertext creation tags	104
2.6-§.	Creating and Configuring Frames	110
CHAPTER III. CREATING WEB-SITES IN THE DREAMWEAVER PROGRAM		
3.1-§.	The main window, menus and components of Dreamweaver	46
3.2-§.	Creating web pages in Dreamweaver	48
3.3-§.	Creating a web-site for Dreamweaver	64
3.4-§.	Creating a frame structure in Dreamweaver	74
3.5-§.	Creating an electronic benefit in Dreamweaver	82
3.6-§.	Working with website design in Dreamweaver	103
CHAPTER IV. ADDITIONAL SOFTWARE OF CREATING WEB-SITES		
4.1-§.	Create simple motion animations and banners in Flash	122
4.2-§.	Creation of interactive animation in the program Flash	138
4.3-§.	Creating Flash sites	144
4.4-§.	Creating Dynamic Sites	152
4.5-§.	Language PHP	201
4.6-§.	Variables and constants in PHP	206
4.7-§.	Variables and constants in PHP	219
4.8-§.	Operations in PHP	226
4.9-§.	Operators in PHP	242
4.10-§.	Functions in PHP	258
4.11-§.	Arrays in PHP	273
	SUMMARY	289
	Glossary	290
	Literature	298
	ATTACHMENT	303

KIRISH

Jamiyatni axborotlashtirish inson hayotining barcha jabhalarida intellektual faoliyat turlarining ko'payishi, iqtisodning o'sishi, fan – texnika taraqqiyotining jadallashish uchun xizmat qiladi. Biz qaysi sohada faoliyat ko'rsatmaylik kundalik ish faoliyatimizga oid tasvirlar yaratish (rasmlar, chizmalar, grafiklar, diagrammalar va boshqa illyustratsiyalar), ularni qayta ishlash, zarur bo'lganda internet tarmog'iga joylashtirish kabi muammolarga duch kelamiz. Hammamizga ma'lumki yuqori sifatli turli tasvirlarni yaratish va qayta ishlash amallarini kompyuter grafikasida o'rganiladi. Bunday tasvirlarni sifatini yo'qotmasdan rang-barang ko'rinishda internetda namoyish qilish uchun esa Web-dizayn imkoniyatlaridan foydalaniladi. Chunki internetda namoyish qilish uchun web-saytlardan foydalaniladi.

Web-saytlarni yaratish va undan foydalanish ta'lim tizimida ham muhim ahamiyat kasb etadi. Ta'limda web-saytlardan nafaqat tijorat yoki namoyish vositasi sifatida, balki bilim berish maqsadlarida ham foydalaniladi. Albatta bunda ta'lim masofadan turib amalga oshiriladi. Bunday ta'lim berish jarayoni o'qituvchi va o'quvchi to'g'ridan-to'g'ri muloqot qilmasdan olib boriladi. Buning uchun ta'lim beruvchi (oliy o'quv yurti, maktab va boshqalar) o'qitiladigan predmetga tegishli ma'lumotlarni web-sahifalarga har xil ko'rinishlarda (matn, rasm, jadval, animatsiya, tovushli ma'lumot va xatto video tasma ko'rinishida) joylashi kerak. Talaba esa dunyoning xohlagan joyidan turib Internet orqali bu ma'lumotlardan bilim olishi mumkin. Ta'limning bunday ko'rinishi masofaviy ta'lim deb yuritiladi. Masofaviy ta'lim bugungi kunga kelib dunyoning ko'plab rivojlangan davlatlarida qo'llanilib kelinmoqda. Masofaviy ta'limdan foydalanish natijasida vaqt va pulni tejash, ortiqcha ovvoragarchiliklarsiz bilim olish imkoniyatlariga ega bo'linadi. Bunday imkoniyatlarni bizga web-sahifalar yaratib beradi. Ma'lum predmetga tegishli ma'lumotlarni (matn, rasm, jadval, animatsiya va boshqalarni) sahifalarga joylab, ularni doimiy ravishda to'ldirib, yangilab borish, o'quvchilarni bilimlarini baholash uchun nazorat test savollarini kiritish imkoniyatlari mavjud. Buning natijasida o'quvchining olgan bilimni nazorat qilish, baholash imkoniyatlari paydo bo'ladi. Bu esa web-texnologiyadan ta'limda foydalanishning eng optimal variantlaridan biri hisoblanadi.

Ushbu o'quv qo'llanmada ana shunday Web-saytlarni yaratishda qo'llaniladigan Dreamweaver va Flash dasturlari, HTML gipermatnlarni belgilash tillari, Java Script, PHP, kabi dasturlash tillaridan foydalanish imkoniyatlari haqida ma'lumot berilgan. Professional dasturlar qatoriga kiruvchi Dreamweaver dasturi o'zining barcha imkoniyatlari bilan har qanday murakkablik va masshtabdagi HTML sahifalarni yaratish va o'zgartirishga qodir. U vizual loyihalash (WYSIWYG) rejimini ta'minlab, dastlabki matnli WEB – hujjatlarning juda aniq ishlashi bilan ajralib turadi va katta tarmoqli loyihalarni qo'llab-quvvatlovchi ichki qurilgan vositalarga ega. Dreamweaverda matnlar hamda ob'yektlarning obrazlari ustida olib boriladigan ishlar bevosita kodlashtirishdan ustun turadigan gipermatnli hujjatlar yaratish usulini vizual deb atash qabul qilingan. Shu bilan birga dastur nafaqat vizual loyihalash vositalarining kuchli bazasiga ega, balki web – sahifalarni deyarli maxsus ko'rib chiqish dasturlari yordamida aks ettirishga ham qodir: Microsoft Internet Explorer yoki Netscape Navigator.

Dreamweaver web–dizayn sohasida gipermatnli hujjatlarni to‘g‘ridan-to‘g‘ri kodlashtirish yordamida yaratishni afzal ko‘ruvchi mutaxassislarga ham ma‘qul keladi. Dreamweaver qobig‘iga gipermatnlarni belgilovchi diskreptorlar bilan ishlash uchun zarur barcha uskunalarga ega to‘liq funksional HTML tahrirlagichi integratsiya qilingan. Dasturi «ochiq arxitektura» printsiplariga asoslangan. Bu amaliy dasturni interfeysi (Application Programming Interface, API) to‘liq ochiq ekanligini bildiradi. Uning yordamida tashqi dasturchilar hamda dasturiy ta‘minot bilan shug‘ullanuvchi firmalar, dastur va uning interfeysiga radikal funksional o‘zgartirishlar kiritishi: yangi uskuna qo‘shish, menyu palitrasi yoki bo‘limini yaratish, yangi ob‘yekt yoki multimedia roligini dasturlashtirish va hokazolar.

Dreamweaver tizimi tarkibiga multimedia roliklarini qo‘yishga mo‘ljallangan qo‘shimcha dastur kiradi, shuning uchun Flash – texnologiyaning interfaol imkoniyatlarini tahrirlagichdan chiqmay turib ko‘rib chiqish mumkin.

Ta‘lim tizimida esa, axborot texnologiyalarini qo‘llab, ta‘lim-tarbiya berish va unga tegishli barcha jarayonlarni axborotlashtirish bilan ta‘lim samaradorligini oshirishga olib kelinadi. Bunda - axborot texnologiyalaridan dars jarayonidagi ko‘rgazmalilik imkoniyatlari oshib o‘quvchilarning mazmuni tushunish darajasi ko‘payadi. O‘quvchi dars vaqtida biror murakkab jarayonni ko‘z oldiga keltirishi qiyin bo‘lib, uni tasavvur qila olmaydi. Ana shunday paytda kompyuterning maxsus dasturlari yordamida shu murakkab jarayonni aynan o‘zidek qilib tasvirlab berish mumkin. Bunda o‘quvchilarda ushbu murakkab jarayon haqidagi tasavvurni hosil qiladi. Bu kabi dasturlar ko‘pincha inson ko‘zi bilan ko‘ra olmaydigan (masalan, mashina dvigateli ichidagi harakatlar, odam qon aylanish tizimi va boshqalar) murakkab jarayonlarni tasvirlab berish uchun xizmat qiladi. Bunday dasturlarga misol qilib Flash, 3D MAX kabi dasturlarni aytishimiz mumkin. Ushbu dasturlar yordamida nafaqat murakkab jarayonlarning tasvirlarini, balki barcha turdagi elektron darsliklarni yaratish imkoniyatlari ham mavjud. Ayniqsa animatsiyali ma‘lumotlarni keltirish va bezashda keng qo‘llash mumkin.

Shuningdek o‘quv qo‘llanmada Flash dasturini o‘rganishga bag‘ishlangan ma‘lumotlar ham berilgan. Dastur imkoniyatlari, uning interfeysi, flash–filmlar, flash-saytlar yaratish va ulardan turli jarayonlarda foydalanish to‘g‘risidagi ma‘lumotlar berilgan.

I BOB. WEB-DIZAYN VA WEB-SAHIFA YARATISH PRINSIPLARI



1.1. Web-texnologiyalarning asosiy tushunchalari

Dizayn (ingliz tilidan *design* injener-konstruktor, lotinchadan *designare* o'lchab ko'rmoq, o'lchab bermoq) — bu ijodiy faoliyat bo'lib, uning asosiy maqsadi sanoat buyumlarini formal sifatini belgilaydi. Bu sifat shuningdek buyumning tashqi qirralarini o'z ichiga olish bilan birga, asosan shu tuzilmaviy va funktsional aloqadorlik xaridor hamda ishlab chiqaruvchi ko'zi bilan qaraganda buyumni yagona holatga keltiradi. Dizayn sanoat ishlab chiqarishi bilan bog'langan insonni o'rab turgan atrof-muhitning barcha jabhalarini qurshab oladi.

Dizaynning turlari – dizaynerlik faoliyati loyihalash predmetining farqlanishi, loyiha ishining maqsadlari va metodlari hamda uning yakuniy natijasi bilan farqlanadi. Dizaynning quyidagi turlari farqlanadi:

- grafik dizayn,
- sanoat dizayn,
- web-dizayn.

Grafik dizayn

Grafik dizaynda – information grafika, reklama va boshqalar bilan ishlanadi. Web-dizayn va grafik dizayn nomining o'xshashligiga qaramasdan, dizaynning turli yo'nalishlari hisoblanadi. Web-dizayn ma'lumotni saytda tasvirlash uchun ishlatiladi. Ammo grafik dizayn bilan web-dizayn o'zaro bog'langan. Grafik dizaynda tayyorlangan illyustratsiyalar saytlarni bezashda keng ishlatiladi. Grafik dizayn aslida foydalanuvchiga uzatiladigan ma'lumotni qabul qilinishini loyihalashi kerak. **Grafik dizaynning yo'nalishlari:**

1. Firma stili,
2. Shriftlar,
3. Plakat,
4. Grafik reklama,
5. Kitobiy grafika,
6. Gazeta grafikasi,
7. Jurnal grafikasi.

Sanoat dizayni – maishiy buyumlar va asbob-uskunalarni ishlab chiqishda, mashinasozlik ob'yektlarida, transport vositalarida va boshqalarda ishlatiladi.

Bulardan tashqari dizaynning bir qancha ko'rinishlari mavjud. Bular: **arxitektura dizayni, kiyim va aksessuarlar dizayni, art-dizayn, reklama dizayni.**

Web-dizayn (ingliz tilidan olingan bo'lib, web-design – web sahifani loyihalash ma'nosini bildiradi) – bu web-sahifani jihozlanishidir. Web dizayn sayt uchun huddi poligrafiya dizayni va qog'ozli nashrlar dastgohlari singari muhim rol o'ynaydi. Web-dizayn deganda nafaqat web-sayt uchun grafikli elementlarni yaratish, balki uning strukturasi loyihalash, unda harakatlanish vositalari, ya'ni butun saytni yaratish tushuniladi. Shu sababli ham biz ushbu o'quv qo'llanmada nafaqat web-muharrirlarni, animator dasturlarni ham yoritib o'tishni ma'qul ko'rdik.

Agar sayt juda ajoyib bezatilgan bo'lsin, ammo unda qulay va tushunarli harakatlanish chizmasi mavjud bo'lmasa, hammabop kerakli matnlar bo'lmasa, agar sayt yaxshi dasturchilar tomonidan tayyorlangan samarali dasturiy vositalarga ega bo'lmasa – bunday sayt tezda o'z foydalanuvchilarini yo'qotadi va undagi noqulaylik (hattoki u professional va munosib stilda tayyorlangan bo'lsa ham) muvofaqiyatsizlikka mahkum bo'ladi. Yaxshi web dizayn saytning sahifalarini hamohanglik bilan birlashtiradi. Boshqacha aytganda, web dizayn saytning joylashish o'rnini belgilaydi.

Foydalanuvchilarning ehtiyojlari, istaklari va cheklovlarini hisobga olgan holda saytni yoki web-ilovani loyihalash yaxshi samara beradi. Foydalanuvchilarning ehtiyojlarini hisobga olgan holda loyihalashtirish "User-Centered Design" deb nomlanadi va zamonaviy web-dizayn uchun asos hisoblanadi. Saytni mavjud muammolarni qanday hal qilishi yoki undan qanday foydalanishni yaxshiroq bilish uchun sayt dizayni ko'pincha foydalanuvchi izlanishlari, tajribasi jumladan intervyu va kuzatuvlar bilan boshlagan ma'qul. Dizaynerlar o'zlarining dizaynerlik ishlarining yaroqliligini ta'minlash uchun dizayn tayyorlash jarayonining har bir bosqichida foydalanuvchi tomonidan o'tkazilgan test sinovlarini o'tkazish odatiy holdir¹.

Web-dizaynni o'rganishda quyidagi uchta tushuncha muhim ahamiyat kasb etadi: **Web-sahifa**, **Web-sayt** va **Web-server**.

Web-sahifa – o'zining unikal adresiga ega bo'lgan va maxsus ko'rish dasturi yordamida (brauzer) ko'riluvchi hujjatdir. Unga matn, grafika, ovoz, video yoki animatsiya ma'lumotlar birlashmasi - multimediyali hujjatlar, boshqa hujjatlarga gipermurojaatlar kirishi mumkin.

Web-sayt – bir qancha web-sahifalarning mantiqiy birlashmasi. Inglizcha "site" (tarjimasi "joy") so'zining o'zbekcha talaffuzi. Umumjahon o'rgimchak to'ri ma'lum axborot topish mumkin bo'lgan va noyob URL bilan belgilangan virtual joy. Mazkur URL web-saytning bosh sahifasi manzilini ko'rsatadi. O'z navbatida, bosh sahifada web-saytning boshqa sahifalari yoki boshqa saytlarga murojaatlar bo'ladi. Web-sayt sahifalari HTML, ASP, PHP, JSP, grafik va boshqa fayllardan tashkil topgan bo'lishi mumkin. Web-saytni ochish uchun brauzer dasturidan foydalaniladi. Web-sayt shaxsiy, tijorat, axborot va boshqa ko'rinishlarda bo'lishi mumkin.

Web-server – tarmoqqa ulangan kompyuter yoki undagi dastur hisoblanib, umumiy resurslarni mijozga taqdim etish yoki ularni boshqarish vazifalarini bajaradi. Web-serverlar ma'lumotlar bazalari va multimediyali ma'lumotlarni bir biriga moslashtiradi. Web-serverda Web-sahifa va Web-saytlar saqlanadi. Web-server - Internet yoki Internetga ulangan umumfoydalanishdagi axborot serveri. Unda hujjatlar va fayllar – audio, video, grafik va matn fayllari – saqlanib, ular foydalanuvchilarga HTTP vositalar orqali taqdim etiladi. Web-server nomi u umumjahon tarmog'ining qismi bo'lgani uchun kelib chiqqan. Maxsus dasturiy ta'minotga ega bo'lgan, bir yoki bir necha web-sayt fayllarini saqlash va ularga ishlov berish mumkin. Bir necha web-sayt bitta kompyuterda ishlasa, Web-server deganda web-sayt ishlovchi virtual makon (dasturiy ta'minot va kompyuterdagi joy)

¹ Jennifer Niederst Robbins. Learning Web Design, Fifth Editions. A beginner's guide to HTML, CSS, JavaScript, and Web Graphics. O'Reilly Media. Canada. 2018. 808 pages.

tushuniladi. Shunga ko‘ra ko‘pchilik uchun “web-server” deganda “web-sayt” tushuniladi. Ko‘p axborotni saqlovchi web-saytlar bir paytning o‘zida bir necha kompyuterda saqlanishi va ularga ishlov berilishi mumkin. Web-server mijozlarning web-saytga so‘roviga javob beradi.

Bu tushunchalarga qo‘shimcha web hujjat tushunchasini ham ko‘rib o‘tamiz.

Odatda, maxsus HTML (Hypertext Markup Language) tilidagi hujjat. Web-hujjat jahon o‘rgimchak uyasining asosini tashkil qiladi. Ular gipermatndan iborat bo‘lib, foydalanuvchiga ajratib ko‘rsatilgan so‘z yoki jumlag qaratib, ma‘lumotlarni o‘qish, hujjatning boshqa qismiga yoki ayni hujjat bilan giperhavola yordamida bog‘langan boshqa web-hujjatga o‘tish imkonini beradi. Web-hujjat, shuningdek, matn, tasvir, tovush, videolarni mujassamlovchi gipermuhit ma‘lumotni ham o‘z ichiga olishi mumkin. Web-hujjatni ochish, ularni o‘qish yoki aks ettirish Internet brauzerlari yordamida amalgam oshiriladi. Web-hujjat tushunchasi “web-sahifalar” va “web-saytlar” tushunchalari bilan chambarchas bog‘liq. Odatda «web-sahifasi» atamasi web-hujjat atamasining sinonimini bildiradi, «web-sayti» atamasi esa yagona mavzu ostida birlashtirilgan yoki bitta tashkilot, muallif yoki foydalanuvchiga tegishli bo‘lgan sahifalar majmuasiga tegishlidir.

Munosib web-dizaynni yaratishda qatnashuvchi quyidagi elementlarni sanab o‘tish mumkin:

Shrift – web-sayt ichida imkoni boricha bitta shrift, yoki odatda ikki-uchta bir-biriga o‘xshash (oson qorishadigan) shriftlarni qo‘shib ishlatish mumkin bo‘lgan xususiyatlar va tanlangan fonda o‘qish mumkin bo‘lgan ranglarga ega bo‘lishi lozim.

Abzats – imkoni boricha web-saytning barcha sahifalarida matn va boshqa vizual materiallarni tekislashning (sozlashning) bir ko‘rinishlarida bo‘lishi maqsadga muvofiq.

Web-saytning rangli sxemasi – sahifaning oddiy matnini, havola va tashrif buyurilgan havolalarni taqdim qilishning uch xil ranglarini belgilaydi. Rangli sxema, yoki turli mavzudagi saytning bo‘limlari uchun bunday sxemalarning bir nechta ko‘rinishi saytning barcha sahifalarida takrorlanishi lozim.

Web-dizayn o‘zida belgilangan rangli va mantiqiy sxemani birlashtirgan qandaydir grafikli jihozlashni nazarda tutadiki, shuning uchun ish boshida saytni jihozlashning umumiy kontseptsiyasini o‘ylab ko‘rish kerak bo‘ladi. Barcha grafik elementlarni ikkita katta sinfga ajratish mumkin: chizilgan va biror real voqelik fotosuratlari. Saytni bezash ishlarida bu ikki tipni aralashtirib yubormaslik yoki ularni web-saytning tematik yoki mantiqiy qismlariga to‘g‘ri taqsimlash maqsadga muvofiq. Shuni alohida takidlab o‘tish kerakki, saytda har qanday tipdagi foto suratlarni illyustratsiya sifatida ishlatishdan oldin, ularni tegishli tarzda qayta ishlash (rang va tonda tuzatish, korrektsiya) zarur.

Web-dizaynning asosiy maqsadi – saytni egasiga foyda (moddiy, ma‘naviy) keltiradigan ko‘rinishda taqdim qilishdan iborat. Foyda sayt orqali mollarni to‘g‘ridan-to‘g‘ri sotish orqali bo‘lishi, shuningdek unga qo‘shimcha firmalarni, mahsulotlarni reklama orqali yangi mijozlarni o‘ziga tortish yo‘li bilan ham kelishi mumkin. Ta‘limda esa bularga qo‘shimcha o‘quv resurslarini yetkazib berish, ta‘limni boshqarish ishlarini amalga oshirish mumkin.

Bizga ma'lumki web-dizayner boshqa kasblarga nisbatan yosh (yangi) kasb hisoblanadi. Internetga bo'lgan talabning ortishi bilan, saytlarning dizayniga bo'lgan talab va web-dizaynerlar soni ham ortib bormoqda.

Endi web loyihalarning asosiy tiplari va ularni har birini vizual bezashga qo'yiladigan talablar bilan tanishib chiqamiz. Saytlar tarkibiga ko'ra quyidagi asosiy tiplarga bo'linadi:

Uy sahifasi (Домашняя страница) tushunchasining bir nechta ko'rinishi mavjud:

1. Alohida shaxsga tegishli web-sayt. Bu saytda shaxs haqidagi bir necha mavzudagi ma'lumotlar, matnlar, foto jamlanmalar berilishi mumkin. Odatda bunday saytlarga turli sohalarning (masalan, fan-texnika, madaniyat, sport va boshqa sohalarda) mashhur kishilari ega bo'ladi.

2. Brauzerda birinchi yuklanadigan web-sahifa. Saytdan foydalanish undan boshlanadi. Odatda, foydalanuvchi o'zining qaerda ekanligi va saytning boshqa sahifalarida nimalarni ko'rishi mumkinligi haqida uy sahifasidan ma'lumot oladi. Brauzer tomonidan dastur yuklangandan so'ng terminalda paydo bo'ladigan web-sahifaning, portalning, majmuaning birinchi sahifasi. Odatda, prezentatsiya va navigatsiya bo'yicha ham asosiy ish bajaradi.

Prezentatsion sayt – nomidan ko'rinib turibdiki, sayt firma/xizmatlar/mahsulotlar haqidagi ma'lumotlarni taqdim qiladi. Ma'lum sondagi sahifalarni o'zida birlashtirib, kamdan-kam holatda yangilanadi. Saytning asosiy vazifasi – taqdim qilinayotgan xizmatlar va uning afzalliklarini tashrif buyuruvchilargacha vizual jihozlash yordamida maksimal tezlik va samara bilan yetkazib bera olishdan iborat. Bunday loyihalar tartib qoidalar bo'yicha aniq, noyob dizayn, grafikli jihozlanish matnli ma'lumotlar oldida ustunlik qiladi. Kamdan-kam ishlatiladigan flash-animatsiyalar (yoki sayt to'laligicha Flashda tayyorlangan bo'lishi mumkin).

Misol sifatida <http://www.loyalstudios.com> va <http://www.designchapel.com> saytlarida keltirilgan sahifalarni aytish mumkin.

Korporativ saytlar – tashkilotning (firmalar, korporatsiyalar) internet tarmog'idagi vakolatxonasi hisoblanadi va binobarin, bunday loyihalarda eng asosiy talab ma'lumotni ya'ni grafika yoki matnning jihozlanishiga qaratiladi.

Informatsion resurslar - bu elektron kutubxonalar va gazetalar. Bunday tipdagi saytlarda ma'lumotlar bir sutkada bir necha marta yangilanadi. Ma'lumotlarning hajmi juda ham katta, va bunday saytlarni ishlab chiqishda shuni ta'kidlash kerakki, tashrif buyuruvchi saytni tomosha qilish uchun emas – ular mazmuni uchun kelishadi. Yangiliklar nashri (misol uchun - <http://www.cnews.ru>) yoki <http://sight.nmi.ru> singari rasmlar galereyasi, loyihaning bosh maqsadi – yuqori tezlikda ma'lumotni tashrif buyuruvchiga taqdim qilish yoki bu ma'lumotni qidirish imkoniyati. Har qanday grafikli bezaklar sahifaning hajmini oshirib yuboradi – ya'ni sahifaning yuklanish vaqtini oshirib yuboradi. Shuning uchun mumkin qadar bunday loyihalarda dizayn matnli ko'rinishda ishlab chiqiladi. Kartinkalar, fotografiyalar, grafiklar, sxemalar illyustratsiyalar hisoblanadi.

Elektron tijorat – axborot texnologiyalariga asoslangan biznes hisoblanadi. Elektron tijorat (ingliz tilidan e-commerce) - internetda o'zining sayti va virtual

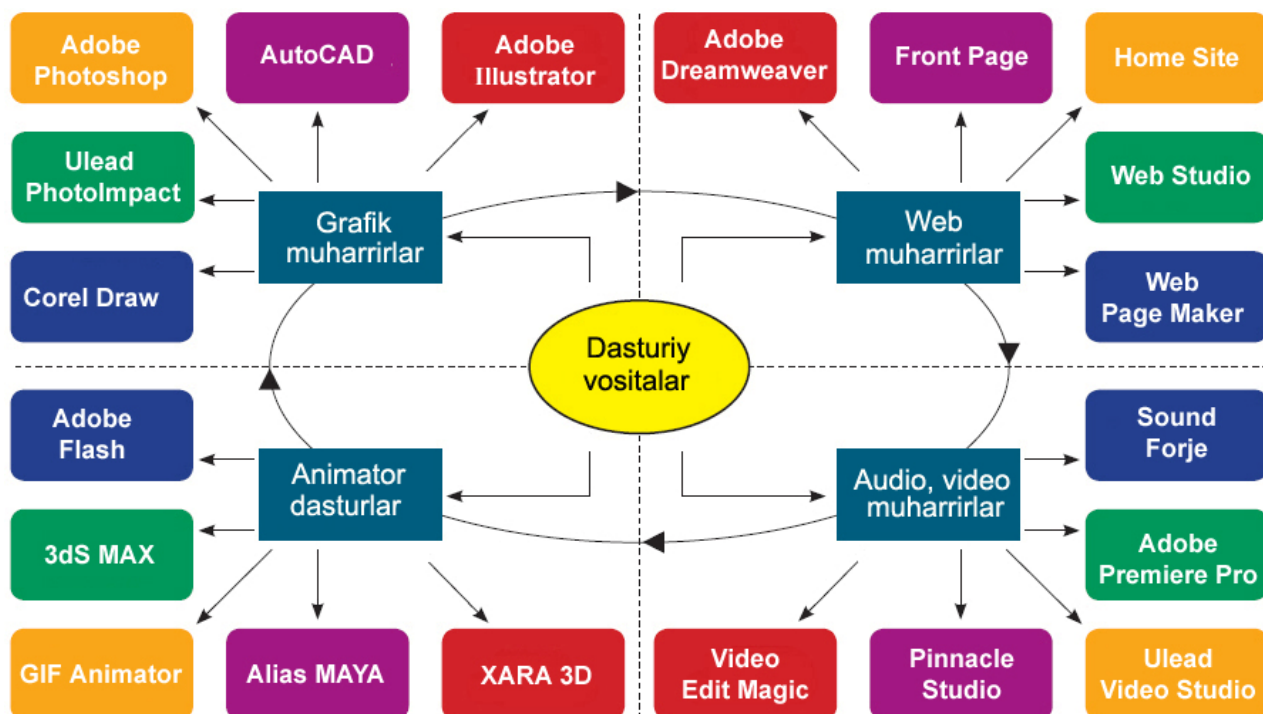
magazini, shuningdek o'zining (firma yoki kompaniyaning) boshqaruv tizimi mavjudligi bilan asoslanadi. Elektron tijoratga quyidagilar kiradi:

- elektron ma'lumot almashish (Electronic Data Interchange, EDI),
- elektron jamg'arma ko'chirish (Electronic Funds Transfer, EFS),
- elektron savdo (e-trade),
- elektron karmon (e-cash),
- elektron marketing (e-marketing),
- elektron banking (e-banking),
- elektron sug'urta xizmatlari (e-insurance).

1.2. Saytning texnik topshirig'ini ishlash chiqish va loyihalash bosqichlari

Bugungi kunning talablaridan kelib chiqib to'laqonli web-saytlar yaratishda bir qancha turdagi dasturiy vositalardan foydalaniladi. Chunki to'laqonli web-saytni shakllantirish uchun uning turli formatdagi informatsion ob'yektlariga turli muharrirlar yordamida ishlov beriladi. Masalan, saytga tegishli turli formatdagi tasvirlarni (rasmlarni) grafik muharrirlar yordamida yaratiladi va ularga ishlov beriladi.

Shu kabi to'laqonli web-saytlar yaratishga xizmat qiluvchi dasturiy vositalarni to'rtta katta guruhga ajratish mumkin (4.1-rasm) [11,12,13].



4.1-rasm. Web-saytlar yaratishda qo'llaniladigan dasturiy vositalar guruhlari.

Undan tashqari web-muharrirlarning ham turlari juda ko'p. Ular o'zining imkoniyati, tezkorligi, interfeysi va boshqa xususiyatlari bilan farqlanadi. 4.1-jadvalda shunday dasturlardan asosiylari keltirilgan.

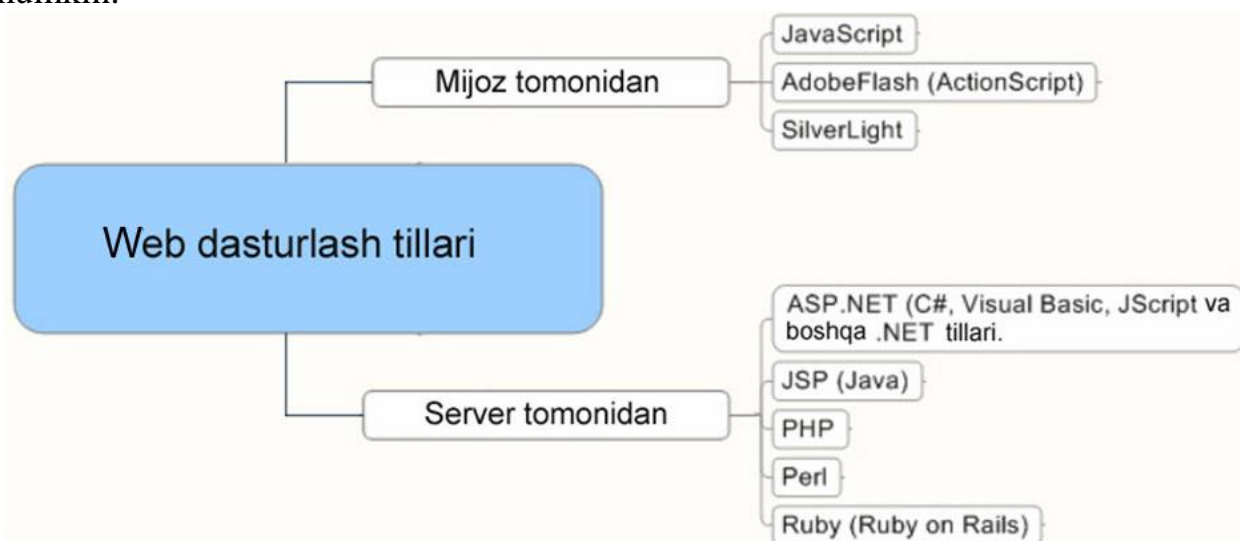
4.1-jadval. Web-muharrirlar ro'yxati

№	dastur nomi	izoh
1.	Bloknot (Notepad)	Windows operatsion tizimi standart vositalaridan bo'lib, undan HTML tilini yaxshi bilgan o'rganuvchilar foydalanadi.
2.	HomeSite 4.5	HTML-muharrir. Teglarni ranglar bilan ajratish, bir vaqtning o'zida bir nechta faylda qidirish va almashtirish va boshqalar.
3.	UltraEdit-32	Dasturchilar uchun keng imkoniyatli matn muharriri. Bir vaqtning o'zida bir nechta faylni tahrirlash imkoniyati mavjud.
4.	Web Page Maker	Dastur Software Inc mahsuloti hisoblanadi. Qisqa vaqt ichida web sahifa yaratish va yuklash imkoniyatini beradi. Unda HTML tilini bilish shart emas.
5.	DTM Lome 1.2	Ish boshlovchilar va professionallar uchun oddiy, tezkor va qulay HTML-muharrir.
6.	Artisteer.2.3.0.20943	Dastlabki web dizaynni avtomatlashtirgan dastur. Photoshop, CSS, HTMLlarni bilish talab qilinmaydi.
7.	CodeLobster PHP Edition 3.3	Juda qulay PHP, HTML, CSS muharriri
8.	Amaya	web-brauzer va HTML-muharrir The World Wide Web Consortium (W3C) – tomonidan ishlab chiqilgan.
9.	NeonHTML	O'zida vizual uskunalarni birlashtirgan HTML va CSS muharrir.
10	Serif WebPlus X2	Professional darajadagi web-saytlar yaratishga mo'ljallangan web-dizayn dasturi
11	PHP Designer 2008 Professional 6.2.5.2	PHP tilida ish boshlayotgan va professional foydalanuvchilar uchun dastur PHPdan tashqari, dasturda HTML, MySQL, XML, CSS, JavaScript, VBScript, JAVA, C, Python va Rubylarni ham ishlatish mumkin.
12	Dreamweaver CS4	Murakkab strukturali web-saytlarni loyihalash, yaratish va boshqarishga mo'ljallangan professional dastur.
13	Wap Editor 0.6d	Qulay va ko'p funktsiyali web sahifalar yaratishga mo'ljallangan dastur.
14	BestAddress HTML Editor 2008 Professional 12.2.0	Ko'p funktsiyali HTML-muharrir. Dasturda DHTML kodlari yordamida menyu yaratish imkoniyatlari mavjud.
15	DzSoft PHP Editor 4.2.1	PHP – skriptlar yozishga va HTML-sahifalar yaratishga mo'ljallangan dastur.
16	SaveChm 1.2	Rasmlı web sahifalarnı tezkorlık bilan CHM (Windows-help) formatida saqlıovchi dastur.
17	MenuEditor 1.0.3.231	Web-saytlarda harakatlanish uchun qulay menyularni yaratishga mo'ljallangan dastur
18	Quick Page 2008 (3.0.3.254)	Web-saytlar yaratishga mo'ljallangan professional dasturiy vosita.
19	AAA Web Album	Bu dasturiy vosita web sahifalar uchun foto albımlar yaratish uchun mo'ljallangan.
20	Multiple File Search and Replace 2.0	Bir vaqtning o'zida web saytnıng bir nechta fayllari bilan ishlashga mo'ljallangan dasturiy vosita
21	Incomedia Website Evolution X5 7.0.9	Foydalanish uchun qulay bo'lgan, tushunarli interfeys va qadamba-qadam yordamchidan foydalanib bir necha minutda web sayt yaratish mumkin
22	Arachnophilia 5.3.2177	Java asosida qurilgan HTML-muharrir Sun Microsystems mahsuloti hisoblanadi
23	Web Page Maker v2.5 Rus	Dasturning arxivida bir qancha tayyor sayt shablonlari mavjud bo'lib, bir necha daqiqada HTML sahifalar yaratish imkonini beradi
24	Incomedia WebSite X5 Evolution 7.0	Shaxsiy web saytlar yaratishga mo'ljallangan dastur
25	AceHTML Pro v6.03.0	Web-saytlar yaratish va ularga xizmat ko'rsatuvchi dasturlar paketi
26	Sothink DHTMLMenu 8.3 build 71210	Web-ustalar uchun mo'ljallangan dasturlar paketi
27	HtmlReader 2.5	html- sahifalarnı yaratish, tuzatish va ko'rishga mo'ljallangan dastur.

28	PHP Designer 2008 Professional v6.0.0	PHP, HTML, XHTML, CSS, Perl, C *, JavaScript, VB, java va SQL muharriri.
29	Atani 4.3.4 Rus	Web-sahifalar uchun bannerlar yaratishga mo'ljallangan dastur
30	Sothink DHTMLMenu	DHTML menyular yaratishga mo'ljallangan dastur. Dasturdan foydalanishda DHTML yoki JavaScript bilish talab qilinmaydi
31	TIGER CMS v1.0 Lite	Shaxsiy web-sahifalar yaratishga mo'ljallangan dastur

Web–dizaynda qo'llaniladigan bir qancha dasturiy vositalar mavjud. Bir qancha uskunaviy dasturlar, turli dasturlash tillari va boshqalar. Eng ko'p qo'llaniladigan tillarga misol qilib, HTML- gipermatnlarni belgilash tili, PHP, ASP, Java, JavaScript, DHTML va boshqa tillarni aytish mumkin.

Web-dasturlash tillari ikki guruhga bo'linadi: mijoz va server tillari. Biz yuqorida web-serverga ta'rif berdik. Unga ko'ra server sizning saytingiz saqlanadigan, brauzerning so'rovlari qayta ishlanadigan qaysidir sahifadagi dasturlar saqlanadi. Undan tashqari serverga quyidagicha ta'rif berish mumkin: Server – tarmoq ishini ta'minlovchi maxsus kompyuter. Server disklarida kompyuterlarni birgalikda ishlash imkonini beruvchi dasturlar, ma'lumotlar bazalari va boshqalar saqlanadi. Bundan tashqari serverlar modemli va faksli aloqalarni, ma'lumotlarni bosmaga chiqarish ishlarini amalga oshiradi. Serverda joylashgan dasturlardan foydalanish doirasi umumiy masalaning qo'yilishiga ko'ra cheklangan bo'lishi mumkin.

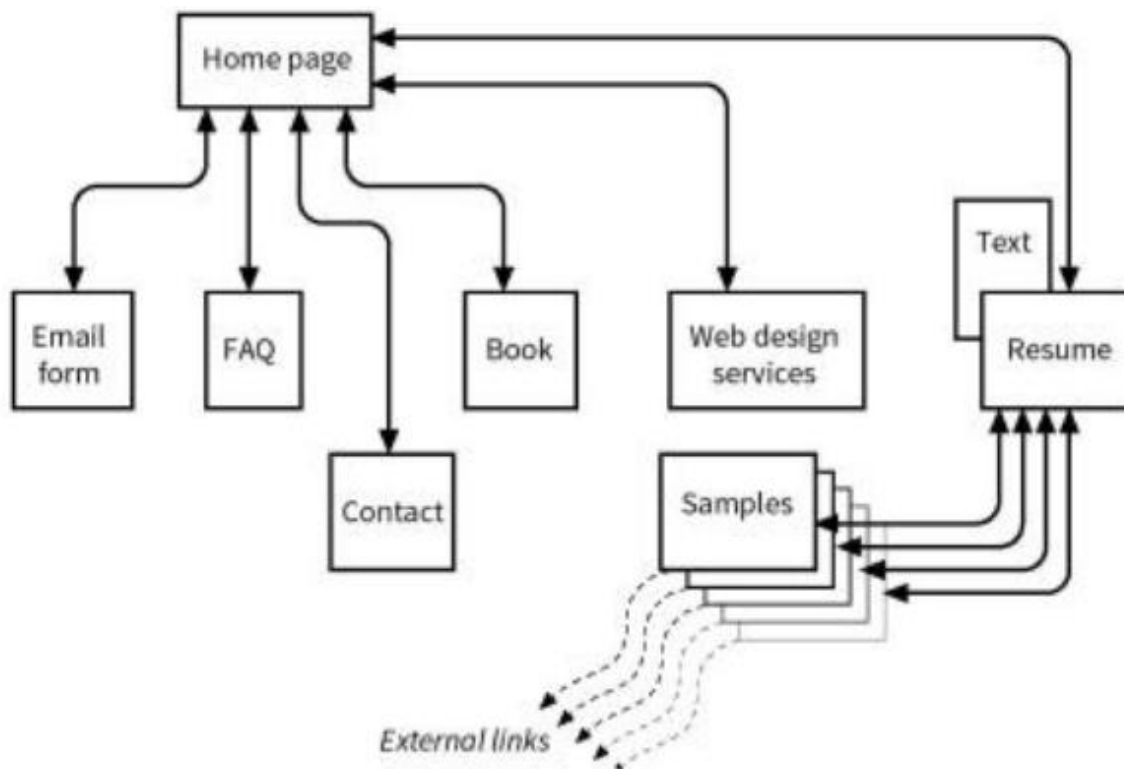


4.2-rasm. Web-dasturlash tillarining guruhlariga bo'linishi.

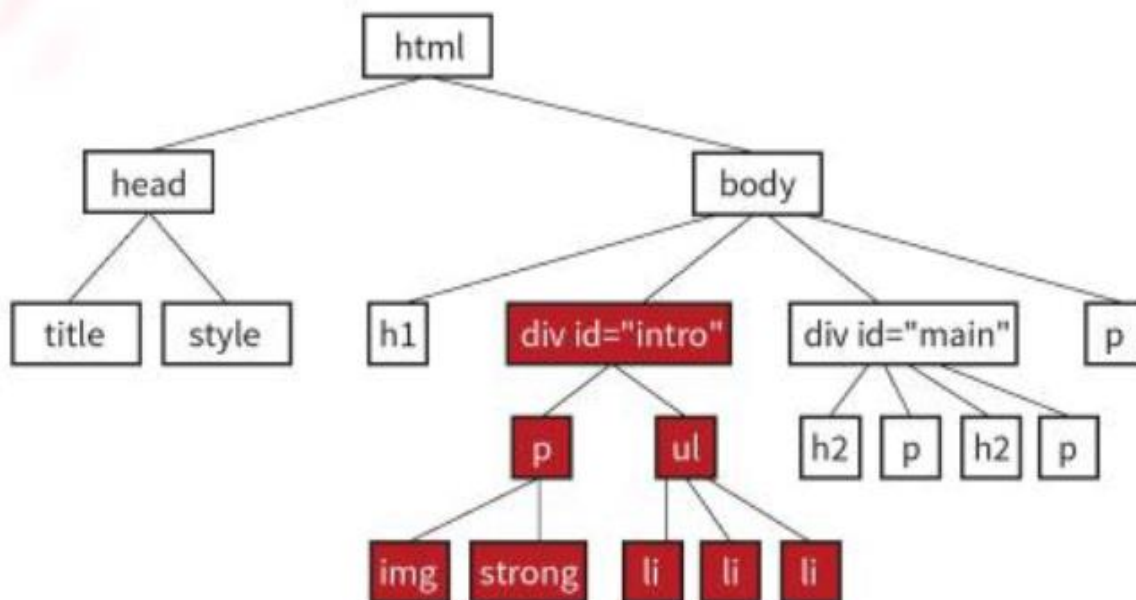
1.3. Web-sahifaning tuzilishi va yaratish imkoniyatlari, asosiy tushunchalari

Saytning tuzilmasi. Saytning diagrammasi umuman saytning tuzilishini va individual sahifalarning bir-biri bilan qanday bog'liqligini ko'rsatadi. 1.1-rasmda juda oddiy sayt diagrammasi ko'rsatilgan².

² Jennifer Niederst Robbins. Learning Web Design, Fifth Editions. A beginner's guide to HTML, CSS, JavaScript, and Web Graphics. O'Reilly Media. Canada. 2018. 808 pages.



Gap shundaki, web-dizaynerlar sifatida biz yaratgan sahifalarimizni qanday ko‘rishini aniq bilmaymiz. O‘nlab brauzerlardan qaysi biri ishlatilishini bilmaymiz, u stasionar kompyuterda, portativ qurilmalarda yoki boshqa biron-bir smartfonlar mos tarzda brauzer oynasi qanchalar katta bo‘lishi kerak, qandaydir shriftlar o‘rnatilganligi, JavaScript kabi funktsiyalar mavjudmi, Internet ulanishi qanchalik tezligi, sahifalarni ekran o‘quvchisi o‘qiy olishlarnin hisobga olish kerak³.



³ Jennifer Niederst Robbins. Learning Web Design, Fifth Editions. A beginner’s guide to HTML, CSS, JavaScript, and Web Graphics. O’Reilly Media. Canada. 2018. 808 pages.

JavaScript nima?

Zamonaviy web-saytlarning sahifasidagi JavaScript elementlari muhim to'ldiruvchi tarkibiy qismi hisoblanadi. Holbuki, HTML sahifasi tuzilmasini JavaScript yangi imkoniyatlar bilan ta'minlaydi va CSS stillar esa matnlar, jadvallar va boshqa elementlar ko'rinishini o'zgartiradi. JavaScript sahiga turli amallarni boshqaradigan xatti-harakatlar komponentini qo'shadi. Skriptlar serverda mustaqil qismlar sifatida alohida fayllarda saqlanishi mumkin⁴.

Mijoz tillaridan eng ko'p tarqalgani JavaScript tili bo'lib, Netscape (Netscape Navigator, www.netscape.com) SunMicrosystems (www.sun.com) kompaniyalari hamkorligida ishlab chiqilgan. **JavaScript** faol HTML-sahifalar uchun senariylar yozishga mo'ljallangan. JavaScript avtonom ravishda ishlaydigan ilovalarni yaratishga mo'ljallanmagan. JavaScriptdagi dasturlar bevosita HTML-hujjat kodi ichiga quriladi va brauzerda sahifani yuklanishi vaqtida faollashadi. Sodda qilib aytganda HTMLning cheklangan, sodda imkoniyatlarini kengaytirish va to'laqonli sahifalar yaratishga mo'ljallangan. JavaScript yordamida odatda ma'lumotli va muloqot oynalarini chiqarish, animatsiyalarni ko'rsatish kabi vazifalarni bajarish mumkin. Bundan tashqari, JavaScript-senariy ba'zan o'zi ishlab turgan brauzer va platforma tipini aniqlash mumkin. JavaScript-senariylar foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni to'g'riligini tekshirishda ham qulay hisoblanadi.

ActionScript nima?

ActionScript – ob'jektga yo'naltirilgan dasturlash tili bo'lib, Flash-ilovalar tarkibida interfaollik, ma'lumotlarni qayta ishlash va boshqa ko'plab imkoniyatlarni beradigan ECMAScript dialektlaridan biri hisoblanadi. ActionScript imkoniyatlari va unga oid sodda misollar bilan VI bobda batafsil tanishib chiqamiz.

Silverlight nima?

Microsoft Silverlight – bu dasturiy platforma bo'lib, RIA (Rich Internet application)ga xos bo'lgan animatsiyalar, vektorli grafika va audio-video roliklardan tarkib topgan ilovalarni yuklash imkonini beradigan brauzer uchun pliginalarni o'z ichiga oladi. Bu esa Silverlight barcha brauzerlarda WMV, WMA va MP3 kabi formatlarni Windows Media Player singari qo'shimcha komponentlarsiz ijro qilish imkoniyatini beradi. Dasturning 2.0 versiyasi 2008 yil oktyabrda ishlab chiqilgan bo'lib, unga .NET tillari bilan ishlash imkoniyati qo'shilgan. Bundan tashqari Microsoft kompaniyasi 2009 yil iyulda IDE bilan integratsiyalashgan Silverlight 3 va Microsoft Expression Studio 3 dasturlarini ishlab chiqdi.

Silverlight platformasi Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows 7, Mac OS X 10.4, Mac OS X 10.5, Mac OS X 10.6 operatsion tizimlariga va Internet Explorer 6.0/7.0/8.0, Mozilla Firefox 1.5/2.0/3, Safari 3.1, Google Chrome 3.0, Opera 9.50 brauzerlariga mo'ljallangan edi.

⁴ Jennifer Niederst Robbins. Learning Web Design, Fifth Editions. A beginner's guide to HTML, CSS, JavaScript, and Web Graphics. O'Reilly Media. Canada. 2018. 808 pages.

VBScript nima?

VBScript tili Microsoft korporatsiyasi tomonidan yaratilgan bo‘lib, Visual Basic tilining bir qismi hisoblanadi. VBScript tili Internet Explorer va Microsoft Internet Information Server (IIS)lar bilan ishlashga mo‘ljallangan tildir. VBScript tilining JavaScript tili bilan umumiy qismlari bir nechta. Jumladan, u aynan Microsoft Internet Explorer bilan ishlash va uning qo‘llanish sohasini cheklay olish imkoniyatiga ega. VBScript interpretatorli (sharhlovchi, izohlovchi, tushuntirib beruvchi) til hisoblanib, Microsoftning Web-texnologiyalari (masalan ASP (Active Server Page)) bilan hamkorlikda ishlay oladi. Shunga qaramay VBScript klient tomonida ishlovchi senariy hisoblanadi, ASP esa server tomonida ishlaydi.

Dasturning ishlashi to‘liq sayt joylashtirilgan serverga bog‘liq. Server tillarini ishlashining muhim jihatlaridan biri bu serverda tartiblangan holda saqlanuvchi va xohlagan paytda chaqirilishi mumkin bo‘lgan ma‘lumotlarni bevosita ma‘lumotlar bazalarini boshqarish tizimlari bilan o‘zaro bog‘langan holda faoliyat yuritadi. Eng ko‘p ishlatiladigan ma‘lumotlar bazasini boshqarish tizimlari quyidagilar: Firebird, IBM DB2, IBM DB2 Express-C, Microsoft SQL Server, Microsoft SQL Server Express, mSQL, MySQL, Oracle, PostgreSQL SQLite, Sybase Adaptive Server Enterprise.

Server tomonida bajarilishi kerak bo‘lgan senariylar odatda sayt papkasining ichidagi maxsus papkaga joylashtiriladi. Foydalanuvchi so‘roviga asosan server bu senariyni bajaradi. Bajarilgan senariy natijasi web-serverga uzatiladi va undan so‘ng klientga uzatiladi. Server tomonidagi senariylarni tashkil etish uchun odatda Perl, ASP, PHP, JSP, SSI kabi tillar va texnologiyalardan foydalaniladi.

Perl nima?

Perl – bu protsedurali dasturlash tili bo‘lib, juda keng qamrovli topshiriqlarni echish uchun mo‘ljallangan universal uskunadir. Perl boshqa to‘laqonli tillar singari platformalarga, operatsion tizim va boshqa vositalarga bog‘liq bo‘lmagan funksiyalar va qoidalar to‘plamiga ega yadrodan tashkil topgan. Perl tili Web-ilovalar yaratishda eng ommabop tillardan biri hisoblanadi. Matnlarni qidirish va tahrirlash, fayllar bilan qulay ishlay olish qoidalari uchun Perl tili Internetning asosiy tillaridan biri bo‘lib qoldi. Perl – interpretatorli til hisoblanadi, shu bois unda yaratilgan senariylar ishlashi uchun server kompyuterda Perl-interpretator o‘rnatilgan bo‘lishi kerak. Bevosita Perl-kodning interpretatsiya qilinish jarayoni uning samaradorligini pasaytiradi. Bugungi kunda Perlning asosiy yutuqlaridan, uning barcha platformalar uchun ishlay olishi va uning barcha resurslari bepul tarqatilayotganligidir. Ko‘pgina web-serverlar UNIXda ishlaydi, Perl interpretator esa bu operatsion tizimning bir qismi hisoblanadi.

ASP nima?

ASP (Active Server Pages)

ASP - ma‘lumotlar bazalari tashkil etish va ular bilan ishlash vazifalarini bajarishda juda moslashuvchan, qulay vositadir. ASP vositalari server tomonida ishlaydi va HTML-kod va senariylar kabi fayllarni qayta ishlaydi. ASP texnologiyasi VBScript, Java va JavaScript tillarini qo‘llab quvvatlaydi. ASP-kod joylashtirilgan

Web-sahifalar fayllari kengaytmasi .asp bo‘ladi. ASP texnologiya Windows NT va Microsoft IIS web-serveriga mo‘ljallangan bo‘lib, imkoniyatlari va samaradorligi yuqori bo‘lganligi bois ko‘pgina kompaniyalar o‘z vositalariga ASPni qo‘llab quvvatlash imkoniyatlarini kiritmoqdalar. Ko‘pgina HTML- muharrirlar, masalan, Adobe GoLive ham ASPni qo‘llab quvvatlaydi. ASP texnologiyasi bir nechta qulayliklarni o‘zida jamlagan: HTML-hujjatni dinamik generatsiyalaydi, formalarni qo‘llab quvvatlaydi, ma’lumotlar bazasiga ruxsatni tashkil etadi va u bilan ishlay oladi. ASP – dasturlash tili ham, ilova ham emas, u interfaol web-sahifa hosil qilish texnologiyasi.

JSP

JSP (JavaServerPage) texnologiyasi o‘zining funktsional imkoniyatlariga ko‘ra ASPga o‘xshashdir. Asosiy farqi shundaki, bunda VBScript va JavaScript bilan birga Java tili ham qo‘llanila oladi. Shunga qaramay JSP Javadan oldinroq qo‘llanilgan va ushbu texnologiya mukammal Web-ilovalar yaratish uchun yetarli imkoniyatga ega.

SSI

SSI (Server Side Include) HTML-faylni dastlab serverda qayta ishlaydi va undan so‘ng uni mijozga uzatadi. Dastlabki qayta ishlash vaqtida xujjatga dinamik generatsiya qilingan ma’lumotlar qo‘shiladi, masalan joriy vaqt haqidagi ma’lumot. Umuman olganda SSI texnologiyasi HTML-faylning tarkibiga qo‘shimchalar qo‘shishga mo‘ljallangan HTMLning qismi hisoblanadi.

CSS nima?

Hujjatga CSS stillarini qo‘shganda, siz web-brauzerga ma’lum elementlar ko‘rinishini xohlashinzi bildirasiz. Har bir stil brauzer uchun formatlashning bir turi hisoblanadi. Siz ushbu yo‘riqnomalarni turli xil shakllarda ishlatishingiz mumkin, ular quyidagi misollarga o‘xshashdir⁵:

- barcha havolalarni qizil matnda aks ettiring;
- barcha sarlavhalar uchun ma'lum bir shriftdan foydalaning;
- ushbu paragraf atrofida biroz bo‘sh joy yarating;
- ushbu rasmga soya qo‘shing;
- barcha raqamlangan ro‘yxatlar uchun kichik harflar bilan Rim raqamlaridan foydalaning;
- har doim matnning ushbu qismini oynaning o‘ng tomonida ko‘rsatib turing;
- biror rasmni 45 darajaga aylantiring.

DHTML nima ?

DHTML (dinamik HTML) – bu serverga ortiqcha bog‘lanishlarsiz interfaol Web-sahifalar yaratish imkonini beradigan vositalar to‘plamidir.

DHTML tili HTML va JavaScriptlardan farqli ravishda bir qancha qo‘shimcha internet texnologiyalariga ega. Dinamik HTML - Netscape va Microsoft tomonidan o‘ylab topilgan tijorat termini bo‘lib, unda Web-brauzerlarning dinamik imkoniyatni

⁵ Paul McFedries. Web Design Playground: HTML & CSS the Interactive Way 1st Edition., Manning Publications; 1st edition, 2019. 440 pages.

kengaytirish maqsadida kiritilgan. Bu texnologiyalar HTML yordamida Web-sahifa yaratishdagi mavjud imkoniyatning cheklanganligi va ana shu cheklovdan o'tib ketish uchun yaratilgan yoki qo'shilgan hisoblanadi. Chunki bizga ma'lumki HTMLning faqatgina o'zidan foydalanib bugungi kunda mukammal Web-sahifalar yaratib bo'lmaydi. Boshqacha aytganda mavjud Internetda matnli ma'lumotlar va grafik ma'lumotlarga mo'ljallangan juda yaxshi texnologiyalar mavjud edi. Ammo multimedia imkoniyatlariga ko'nikib qolgan odamlar esa bunga ko'nika olmadi. Shu sababli Internet imkoniyatlarini kengaytirish kerak edi.

DHTML asosida yaratilgan sahifa serverdagi qo'shimcha ma'lumotlarga murojaat qilmasdan turib ham o'zgarish imkoniyatiga ega. DHTML o'zida turli xildagi standartlarga (CSS, DOM, JavaScript va gipermatnlarni belgilash tillariga) asoslangan kombinatsiyalarni namoyon qiladi. Bu standartlarning kombinatsiyasi sizning web-sahifangizni interfaoligini oshirish imkonini beradi. Boshqacha aytganda, sahifaga tashrif buyurgan foydalanuvchi uning tashqi ko'rinishi va mazmunini serverga murojaatlarsiz o'zgartira olish imkoniyatidir.



Savol va topshiriqlar:

1. Web-dizayn nima? Web-sayt nima? Web-server nima? Web-sahifa nima?
2. Web-resurslarni yaratuvchi qanday dasturiy vositalarni bilasiz?
 1. Mijoz tomonidan senariylar nima?
 2. Server tomonidan senariylar nima?
3. VBScript nima?
4. Silverlight nima?
5. ActionScript nima?
6. Java Script nima?
7. DHTML nima?
8. Perl nima?
9. SSI nima ?
10. JSP nima ?
11. ASP nima ?

II BOB. HTML TILIDA WEB SAHIFA YARATISH



2.1. HTML tiliga kirish, uning asosiy elementlari. Teg tushunchasi.

Bizga ma'lumki, har qanday murakkablikdagi web-saytlar HTML tiliga asoslangan bo'lib, bu tilni bilmasdan turib to'laqonli web-saytlar yaratib bo'lmaydi. Endi HTML tilining asoslari bilan tanishib chiqamiz.

HTML (Hyper Text Markup Language) - giper matnlarni belgilash tili degan ma'noni anglatadi. HTML web-sahifalar uchun dastur yozishning standart tili hisoblanadi. HTML, World Wide Web konsortsiumi tomonidan belgilangan (aniqlangan) va ko'p sonli o'zgarishlardan so'ng bugungi ko'rinishi ishlab chiqilgan. Ayni paytda uning beshinchi HTML5 qayta ishlangan versiyasi keng qo'llanilmoqda. HTML, matni, shrift xususiyatlarini, abzatsni formatlash, sahifa maketi, rasmni joylashtirish, giperhavolalar va boshqa sahifada uchraydigan ob'yektlarni brauzer dasturiga qanday aks ettirishni belgilab beradi. HTML teglarni qo'llaydi. Bu teglar brauzerga matn yoki grafikani qanday shakllantirishni yetkazib beruvchi vosita hisoblanadi. Masalan, sahifadagi matnlarni sarlavha sifatida, ro'yxat sifatida yoki oddiy matn sifatida. Web-sayt yaratuvchilari sahifani HTML-teglari yordamida belgilash yo'li bilan yaratadilar. HTML – web-sahifada matn, tasvir va boshqa ma'lumotlarni qanday ko'rinishda joylashtirilishini belgilovchi vosita⁶.

Bu til 1989 yilda Jenevada Tim Berners-Li (Tim Berners-Lee) tomonidan yaratilgan. Keyinchalik Tim Berners-Li «Internet otasi» nomiga sazovor bo'lgan. Til dasturlash tili emas, faqat Web-sahifa yaratish uchun mo'ljallangan. Web-sahifada biror ma'lumotlarni qayta ishlash algoritmlarini o'rnatish uchun Java tilida tuzilgan dasturlardan foydalaniladi.

Ushbu ma'lumotlardan foydalanib, ko'zda tutilgan asosiy maqsad - HTML haqida asosiy tushunchalarni misollar yordamida tushuntirib berish, HTMLning bir-oz qiyin bo'lgan elementlari hisoblanmish jadval tuzish, freymlardan foydalanish va sahifaga tasvirlarni joylashtirish, sahifalarni bir-biriga faol havolalar orqali bog'lash usullarini ko'rsatib berishdan iborat.

Teg tushunchasi. HTML sahifa – bu oddiy matn fayl bo'lib, HTML kengaytmasiga ega. HTML sahifani yaratish uchun maxsus dastur shart emas. Har qanday matn muharriri yordamida HTML sahifa yaratish mumkin. Ana shunday matn tahrirlovchi oddiy dasturlardan biri bo'lgan Notepad (Bloknot) standart holatda Windows mo'hitida mavjud. HTML sahifaning asosiy elementlaridan biri bo'lgan Tegar haqida dastlabki tushunchalar bilan tanishib chiqamiz.

HTML teg:

Bu HTML hujjatdagi turli elementlarni (matn, tasvir va boshqalar) web-sahifada qanday ko'rinishda aks etishi lozimligini brauzerda ko'rsatuvchi buyruqlar. Tegar ma'lum o'lcham va vazifani bajaruvchi buyruqlardir.

Teg quyidagi ko'rinishda bo'ladi:

<TEG> hujjat elementi </TEG>

⁶ Ralph M. Stair, George W.Reynolds. Fundamentals of Information Systems. Sixth edition. Course Technology, Cengage Learning. USA. 2012. 182-p.

Ko'rib turganingizdek teglar < > uch burchaksimon qavs ichiga olib yoziladi. Uch burchaksimon qo'shtirnoq ichiga Teg nomi va uning o'lchamlari yoziladi.

Quyida teg va uning o'lchamlari yozilishiga misol keltirilgan:

 Hello World!

Misolda keltirilgan teglar HTML hujjatga kiritilgan **Hello World!** jumla qizil rangda, ajratilib yoziladi. Teglar orasida kiritilgan matn, teg o'lchamlarida qanday buyruqlar berilgan bo'lsa, shu buyruqlarga bo'ysungan holda ekranda hosil bo'ladi. Bu misolda Hello World! jumlasiga nisbatan ikkita teg ishlatilgan. 1 - teg va 2 - teg Hello World! jumlasidan keyin yana ikki teg va yozilgan. Bularning farqi shundaki, birinchi ikkita teg ochuvchi va jumladan so'ng yozilgan ikkita teg esa jumla oldida yozilgan teglarni davomi ya'ni yopuvchi tegdir. Yopuvchi teglarning vazifasi ochilgan teg ta'sir maydoni tugaganligini belgilashdan iborat. HTML hujjatda deyarli barcha teglarni yopish lozim. '/' slesh belgisini qo'yish orqali yopuvchi teg yasaladi.

Eslatma: har doim eng oxirgi ochilgan tegni birinchi navbatda yopishni unutmang. Teglarining ochilishi va yopilish tartibini almashtirib yubormang. Aks holda xatoliklar kelib chiqishi mumkin.

Endi e'tiborimizni ga qaratamiz. FONT (shrift) teg nomi, COLOR (rang) - esa tegning tarkibidagi qo'shimcha o'lcham, red (qizil) - rangning belgisi. Bu teg matnni qizil harflar bilan yozilishini anglatadi. Ayni misolda ranglarni o'zgartirish uchun red so'zi o'rniga boshqa rang nomini kiritish kifoY. Matn shu zahoti siz kiritgan rang bilan ifodalanadi. Ushbu teg va uning qo'shimcha o'lchamlari qanday yozilishini eslab qoling. Bu sizga keyinchalik ham as qotadi. Teglarining o'lchami bir nechta bo'lishi mumkin va ularni kiritganda har - biri orasida bo'sh katakcha ya'ni probel qoldirishni esdan chiqarmang.

Siz HTMLda teg tushunchasi haqida qisqacha tushunchaga ega bo'lgansiz. HTML teglarni ishlatganda ularni to'g'ri yozish muhim ahamiyatga ega. Quyida teglarni to'g'ri yozish va ishlatish shartlari haqida so'z yuritamiz. Bu borada quyidagi qoidalar mavjud:

- **Barcha teglar uch burchakli qo'shtirnoq ichida yoziladi** '<TEG>'.
Ochuvchi teg - uch burchakli burchakli qo'shtirnoq, teg nomi va uning qo'shimcha o'lchamlari (zarur bo'lgan taqdirda). Yopuvchi teg - uchburchakli qo'shtirnoq, '/' slesh belgisi va teg nomi.

- **Probel orqali yoziluvchi teglar o'lchamlari.**
Teg nomidan so'ng uning qo'shimcha o'lchamlari bo'lishi mumkin. Barcha o'lchamlar probel orqali ajratib yoziladi. Bitta harf hajmidagi bo'sh joy qoldirilish HTMLda probelni anglatadi.

- **Qo'shtirnoq ichida yoziluvchi yuklatilgan vazifalar ma'nosi.**

Ma'lum vazifani bajaruvchi buyruqlar tenglik belgisidan so'ng qo'shtirnoq "..."
ichida yoziladi. Agar probel ishlatilmagan taqdirda qo'shtirnoqni tushirib qoldirish mumkin. Har hil hatoliklar kelib chiqishini oldini olish uchun qo'shtirnoqlarni ishlatish maqsadga muvofiq.

- **Aksariyat teglar yakunlash buyrug'ini talab qiladi.**

Agar siz tegni yopmagan taqdiringizda aksariyat hollarda hatolik kelib chiqmaydi. Ammo ochilgan teg o'ldamlari butun hujjatga ta'sir qiladi. Jadval ichida yopilmagan taqdirda bu hatolikni keltirib chiqarishi mumkin. Bu haqda jadvallar tuzish bo'limida so'z yuritamiz.

- **Teglarni yopish qoidasi**

Har doim teglarni qat'iy tartibda rioya qilgan holda yoping. Birinchi navbatda eng oxiri ochilgan teg yopiladi va shu tariqa navbati bo'yicha barchasi yopiladi.
<TEG1> <TEG2> <TEG3> ... </TEG3> </TEG2> </TEG1>

- **Teg harflar yozish shartlari.**

HTML hujjatda teglarni katta yoki kichik harflar bilan yozilishi hech qanday ahamiyatga ega emas. Hamma teglarni katta yoki kichik harflar bilan yozishingiz mumkin. Hattoki mana bunday <teg></Teg> **Sizga maslahatimiz:** barcha teglarni bir hil shaklda yozgan ma'qul.

- **Teg o'ldamlarini yozish tartibi qanday bo'lishi ahamiyatsiz.**

Teg tarkibidagi o'ldamlar qanday tartibda yozishingiz hech qanday o'zgarishlarga sabab bo'lmaydi. Quyidagi ikki variant ham to'g'ri:

<TEG 1-o'ldam = yuklatilgan vazifa. 2 – o'ldam = yuklatilgan vazifa. 3 – o'ldam = yuklatilgan vazifa. >

<TEG 3 – o'ldam = yuklatilgan vazifa. 1 – o'ldam = yuklatilgan vazifa. 2 – o'ldam = yuklatilgan vazifa.>

Siz bilan birgalikda eng sodda HTML sahifa yaratishga harakat qilib ko'ramiz. Buning uchun bizga Notepad (Bloknot) matn muharriri kerak bo'ladi. Dasturni ishga tushirish uchun asosiy menyu Start (Pusk) > Programs (Programmi) > Accessories (Standartniye) > Notepad (Bloknot) buyrug'ini tanlang.

Quyida keltirilgan HTML teglarni Notepad (Bloknot)ga qanday bo'lsa shundayligicha yozing:

```
<HTML>
<HEAD>
<TITLE> Eng sodda ko'rinishdagi HTML sahifa!</TITLE>
</HEAD>
<BODY BGCOLOR="white">
<B><FONT COLOR="red">Salom!</FONT></B><BR>
Bu mening ilk HTML sahifam.
</BODY>
</HTML>
```

Navbatdagi amalga oshiradigan ishimiz, uni kompyuter xotirasiga saqlash. Buning uchun Notepad (Bloknot) dasturi File (Fayl) menyusi tarkibidagi Save As... (Soxranit kak...) buyrug'ini tanlang va fayl saqlanishi lozim bo'lgan papkani tanlang hamda faylni **nomi.HTML** kengaytmasi bilan saqlang.

HTML tilida web-sahifa yaratishda quyidagi ko'rsatmalarga amal qiling⁷:

- ushbu satrni sahifaning bosh sarlavhasi sifatida foydalaning;
- ushbu satrlarni sarlavha sifatida ko'rib chiqing;

⁷ Paul McFedries. Web Design Playground: HTML & CSS the Interactive Way 1st Edition., Manning Publications; 1st edition, 2019. 440 pages.

- ushbu matnning katta qismini alohida paragraf qiling;
- ushbu beshta ketma-ketlikni belgi qo‘yilgan ro‘yxatga aylantiring;
- ushbu oltita ketma-ket qadamlarni raqamlangan ro‘yxatga o‘zgartiring;
- ushbu iborani havolaga aylantiring.

Eslatma: har qanday HTML sahifa **fayl nomi.HTML** yoki **fayl nomi.htm** kengaytmasiga ega bo‘ladi. Faylni kompyuter xotirasiga saqlagach, uni brauzer dasturlari yordamida ochishingiz mumkin. Brauzer dasturlari internet ishlashini ta‘minlovchi maxsus dasturlar hisoblanadi. Gipermatnni o‘qish, web-resurslarda navigatsiyalash va ko‘rib chiqish dasturi. Web-brauzeri gipermatnlarni o‘qishdan tashqari, tovushni va video ma‘lumotlarni qaytadan chiqarish, ya‘ni, gipermediyani qayta chiqarish, tarmoqning boshqa kompyuterlari bilan ulanishni o‘rnatish va ularda ishlayotgan serverlarga web-hujjatlarga so‘rovlarni yuborish, tarmoqning boshqa foydalanuvchilari bilan muloqot tashkil qilish va uni quvvatlash kabi qo‘shimcha imkoniyatlarga ega bo‘lishi mumkin. Eng ommaviy tarqalgan web-brauzerlarga misol qilib Internet Explorer, Opera, Avant Brauzer, Mozilla kabi dasturlarni aytishimiz mumkin. 5.1-jadval brauzer dasturlarning ro‘yxati va imkoniyatlari keltirilgan.

5.1-jadval. Brauzer dasturlari ro‘yxati

№	Brauzer nomi:	Imkoniyatlari
1.	Opera 10.63	Opera Software kompaniyasi dasturidir. Bu dastur 43 tildagi foydalanuvchilar uchun mo‘ljallangan bo‘lib, Windows, Linux va Mac operatsion tizimlarida ishlaydi.
2.	Mozilla Firefox	Mozilla Firefox - tezkor, ko‘p funktsiyali brauzer dasturi. Firefox dasturida bir oynada bir nechta sahifani ochish mumkin. Shuningdek, siz giper havolalarni shu sahifani (siz ayni paytda ko‘rib turgan sahifani) tark etmasdan turib ochish imkoniyatiga ega bo‘lasiz. Brauzer web standartlarni juda yaxshi qo‘llab, unda juda qulay interfeys mavjud va o‘zida qidiruv paneli (Google va boshqalar) joylashgan.
3.	Google Chrome	Google Chrome – bu brauzer, murakkab texnologiyalarni va sodda dizaynni birlashtirgan va bu esa Internetda ishlashni tezlashtiradi va xavfsizligini oshiradi.
4.	Apple Safari	Apple Inc. kompaniyasining brauzer dasturi hisoblanadi. Dastlab dastur Mac OS operatsion tizimi uchun ishlab chiqilgan edi. Dasturda ko‘p oynalik interfeys mavjud bo‘lib, unda xavfsizlik uskunalari bilan boyitilgan.
5.	Maxthon	internet-brauzer dasturi. Maxthon web-sahifalarni ko‘p oynali rejimda ko‘rish imkonini beradi. Dasturning asosiy xususiyatlaridan biri ko‘p sonli bog‘lanishlar imkoniyati mavjud.
6.	Avant Browser	Bu brauzer ham ko‘p oynali rejimda ishlaydi. Ya‘ni asosiy oyna ichida ko‘plab oynalarni ochish imkoniyati mavjud. Avant Browser dasturida sichqoncha tugmasini bir bosish bilan grafiklar, video, tovushlar va Flash animatsiyalarni yuklashni taqiqlash yoki aksincha ruxsat berish imkoniyati mavjud.
7.	Netscape Navigator	Dastur o‘zining oddiy va yengil sozlanadigan foydalanuvchi interfeysi bilan va tezkorligi, ishlashga qulayligi hamda web–sahifalarni ko‘rishni boshqarish imkoniyatlari kengaytirilgan.

Bizga ma‘lumki yaratilgan web sahifalarni ishlatib ko‘rishda o‘zimiz bilgan yoki bilmagan holda brauzer dasturiga murojaat qilamiz. Ana shu paytda yaratilgan web-sahifani ochish uchun uning fayli ustida sichqonchani chap tugmasini ikki

marta ketma-ket bosish yetarli. Bundan tashqari avval brauzer dasturini (Internet Explorer) ishga tushirib olib, keyin esa **File > Open** buyrug'ini berish orqali ham faylni ishga tushirish mumkin. Hammamizga ma'lumki, web sahifani yaratish vaqtida uning qanday dizaynda chiqayotganini ko'rish uchun biz brauzer dasturlariga qayta – qayta murojat qilamiz. Ayrim professional dasturlarda (Dreamweaver va boshqalar) natijani ko'rish uchun to'g'ridan-to'g'ri maxsus tugma bosiladi. Bunda to'g'ridan-to'g'ri belgilangan brauzerda sahifa yuklanadi. Bu haqida Dreamweaver dasturida batafsil ma'lumot beramiz.

Siz yaratgan ilk HTML sahifa asosida HTML hujjat asosiy strukturasi tahlil qilamiz. Kompyuter xotirasiga saqlangan HTML hujjatni Notepad yordamida oching. Har qanday HTML hujjat ichki qismi quyidagilardan tashkil topgan bo'ladi:

<HTML>

bu teg HTML hujjatni ochish (boshlash) uchun xizmat qiladi va u HTML hujjatning birinchi ochuvchi va oxirgi yopuvchi teg hisoblanadi. Boshqacha aytganda web sahifaning boshlanish va tugash chegarasini ifodalaydi. Uning yagona vazifasi, mazkur hujjatni ochayotgan dasturga bu fayl HTML hujjatlari haqida axborot berishdan iborat.

<HEAD>

HEAD tegidan so'ng HTML hujjatning tizimli axborotlari haqida ma'lumot beradi. Bizning misolimizdagi yagona tizimli axborot - TITLE tegi hisoblanadi.

<TITLE>

Bu teg ochuvchi teg <HEAD> va yopuvchi teg </HEAD> orasida yoziladi. Ochuvchi va yopuvchi TITLE orasida kiritilgan axborot siz yaratayotgan hujjat nomini anglatadi. Siz kiritgan nom brauzer dasturi tomonidan va internetdagi qidiruv tizimlari tomonidan ishlatiladi. Biz keltirilgan misolda «Eng sodda ko'rinishdagi HTML sahifa!» matni brauzer dasturi yuqori qismida ya'ni sarlavha satrida namoyish etiladi.

<BODY>

BODY ochuvchi va yopuvchi teglari orasida yozilganlarning barchasi sizning HTML hujjatingizning tanasi hisoblanadi. Agar siz ushbu teg orasiga matn yozsangiz yoki tasvir kiritsangiz - bularning barchasini brauzer ekranda namoyish etadi. <BODY> tegi haqidagi batafsil ma'lumotni siz "Hujjat tanasini formatlash" bo'limida o'qishingiz mumkin.

Har qanday HTML hujjat quyida keltirilgan tartibda yozilishi shart:

<HTML> HTML hujjat boshi

<HEAD> Hujjat nomlanishining boshlanishi

<TITLE>...</TITLE> HTML sahifa nomi (sarlavhasi)

<HEAD> Hujjat nomlanishining yakunlanishi

<BODY BGCOLOR=white> Hujjat tanasining boshlanishi

.....

</BODY> Hujjat tanasining yakunlanishi

</HTML> HTML hujjat yakuni

Windows muhitidagi Notepad, TextPad, UltraEdit, EdutPlus kabi matn muharrirlari mavjud. Ushbu ko'rsatilgan barcha matn muharrirlarini HTML kodlarni yozishda qo'llash mumkin.

Ko'pgina matn muharrirlarida HTML kodlarni hattoki, veb matn muharrirlarida ham kodlarni yozish mumkin, masalan, PHPEDIT. Ba'zilar avtomatik ravishda operatorlar, funksiyalarni tanish va ularni har-xil ranglarda tasvirlash imkoniyatiga ega. Ba'zi veb matn muharrirlarida HTML hujjatni Web brauzerda sinab ko'rish tugmasi mavjud.

HTML hujjatni yaratishga mo'ljallangan maxsus dasturlar (HTML muharrirlar) ham mavjud: FrontPage, Adobe GoLive, Macromedia Dreamweaver, Nestcape Composer. Veb matn muharrirlari 2 turga bo'linadi:

- kod muharrirlari;
- WYSIWYG texnologiyasi (What You See Is What You Get – nimani ko'rsang shuni olasan) asosida ishlaydigan muharrirlar. Bu muharrirlar yordamida foydalanuvchi HTML buyrug'i va elementlarni yozmaydi, oddiy matn muharrirlaridek matn yozadi, tasvirlarni kerakli joyga joylashtiradi, formatlarni tanlaydi va h.

Web animatsiyalar turlicha bo'ladi. Masalan, *GIF Animator* dan foydalanib, tasvirlarni «Jonlantirish» mumkin. Bu Web reklamalarda ko'p qo'llaniladi.

GIF animatsiyalarni Ulead GIF Animator (<http://www.ulead.com>) va Animagic GIF Animator (<http://www.rtlsoft.com/animagic/index.html>) dasturlari yordamida hosil qilish mumkin. GIF tasvirlardan yuqoriroq o'rinda Flash yordamida yaratilgan animatsiyalar turadi. Bu animatsiyalar interaktiv bo'lganligi sababli Flash texnologiyalar ommalashib borayapti. Bu texnologiyani ayniqsa avtomobil ishlab chiqaruvchilar, web-bizneschilar, umuman, multimedia – prezentatsiya yaratmoqchi bo'lganlar keng qo'llaydilar.

Adobe (<http://www.adobe/flash/>) dan tashqari CoffeeCup Firestarter (<http://www.coffeecup.com>) dan ham foydalanib animatsiyalar yaratish mumkin.

Agar foydalanuvchi multimediya sahifasini yaratmoqchi bo'lsa, qo'shimcha instrumentlardan foydalanishga to'g'ri keladi, masalan, Quick Time, Windows Media yoki boshqa multimedia dasturlari.

Microsoft kompaniyasi Windows ME va undan keyingi chiqargan operatsion tizimlarida Windows Movie Maker dasturini qo'shib chiqarayapti. Bu dastur yordamida video roliklarni tahrirlash mumkin bo'ladi. Yana Sound Forge XP (<http://www.soniefoundry.com>) dasturidan tovushlarni tahrirlash va konvertatsiya qilishda foydalanish mumkin.

Web sahifalar yaratishda fayllarni joylashtirishning quyidagi usullari mavjud:

- **Hamma fayllar bitta katalogda.** Bir nechta HTML sahifali va bir nechta tasvirlarga ega kichik saytlarni yaratishda hamma fayllarni bitta katalogda joylashtirish mumkin. Bu usulning qulayligi shundaki, o'zaro murojaatlarda hech qanday yo'l ko'rsatish shart emas. Yomon tomoni esa, sayt rivojlanib, kengayib borsa fayllar ko'payib hammasi aralashib ketadi.

- **Funksional kataloglar.** Murakkab saytlarni yaratishning bir usulidir. Har bir katalogda sistemaning bitta funksiyali qismi joylashtiriladi. Eng yuqorida boshlang'ich fayl – index.htm va unga tegishli tasvirlar bo'ladi. Kataloglar nomlari taxminan quyidagicha bo'lishi mumkin: “Biz_haqimizda”, “Aloqalar”, “Mahsulotlar”. Har bir katalogda tegishli HTML fayllari va tasvirlar saqlanadi.

– **Fayl turlari bo'yicha kataloglar.** Bu usulda har bir katalogda bir xil turdagi fayllar saqlanadi. Bu yerdagi fayllar qaysi funksiyani bajarishi muhim emas. Tashqi katalogda faqat index.htm fayli joylashadi. Kataloglar nomlari taxminan quyidagicha bo'lishi mumkin: "images", "aloqalar", "downloads",...

Ixtiyoriy faylning kengaytmasi uning muhim qismidir. Chunki brauzer fayllarni uning kengaytmasiga qarab "taniydi".

Odatda birinchi sahifa har doim index.htm, index.html kabi saqlanadi. Ko'pchilik brauzerlar agar URL aniq ko'rsatilmasa bu fayllarni avtomatik yuklashga harakat qiladi. Masalan, brauzerning adreslar paneliga <http://www.microsoft.com> deb yozilsa <http://www.microsoft.com/index.htm> faylini yuklaydi.

Fayl nomiga qarab uni ajratib olish oson bo'lishligi uchun, faylni saqlashda unga mantiqiy nom tanlash kerak bo'ladi. Masalan, about_inst.htm, fiz_mat.htm, rector.jpg kabi nom qo'yish maqsadga muvofiq. Notepad ni ochamiz. HTML faylni yaratishni boshlaymiz.

Sarlavha yozish uchun:

```
<head>  
</head>
```

teglari yoziladi.

Brauzer bu teglar o'rtasidagi matnni sarlavha deb tushunadi va brauzerning eng tepa qismiga shu matnni yozadi. Endi sahifaning tanasini hosil qilamiz:

```
<body>  
</body>
```

Bu teglar o'rtasiga sahifada tasvirlanuvchi barchasi: matn va tasvirlarni joylashtirish kerak. HTML shablon bo'lishi uchun HTML sahifaning sarlavha va tanasini o'z ichiga olgan quyidagi zarur teglar yetishmayapti:

```
<html></html>
```

Demak HTML shablon quyidagi ko'rinishga ega bo'ldi:

```
<html>  
<head>  
</head>  
<body>  
</body>  
</html>
```

<head> va </head> teglari o'rtasiga quyidagi teglarni joylashtirish mumkin:

<title>, </title> – hujjat nomi. Masalan:

```
<title> TDPU ning Web sahifasi </title>
```

<meta /> – ma'lumotlar haqidagi ma'lumot. YA'ni bu yerda asosiy terminlar yoziladi. Qidiruv tizimlari ishlaganda aynan shu terminlar bo'yicha kidirish ishini olib boradi, saytlarni topadi. Masalan:

```
<meta name="keywords" content="TDPU, Nizomiy, Pedagogika, Universiteti,  
Fakultet, Student"/>
```

content 50-200 ta so'zni o'z ichiga olishi mumkin.

name=("keywords", "autor", "copyright", "description") – maxfiy qidirishda qo'llaniladi, HTTP serverga ruxsat berish uchun http-equiv qo'llaniladi.

<body> va </body> teglari o'rtasiga sahifaning tanasi joylashtiriladi. Bu sohada brauzer oynasida foydalanuvchi ko'rishi kerak bo'lgan matn, gipermurojaat, tasvirlar, jadvallar va h. joylashtiriladi.

<p>, </p> teglari ular orasidagi matnni alohida abzats ko'rinishida yozilishini ta'minlaydi. Masalan:

<p> Bu yangi abzats</p>

<p> </p> – bo'sh abzats joyini tashlaydi.

Agar yozuvni yangi abzatsdan emas, yangi qatordan boshlash kerak bo'lsa,
 tegidan foydalanish mumkin. Masalan:

<p> Toshkent shahri

Bunyodkor ko'chasi 27 uy

Tel:71254-92-02



2.2. HTML tilida matnlar bilan ishlash

Matn tahrirlash o‘zlashtirib olishingizni osonlashtirish uchun uni ikkiga bo‘lib tushuntirishni lozim topdik. Ularning har biri matn tahrirlashda ishlatiladigan HTML teglarni batafsil tahlili misollar yordamida aks ettirilgan.

- Matnning alohida bo‘laklarini tahrirlovchi teglar;
- Matnning yirik bo‘laklarini tahrirlovchi teglar;

Matnning alohida bo‘laklarini tahrirlovchi teglar

Quyida tasnifi keltirilgan barcha matn tahrirlash uchun ishlatuvchi teglar ochilgandan so‘ng ularni yopish shart. Ochilgan tegning ta’sir doirasi u yopilgunga qadar davom etadi. Agar siz tegni yopishni unutsangiz, uning ta’sir doirasi hujjat oxirigacha davom etadi. Barcha teglash quyidagi tartibda yoziladi: <TEG1><TEG2> Kerakli matn </TEG2></TEG1>. Har doim eng oxiri ochilgan tegni birinchi bo‘lib yopishni unutmang.

** Kerakli matn **

Bu teg orasida yozilgan matn qalin harflar bilan belgilanadi.

<I>Kerakli matn</I>

Matn o‘ng tomonga egiltirib yoziladi.

<BIG>Kerakli matn</BIG>

Matni kattalashtirish. Yozilgan matn boshqalariga nisbatan kattalashtirilgan harflar bilan belgilanadi.

<SMALL>Kerakli matn</SMALL>

Matni kichraytirish. Yozilgan matn boshqalariga nisbatan kichiklashtirilgan holda belgilanadi.

<STRIKE>Kerakli matn</STRIKE>

Teg orasida yozilgan matn ustidan chiziq tortiladi.

Kerakli matn

Matn ichidagi ayrim so‘zlarni kuchli ajratib ko‘rsatish uchun ishlatiladi.

Kerakli **_{matn}**

Ko‘rinib turganidek bu teg orasida kiritilgan matn, matn chizig‘idan pastda kichik harflar bilan yoziladi (indeks).

Kerakli **^{matn}**

Yuqorida keltirilgan tegning teskari shakldagisi bo‘lib, u matnni yuqorida kichik harflar bilan yozilishini ta’minlaydi (daraja).

<U>Kerakli matn</U>

Yozilgan matn ostida chiziq tortiladi. Bu teg yordamida ayrim so‘zlarni belgilab ko‘rsatish mumkin.

<CENTER>

Kerakli matn

</CENTER>

Bu teg kiritilgan matnni sahifa o‘rtasida bo‘lishini ta’minlaydi. Agar bu teg jadval ichida qo‘llansa, u holda bu jadval katakchasi ichidagi matnga ta’sir etadi. Bu tegni jadval ichida qo‘llash usullari haqida jadvallar bilan ishlash bo‘limida tushuntirib o‘tamiz.

Bu teg matn qatorlarini surish uchun xizmat qiladi. Ma'lum jumladan so'ng ikkinchi jumlaning yangi qatordan boshlamoqchi bo'lsangiz, jumla so'ngida ushbu tegni yozing.

Endi hech bo'lmaganda bitta qo'shimcha buyruqqa ega bo'lgan teglar bilan tanishib chiqamiz.

Kerakli matn

Bu teg harfning qo'shimcha o'lchamlarini belgilash uchun ishlatiladi. Quyida ushbu tegning eng ko'p qo'llaniladigan qo'shimcha buyruqlari bilan tanishib chiqamiz:

SIZE

Harf o'lchamini belgilashda qo'llaniladi. Matndagi harflar o'lchamini 1 dan boshlab xohlagan son bilan belgilash mumkin.

Masalan: Salom! Bu mening web sahifam. (Izoh)

Bundan tashqari siz SIZE o'lchamlarini plyus + va minus - bilan ham belgilashingiz mumkin. Bu usul qo'llanganda harf o'lchami ishlatilayotgan harfga nisbatan katta yoki kichik belgilanadi. Misol uchun siz 3 ga teng kattalikdagi harfdan foydalanayotgan bo'lsangiz, SIZE=+2 harf o'lchami SIZE=5 ga, SIZE=-1 o'lchami esa SIZE=2 ga teng bo'ladi.

COLOR

Bu tegdan so'ng rang nomi yoki rangni belgilovchi olti xonali lotin harflari hamda raqamli kod kiritiladi va shu tariqa harf rangi belgilanadi. Turli brauzerlar turli rang nomlarini belgilay oladi. Pastda keltirilgan jadvalda ranglar nomi yozilgan. Rang nomlari ingliz tilida kiritiladi. Rang nomi va ko'rinishini yodlab olishingiz uchun ularning har birini nomi o'sha rangda belgilangan. Bu ranglarni Netscape Navigator va Internet Explorer navigatorlari muammosiz taniydi va o'qiy oladi (2-ilova).

Rangni belgilovchi lotin alifbosi va raqamli kod '#' simvol bilan boshlanadi. Raqamlardan 1 dan 9 gacha yoki A dan F gacha bo'lgan lotin harflarini ishlatish mumkin. Rang kodi qanday bo'lishidan qat'iy nazar u RGB (RED – qizil, GREEN – yashil, BLUE – ko'k) ranglar tizimida aks ettiriladi. Rangni belgilash uchun uchta rangni bir-biriga qo'shish lozim. Har bir rang ulushini 0 dan 255 gacha bo'lgan miqdorda belgilanadi. Misol uchun toza qizil rangni belgilash uchun qizil rang miqdorini 255 ga, yashil va ko'k ranglar miqdorini esa 0 ga tenglashtirish lozim va bu mana bunday yoziladi . Olti xonali alifboli - raqamli kodini qo'llash juda ham qiyin. Shu bois bizning rangni kodini belgilab beruvchi jadvalimizdan foydalanishingiz mumkin (2-ilova).

FACE

Matnni qanday shriftda yozilishini belgilash uchun shrift nomi keltiriladi. Shrift nomi kompyuterda o'rnatilgan bo'lishi kerak yoki umumiy ishlatiladigan shriftlardan foydalanish mumkin. Agar siz qo'llagan shrift foydalanuvchining kompyuterida topilmagan taqdirda brauzer standart shriftni tanlaydi.

Masalan:

Salom!

Bu mening web sahifam.

Siz har doim tegining FACE buyrug'ini tanlaganingizda, standart shrift turlaridan foydalaning. Misol uchun Windows muhitida ishlovchi barcha kompyuterda o'rnatiladigan standart shriftlar quyidagilardan iborat:

Arial

Arial Black

Arial Narrow

Courier

Courier New

Garamond

Helvetica

Times

Times New Roman

va boshqalar.

Aytaylik siz Times va Times New Roman shriftining boshqacha turini tanlagan bo'lsangiz, u holda foydalanuvchi kompyuterida bu shrift o'rnatilmagan bo'lsa, brauzer siz tanlagan shriftni standart Times New Roman shriftiga almashtiradi.

Matnning yirik bo'laklarini tahrirlovchi teglar

Endi siz bilan matndagi yirik bloklar va abzatslarni tahrirlovchi teglar va ularning qo'shimcha buyruqlarining ishlash tartibini o'rganib chiqamiz.

<PLAINTEXT>

Ushbu tegdan so'ng kiritilgan barcha HTML teglarning ta'sirini butunlay kesadi va ular huddi ekranda ko'rsatilishi lozim bo'lgan matn kabi aks etadi. Bu teg boshqa teglardan tubdan farq qiladi va uni yopish shart emas. Agar matnda ayrim HTML teglarni ekranda brauzer aks ettirishini xohlasangiz bu tegdan foydalanishingiz mumkin. Biroq yuqorida aytganimizdek bu tegdan so'ng matnni HTML teglar yordamida tahrirlash mumkin emas. Chunki ularni brauzer huddi oddiy matn kabi tushunadi. Agar siz </PLAINTEXT> tegini kiritgan taqdiringizda ham bu <PLAINTEXT> ta'sir doirasini cheklay olmaydi. Web sahifangizda HTML teglarni matn shaklida aks ettirmoqchi bo'lsangiz, u holda uch burchakli qo'shtirnoq '<' o'rni < simvolini qo'llang.

<ADDRESS>

Kerakli matn

</ADDRESS>

Teg o'z nomi bilan adres ya'ni manzillarni belgilash uchun ishlatiladi. Bu teglar orasida yozilgan manzilni har qaysi brauzer o'zi xohlagan tarzda namoyish etadi. Teg yopilganidan so'ng kiritilgan matn yangi satrdan boshlanadi.

<BLOCKQUOTE>

Kerakli matn

</BLOCKQUOTE>

Bu teg matndagi paragraf va satrlarni belgilash uchun qo'llanadi.

<H1>, <H2>, <H3>, <H4>, <H5> va <H6>

yuqoridagi oltita teg sarlavhalarni belgilash uchun qo'llaniladi. Sarlavhalar qalin harflarda yoziladi. Bu teglarning bir-biridan farqi ularning o'lchamlarida. 1 dan 6 gacha bo'lgan sonlar sarlavha o'lchamlarini katta-kichikligini quyida keltirilgan tartibda belgilaydi:

- Sarlavha H1
- Sarlavha H2
- Sarlavha H3
- Sarlavha H4
- Sarlavha H5
- Sarlavha H6

Sarlavhalar <p> tegi kabi alohida abzats ko'rinishida ifodalanadi. Sarlavxalarning 1 dan 6 gacha darajalari mavjud.

```
<h1 align="center">Eng katta sarlavxa</h1>
<h2>2-darajali kattalikdagi sarlavxa</h2>
<h3>3-darajali kattalikdagi sarlavxa</h3>
<p>Bu oddiy matn</p>
<h4>4-darajali kattalikdagi sarlavxa</h4>
<h5>5-darajali kattalikdagi sarlavxa</h5>
<h6>Eng kichik sarlavxa </h6>
```

Har xil ajratuvchi gorizontaal chiziqlar <hr /> tegi yordamida qo'yiladi. Bu chiziq abzatslar oralig'ida joylashib, brauzer oynasining eni bo'ylab chiziladi. Agar foydalanuvchi brauzer oynasining o'lchamlarini o'zgartirsa, chiziq xam mos ravishda o'zgaradi.

<hr /> tegi
 tegi kabi ochiq teg hisoblanadi, ya'ni bu teglarni yopuvchi tegi bo'lishi talab qilinmaydi.

HR – "horizontal rule" (gorizontaal chiziq) dir. <hr /> tegining atributlari:

Atribut nomi	Vazifasi
Size	Gorizontaal chiziqning qalinligi. Eng katta o'lcham 5 pikselni qo'llash tavsiya etiladi.
Width	Chiziqning enini belgilaydi, foizlarda beriladi.
Align	Chiziqning joylashishi: brauzer oynasining chap tomonida, o'ng tomonida, o'rtasida.
noshade	Chiziqning tashqi ko'rinishini belgilaydi.

masalan:

```
<hr size="4" align="center" width="50%" noshade="noshade" />
```

Stillar ikki xil bo'ladi: fizik stillar va mantiqiy stillar.

Matn muharrirlari bilan ishlash jarayonida bilamizki, matnlarni har-xil ko'rinishda ifodalash mumkin: qalinlashtirilgan(polujirniy), qo'lyozma shaklida(kursiv), tagi chizilgan(podcherknutiy) Bu elementlarni ixtiyoriy grafik brauzerlari bir xil ko'rinishda ifodalaydilar. Ba'zi bir fizik stillar:

Element	Vazifasi
, 	Qalinlashtirilgan matn (polujirniy)

<i>, </i>	Qo‘lyozma shaklidagi matn(kursiv)
<tt>, </tt>	Harflar oralig‘ini kengaytirish
<u>, </u>	Tagi chizilgan matn(podcherknutiy)
<big>, </big>	Kattalashtirilgan matn
<small>, </small>	Kichiklashtirilgan matn
_,	Pastki indeks
[,]	Yuqori indeks

 Qalinlashtirilgan matn (полужирный)

<i> Qo‘lyozma shaklidagi matn (kursiv) </i>

<tt> Harflar oralig‘ini kengaytirish </tt>

<u> Tagi chizilgan matn (подчеркнутый)</u>

<big> KATTALAShtIRILGAN MATN</big>

<small> kichiklashtirilgan matn</small>

C_n

ax²+bx+c=0

Mantiqiy stillar:

Element	Vazifasi
, 	<i> tegiga analog teg.
, 	 tegiga analog teg.
<cite>, </cite>	Mazkur hujjatga izoh keltirish.
<dfn>, </dfn>	Dastur kodi
<samp>, </samp>	Dasturning ishlashiga misol. Oldingisi kabi ishlaydi.
<kbd>, </kbd>	Klaviaturadan kiritiladigan matn
<var>, </var>	O‘zgaruvchi yoki miqdor
<abbr>, </abbr>	Abbreviatura
<acronym>, </acronym>	Akronim

Mantiqiy stillar brauzerga matnni qay tarzda ekranga chop etish kerakligini bildiradi. Mantiqiy stillar fizik stillar ishlaymay qoladigan holatlar uchun ham o‘rinli bo‘lishi mumkin: uyali telefon internetga ulanganda tegi orqali qalinlashtirilgan matn uchrab qolsa, uni o‘qiy olmaydi. Bu holatda elementi kerakli natijani bera oladi.

<blockquote> elementi fizik atributlar qatoriga kiradi. Bu element abzatsni vizual holda oddiy matndan ajralib turishini ta‘minlaydi. masalan:

```
<blockquote>
  Diqqat e`lon! Dushanba kuni kafedra majlisi bo`lib o`tadi
</blockquote>
```

<address> elementi sahifa avtorini haqidagi ma‘lumotni o‘zida mujassamlashtirilgan matnni yaratish uchun ishlatiladi. Aksariyat brauzerlarda bu matn qo‘lyozma ko‘rinishiga ega bo‘ladi.

masalan,

```
<address>
```

```
web master: tdpu@tdpu.uz <br />
Toshkent shahri, Y.X.Xojib kuchasi 103 uy.
Tel: 252-74-38
</address>
```

<P>

Kerakli matn

</P>

Bu teg paragraflarni belgilaydi. Paragraf boshi va ohirida bo‘sh qator qoldiradi.

ALIGN

<P> tegining qo‘shimcha buyrug‘i bo‘lib u papagraf ichida yozilgan matnni tekislash uchun ishlatiladi. Bu buyruqning qo‘shimcha o‘lchamlari ALIGN – LEFT (matnni chap burchakdan tekislaydi), RIGHT (o‘ng tomonga tekislaydi), CENTER (sahifa markaziga tekislaydi) va JUSTIFY (matnni sahifa eni bo‘ylab tekislaydi). JUSTIFY ishlatilganda brauzer matnni chap va o‘ng tomon bo‘ylab bir xilda tekislab chiqishga harakat qiladi. Bu jarayonda so‘zlar orasi cho‘zilib ketishi ham mumkin.

<PRE>

Kerakli matn

</PRE>

Bu teg yordamida siz matnni qanday kiritgan bo‘lsangiz shunday ko‘rinishda bo‘lishi ta‘minlanadi. Quyida keltirilgan ikki misolda buni yaqqol ko‘rish mumkin. <PRE> tegidan tashqarida yozilgan matnda probel necha marta ishlatilishidan qat‘iy nazar u bittaga hisoblanadi. <PRE> tegi tarkibida kiritilgan matn qanday bo‘lsa shundayligicha ekranda ko‘rinadi.

Hujjat asosiy qismini tahrirlash

Bu bo‘limda hujjat tanasini belgilab beruvchi <BODY> tegi va uning qo‘shimcha buyruqlari haqida so‘z boradi. <BODY> tegini oldingi bo‘limlarda qisqacha tanishtirib o‘tgan edik. Bu teg bevosita HTML hujjatni ekranda ko‘rinishi lozim bo‘lgan matn, tasvir va boshqa elementlarni o‘z ichiga qamrab oladi. <BODY> ochuvchi hamda </BODY> yopuvchi teglari orasida barcha muhim elementlar joylashadi. Bu tegning qanday qo‘shimcha buyruqlarini quyida birgalikda ko‘rib chiqamiz.

BGCOLOR

Sahifaning orqa foni rangini belgilash uchun ishlatiladi. Bunda rangning ingliz tilidagi nomi yoki harf va sonlardan iborat kod qo‘llaniladi.

BACKGROUND

Sahifaning orqa foni uchun qo‘llaniluvchi tasvir nomi va u joylashgan papkaga yo‘lni ko‘rsatishda qo‘llaniladi. Tasvir joylashgan papkaga yo‘lni ko‘rsatish haqidagi ma‘lumotlarni «HTML sahifaga tasvir joylashtirish» bo‘limida o‘qishingiz mumkin.

TEXT

Hujjatda yozilgan matn harflarini qanday rangda bo‘lishini rang kodlarini kiritish orqali belgilab beradi. Agar siz ushbu tegni ishlatmagan taqdiringizda matn rangi qora rangda bo‘ladi.

LINK

Bu buyruq qo'llanilgan sahifadagi havola rangi teg o'lchamlarida qaysi rang kodi yoki nomi berilgan bo'lsa, o'sha rangda ekranda namoyon bo'ladi.

VLINK

Bu buyruq yordamida ishlatilgan havolalarni belgilab ko'rsatish uchun qo'llaniladi. Siz foydalanuvchilar adashib ketmasligi va havolada ko'rsatilgan sahifaga qayta kirmasligi hamda hali qo'llanilmagan havolalarni qo'llanilgan havolalardan ajratib ko'rsatish uchun ishlatishingiz mumkin.

ALINK

Bu buyruq kursor yordamida tanlangan va sichqoncha chap tugmasi bosilgan paytda havola rangini o'zgartirish uchun ishlatiladi.

Siz hozir o'qigan sahifada <BODY> tegining barcha qo'shimcha buyruqlari quyidagi tartibda qo'llaniladi.

<BODY BGCOLOR = #003399 TEXT = black LINK = blue ALINK = red VLINK = #006666>

Sahifada matndagi ayrim elementlarni ro'yxatlangan tartibda yozish qiyin emas va bundan tashqari o'quvchi matn bilan tanishib chiqishida qulayliklar yaratib beradi. Siz huddi Microsoft Word matn muharririda ro'yxatlarni qanday tuzsangiz huddi shu ko'rinishda HTML hujjatda ham maxsus teglar yordamida belgilash imkoniga egasiz.

Bu bo'limda ro'yxatlangan tartibda matnni kiritish usullari bilan tanishib chiqamiz:

<RO'YXATNI OCHUVCHI TEG>

< Ro'yxat elementining tegi > Matn

< Ro'yxat elementining tegi > Matn

< Ro'yxat elementining tegi > Matn

</ RO'YXATNI YOPUVCHI TEG >

Tartiblangan ro'yxat:

1. Ro'yxat elementi 1
2. Ro'yxat elementi 1
3. Ro'yxat elementi 1

Tartiblanmagan ro'yxat:

- Ro'yxat elementi 1
- Ro'yxat elementi 1
- Ro'yxat elementi 1

Matn ro'yxatini belgilovchi teglar tarkibida siz HTMLning boshqa matn tahrirlash uchun qo'llaniluvchi teglarini ishlatishingiz mumkin.

Ro'yxatlar tuzish – ro'yxatni tartiblash

Matn bo'laklarini tartiblashda sonlar yoki harflar bilan belgilash mumkin. Shuningdek ayni jarayonda simvollardan ham foydalanish imkoni mavjud. Bu haqda keyinroq to'xtalib o'tamiz.

Matn bo'laklarini tartiblashda ochuvchi va yopuvchi teglaridan foydalaniladi. Ushbu teglar ichida tartiblanuvchi matn oldiga buyrug'i yoziladi. tegining yagona qo'shimcha o'lchami TYPE bo'lib, u tartiblovchi belgi ko'rinishini aniqlab beradi.

Ko'pgina grafik brauzerlarda tartiblangan ro'yxatlarning qo'shimcha atributlarini qo'llash mumkin bo'ladi. Ularning yordamida tartiblashni ko'rinishini tanlash mumkin bo'ladi.

Teg nomi	Ro'yxat turi
<code><ol type="A"></code>	Katta harflardan tuzilgan ro'yxat
<code><ol type="a"></code>	Kichik harflardan tuzilgan ro'yxat
<code><ol type="I"></code>	Rim (katta) harflardan tuzilgan ro'yxat
<code><ol type="i"></code>	Rim (kichik) harflardan tuzilgan ro'yxat
<code><ol type="1"></code>	Arab raqamlaridan tuzilgan ro'yxat
<code><ul type="disc"></code>	Doiralardan tuzilgan ro'yxat
<code><ul type="square"></code>	Kvadratlardan tuzilgan ro'yxat
<code><ul type="circle"></code>	Aylanalardan tuzilgan ro'yxat

Bundan tashqari **start** atributi yordamida ro'yxatni boshlang'ich qiymatini tanlash orqali tartiblashni ixtiyoriy joydan boshlash mumkin bo'ladi. Masalan , ro'yxat 5 nomeridan boshlanishi kerak bo'lsa, `<ol start= "5">` deb yozish kerak.

Arab raqamlari bilan belgilash:

```
<OL TYPE=1>
<LI> Matn 1
<LI> Matn 2
<LI> Matn 3
</OL>
```

Natija:

1. Matn 1
2. Matn 2
3. Matn 3

Rim raqamlari bilan belgilash:

```
<OL TYPE=I>
<LI> Matn 1
<LI> Matn 2
<LI> Matn 3
</OL>
```

Natija:

- I. Matn 1
- II. Matn 2
- III. Matn 3

Rim harflari bilan belgilash:

```
<OL TYPE=i>
<LI> Matn 1
<LI> Matn 2
```

 Matn 3

Natija:

- i. Matn 1
- ii. Matn 2
- iii. Matn 3

Bosh harfalari bilan belgilash:

<OL TYPE=A>

 Matn 1

 Matn 2

 Matn 3

Natija:

- A. Matn 1
- B. Matn 2
- C. Matn 3

Kichik harflar bilan belgilash:

<OL TYPE=a>

 Matn 1

 Matn 2

 Matn 3

Natija:

- a. Matn 1
- b. Matn 2
- c. Matn 3

Ro'yxat tuzish – tartiblanmagan ro'yxat

Tartiblanmagan ro'yxat tuzishda va yopuvchi teglaridan foydalaniladi. Ro'yxat ichidagi tartib buyrug'i bilan belgilanadi. tegi uchun ham TYPE o'lchami ishlatiladi. TYPE o'lchami 3 xil turdagi ro'yxat kirita oladi:

Doirasimon belgi yordamida tartiblash:

<UL TYPE=circle>

 Matn 1

 Matn 2

 Matn 3

Natija:

- o Matn 1
- o Matn 2
- o Matn 3

Nuqtasimon belgi yordamida tartiblash:

<UL TYPE=disc>

 Matn 1

 Matn 2

 Matn 3

Natija:

- Matn 1
- Matn 2
- Matn 3

To‘g‘ri to‘rtburchaksimon belgi yordamida tartiblash:

<UL TYPE=square>

 Matn 1

 Matn 2

 Matn 3

Natija:

- Matn 1
- Matn 2
- Matn 3

Ro‘yxat tuzish – aniqlashtiruvchi ro‘yxatlar

Aniqlashtiruvchi ro‘yxat va uning izohidan iborat matnni tartibli tuzish uchun bir-necha teglar ishlatiladi.

Aniqlashtiruvchi ro‘yxatni <DL> ochuvchi va </DL> yopuvchi tegi yordamida amalga oshiriladi. Aniqlashtiruvchi ro‘yxat nomi <DT> tegidan so‘ng yoziladi va uni izohlovchi matn esa <DD> tegidan so‘ng kiritiladi. <DL> tegi qo‘shimcha o‘lchamlarga ega emas.

Aniqlashtiruvchi ro‘yxat:

<DL>

<DT> Aniqlashtiruvchi asosiy so‘z yoki jumla

<DD> Izoh beruvchi matn

</DL>.



2.3. HTML tilida jadvallar hosil qilish

Jadval tuzish – HTML bo‘yicha o‘zlashtirish qiyin bo‘lgan bo‘limlardan biri. Shu bois bu bo‘limni jadval tuzishni eng sodda usullaridan boshlaymiz. Web sahifani mukammal darajada yaratishni xohlasangiz, unda jadval tuzishni chuqur o‘rganib olishingiz kerak. Qisqa qilib aytganda jadvallar yordamida sahifani o‘zingiz xohlagan tarzda boshqara olishingiz mumkin. Jadval ichiga nafaqat matn, balki tasvir va boshqa elementlarni tartibli joylashtirish mumkin.

Jadvalni hosil qilishning asosiy elementi `<table>` hisoblanadi. Ishlatilishi shart bo‘lmagan element `<caption>` dir. `<table>` ning ichida `<tr>` – jadval qatorini bildiruvchi va `<td>` jadval ustunini bildiruvchi elementlar joylashadi. Ustun yoki qator sarlavhasini `<th>` elementi yordamida berish mumkin.

Jadvalni yaratish texnologiyasi:

1. Jadval xosil qilishni boshlash uchun `<table>` va `</table>` elementlari yoziladi.

2. Bu elementlar o‘rtasiga kerakli miqdorda qatorlar soniga mos `<tr>` va `</tr>` elementlari yoziladi.

3. Endi birinchi `<tr>` dan so‘ng kerakli miqdorda `<td>` va `</td>` elementlari yoziladi. Har bir `<td>` va `</td>` elementlari o‘rtasida jadvalning mos qiymatini kiritiladi. `<td>` va `</td>` elementlari o‘rniga `<th>` va `</th>` elementlarini ishlatish ham mumkin. Bu elementlar odatda faqat birinchi qator elementlarini yozishda ishlatiladi. Ko‘pgina grafik brauzerlar `<th>` va `</th>` lar o‘rtasidagi matnni qoraytirilgan (jirniy) kurinishda tasvirlaydilar.

4. Har bir `<tr>` elementidan keyin yuqoridagi uchinchi qadam takrorlanadi.

Jadval tuzish – umumiy qoidalar

Jadval tuzishda ishlatiladigan teglarni tasniflashdan avval, oddiy jadval tuzish sxemasi bilan tanishib chiqsak. Har qanday jadval quyidagi sxema bo‘yicha tuziladi:

`<JADVALNI OCHUVCHI TEG>`

`<JADVAL QATORINI OCHUVCHI TEG>`

`<JADVAL BO‘LAGINI OCHUVCHI TEG>`

Matn, tasvir yoki jadval

`</JADVAL BO‘LAGINI YOPUVCHI TEG>`

`</JADVAL QATORINI YOPUVCHI TEG>`

`</JADVALNI YOPUVCHI TEG>`

Har qanday jadvalda qatorlar va ustunlar mavjud bo‘ladi. Jadvalning bo‘lakchalarida esa axborot kiritilishi mumkin. Yuqorida keltirilgan misoldan ko‘rinib turibdiki, bitta ustun, bitta qator va bitta katakchadan iborat jadvalni tuzish uchun jadval ochiladi va yopiladi. Agar bitta ustundan iborat va bir nechta qatorni o‘z ichiga olgan jadvalni tuzish uchun, jadval qatorini ochuvchi va yopuvchi teglar orasiga bir nechta jadval katakchalarini ochuvchi va yopuvchi teglarini kiritish lozim.

Oddiy jadvallar tuzish

Jadval tuzish usullarini tushunib olish uchun dast avval eng oddiy jadvalni tuzishni o‘rganamiz. Biz tuzmoqchi bo‘lgan jadval uchta ustundan iborat bo‘ladi va

har bir ustunda 3 dona katakchalar o‘rin oladi. Jadvalning birinchi ustunini «Mahsulot», ikkinchisini esa – «Rangi», uchinchisini «Narhi» deb nomlaymiz. Biz tuzgan jadval mana bunday ko‘rinishda bo‘ladi:

5.2-jadval.

Mahsulot	Rangi	Narhi (so‘m)
Qalam	Qora	100
Qalam ochgich	Qora	500

Endi ushbu jadvalni tuzishda qanday HTML teglardan foydalanilganligini ko‘rib chiqamiz:

```
<TABLE>
<TR>
<TD>Mahsulot</TD>
<TD>Rangi</TD>
<TD>Narhi(co‘m.)</TD>
</TR>
<TR>
<TD>Qalam</TD>
<TD>Qora</TD>
<TD>100</TD>
</TR>
<TR>
<TD>Qalam ochgichi</TD>
<TD>Qora</TD>
<TD>500</TD>
</TR>
</TABLE>
```

Jadvalni ochuvchi teg

1 - qatorni ochuvchi teg

1 - Jadval bo‘lagini ochuvchi va yopuvchi teg

2 - Jadval bo‘lagini ochuvchi va yopuvchi teg

3 - Jadval bo‘lagini ochuvchi va yopuvchi teg

1 - qatorni yopuvchi teg

Jadvalni yopuvchi teg

Misolda ko‘rinib turibdiki, jadvalni <TABLE> va </TABLE> teglari yordamida ochiladi va yopiladi. Qatorlar uchun <TR> va </TR> teglari qo‘llaniladi, <TD> va </TD> teglari esa jadval ustunlarini (katakchalarini) belgilash uchun ishlatiladi.

Oddiy jadvalga misol:

1	2	3	
1x1	1x2	1x3	1
2x1	2x2	2x3	2

Uchta ustun va ikkita qatordan iborat jadval yaratilish kerak bo‘lsin:

```
<table border="1">
<caption>2ta satr va 3ta ustunli jadval</caption>
<tr>
<td>1x1</td>
<td>1x2</td>
<td>1x3</td>
</tr>
<tr>
<td>2x1</td>
<td>2x2</td>
<td>2x3</td>
</tr>
</table>
```

Jadval nomini <caption> elementi yordamida berilishi aytib o‘tili. Bundan tashqari bu elementni IZOH sifatida ishlatish ham mumkin. Buning uchun <caption align= «bottom»> deb yozilsa, jadvalning pastki qismida chiquvchi izohga aylanadi.

Jadval chizish - HTML teg o‘lchamlari

Murakkab jadvallar tuzishni boshlashdan avval <TABLE>, <TR> va <TD> teglariga qisqacha tasnif berib o‘tamiz. Mazkur uchta teg bilan qo‘llaniladigan bir xildagi o‘lchamlar mavjud bo‘lib, faqatgina ularning farqli tomoni, ularning ta’sir doiralari chegaralanishida.

BGCOLOR

Tegining yordamida jadvalning orqa fonida qanday rang qo‘llanilishi belgilanadi.

BACKGROUND

Jadvaldagi bo‘lakcha foni sifatida berilayotgan tasvirni joylashgan joy va uning nomini belgilash uchun ishlatiladi. Agar siz bu o‘lchamni <TABLE> tegidan so‘ng qo‘llasangiz, u holda siz tanlagan tasvir jadvaldagi har bir kataklar foni sifatida qo‘llaniladi. <TR> tegidan so‘ng qo‘llanilganda, mazkur qatorda joylashgan hamma

katakchalar uchun, <TD> tegidan so'ng qo'llanilganda esa aniq biror katakchagacha fon o'rnatiladi.

WIDTH

Jadval eni (kengligi)ning o'lchami (umumiy jadval uchun xos). Bu o'lcham bilan jadvaldagi ma'lum bir ustuncha yoki bo'lakning kengligini piksel yohud foiz o'lchamida belgilash mumkin. Bu o'lcham <TABLE> yoki <TR> tegi bilan qo'llanilgan taqdirda ham bir xil ko'rinishdagi, ya'ni siz bergan o'lchamlarni hisobga olgan holda jadval enining kengligini belgilaydi. Shuni doim yodda tutish lozimki, siz bu o'lchamdan foydalangan holda jadval enining kengligini belgilaganingizda bir xildagi o'lchov turidan foydalanishingiz maqsadga muvofiq. Agar siz jadval kengligini piksellarda belgilasangiz (misol uchun WIDTH=100), boshqa o'lchovlar ham pikselda belgilanishi lozim. Agar jadval kengligi foizlar hisobida berilsa (misol uchun WIDTH=50%), boshqa o'lchovlar ham foiz hisobida berilishi lozim. Ayni hollarda brauzer siz yaratgan jadvalni ekranning teng yarmi, ya'ni 50% ga joylashtiradi.

HEIGHT

Jadvalning, qator yoki katakchalarning balandligi o'lchovlarini piksellar yoki foiz hisobida belgilash uchun qo'llaniladi.

BORDER

Jadval ustuni va qatorlari kesishmalarini belgilovchi chiziqlarni qalinligini piksellar hisobida belgilash uchun qo'llaniladi. Agar jadval chiziqlarini nolga teng qilib belgilansa BORDER (BORDER=0) chiziq (border) brauzerda ko'rinmaydigan bo'lib qoladi.

ALIGN

Jadvalning ekranga nisbatan joylashish o'rnini belgilaydi. Agar jadval ichida qo'shimcha jadval tuzilgan bo'lsa, u holda ichki jadvalga bu o'lcham kiritilganda, ichki jadval joylashishi o'zni tashqi jadvalga nisbatan belgilanadi. ALIGNning quyidagi qo'shimcha o'lchamlari mavjud: ALIGN - center (Jadvalni markaz bo'ylab joylashtirish), left (chap burchak bo'ylab joylashtirish) va right (o'ng burchak bo'ylab joylashtirish).

CELLPADDING

Border enining kengligini piksellar yordamida kengaytirish yoki toraytirish.

CELLSPACING

Jadvaldagi katakchalar chegara enining piksellar yordamida kengaytirishi yoki toraytirishi.

Misol:

```
<TABLE BORDER=0 WIDTH=100% CELLSPACING=5  
CELLPADDING=5 BGCOLOR=blue>  
<TR BGCOLOR=yellow>  
<TD>Mahsulot</TD>  
<TD>Narhi</TD>  
</TR>  
<TR BGCOLOR=white>  
<TD>Qalam</TD>  
<TD>100 so'm</TD>
```

```
</TR>
</TABLE>
```

```
<TR> va <TD>
```

ALIGN

Jadvaldagi har bir katakchanning ichidagi elementlarni gorizontalar tarzda tartiblash uchun qo'llaniladi. Uning mavjud o'lchamlari quyidagilardan iborat: - left (chap tomonga tekislash), right (o'ng tomonga tekislash) va center (markaz bo'ylab tekislash).

VALIGN

Jadvaldagi har bir katakchanning ichidagi elementlarni vertikal tarzda tartiblash uchun qo'llaniladi. Uning mavjud o'lchamlari quyidagilardan iborat: - top (katakning yuqori qismiga nisbatan), middle (Yuqori va quyi qismiga nisbatan markazga tekislash).

Jadval tuzish qoidalari

Bu sahifada jadval tuzish jarayonida siz amal qilishingiz lozim bo'lgan eng asosiy qoida va maslahatlar keltirilgan.

- **<TABLE>, <TR> va <TD>** teglarini tartibiga rioya qiling

Har doim jadval tuzish sxemasini buzmaslikka harakat qiling. Avval aytib o'tilgandek eng avvalo jadvalni oching va so'ngra jadval qatorlari va ustunlarini tuzing va eng oxirida jadvalni yoping. Ushbu tartib buzilgan taqdirda turli hatoliklar kelib chiqishi mumkin.

- **<TABLE> tegini jadval so'ngida yoping**

Agar <TABLE> tegini yopmasangiz kutilmagan hatoliklar vujudga keladi. Masalan, Netscape Navigator siz tuzgan jadvalni ekranda umuman ko'rsatmaydi.

- **Ustunda joylashgan qatorlar soni**

Har bir qatorda kataklarning soni bir xilda bo'lishiga e'tibor bering. Bu tartibga, agar siz ayrim kataklarga hech qanday axborot kiritmagan taqdiringizda ham ularni yaratish lozim. Bunday holatlarda bo'sh katakchalarni mana bunday ko'rinishda yozish lozim: <TD> </TD>. ' ' simvoli brauzerga mazkur katakda hech qanday axborot mavjud emasligi haqida xabar beradi. Agar ushbu simvol yozilmagan taqdirda brauzer bu katakni aks ettirmaydi va <TD> tegida BGCOLOR buyrug'i ishlatilgan taqdirda fon rangi ifoda etilmaydi va hatolik yuz beradi.

- **Har bir jadval tegini yangi qatorda yozing**

Jadval tuzishda adashib ketmaslik uchun har bir jadval qatori, ustuni va katakchalar belgilanganda ularni bir-biridan ajratib yozishga harakat qiling. Quyida ana shu holatga misol keltirilgan.

```
<TABLE>
<TR>
<TD> ... </TD>
<TD> ... </TD>
<TD> ... </TD>
</TR>
<TR>
```

```
<TD> ... </TD>
<TD> ... </TD>
<TD> ... </TD>
</TR>
</TABLE>
```

Ushbu qoidalarga rioya qilsangiz jadval tuzishda turli qiyinchiliklarga duch kelmaysiz hamda siz tuzgan jadvallarda hatoliklar bo'lmaydi.

Qator elementlarida ikkita muhim atributlar mavjud: **align** va **valign**. Ular yordamida jadval yacheykasidagi matnni gorizontaal va vertikal holatda ixtiyoriy joylashtirish mumkin. **align** elementi **center**, **left**, **right** (yacheykadagi matnni o'rtadan, chapdan, o'ngdan tekislash), **valign** atributi esa **top**, **bottom**, **center** (yacheykadagi matnni tepadan, pastdan, o'rtadan tekislash) kabi atributlarga ega:

```
<table border="1" align="center">
```

```
<tr align="center">
```

```
<th>Yillar</th>
```

```
<th>Choy</th>
```

```
<th>Kofe</th>
```

```
</tr>
```

```
<tr align="left">
```

```
<td>2001</td>
```

```
<td>35%</td>
```

```
<td>65%</td>
```

```
</tr>
```

```
<tr align="right">
```

```
<td>2002</td>
```

```
<td>45%</td>
```

```
<td>55%</td>
```

```
</tr>
```

```
<tr align="center">
```

```
<td>2003</td>
```

```
<td>55%</td>
```

```
<td>45%</td>
```

```
</tr>
```

```
</table>
```

Ba'zi hollarda, masalan, jadval elementlari bir xil, o'xshash ma'lumotlarga ega bo'lsa **valign** atributining natijasi ko'rinmay qolishi mumkin. Lekin yacheykalarda har xil hajmdagi ma'lumotlar saqlanadigan bo'lsa, uning natijasini ko'rish mumkin.

Bundan tashqari `<th>` elementining `<scope>` atributi mavjud bo'lib, u sarlavha qator yoki ustunga tegishli ekanligini bildiradi. Masalan:

```
<table border="1" align="center">
```

```
<tr align="center">
```

```
<th scope="row">Yillar</th>
```

```
<td>Choy</td>
```

```
<td>Kofe</td>
```

```
</tr>
```

```

<tr align="left">
<th scope="row">2001</th>
<td>35%</td>
<td>65%</td>
</tr>
<tr align="right">
<th scope="row">2002</th>
<td>45%</td>
<td>55%</td>
</tr>
<tr align="center">
<th scope="row">2003</th>
<td>55%</td>
<td>45%</td>
</tr>
</table>

```

Shunday hollar bo‘ladiki, jadvalning bir nechta yacheykalarini (bir nechta ustunni yoki bir nechta satrni) bir xil o‘lchamda ifodalashga to‘g‘ri keladi (bir-biriga yaqin axborotlar yoki jadvalning chiroyli chiqishi ...). Har qanday holatda ham <th> va <td> lar uchun: **colspan** va **rowspan** atributlari qo‘llaniladi. Ulardan biri yacheykani gorizontal o‘lchamini, ikkinchisi vertikal o‘lchamini bildiradi.

Jadval orqasidagi fonni rangini tanlash imkoniyati ham mavjuddir. Buning uchun **bgcolor** atributi ko‘llaniladi:

```

<table border="1" align="center">
<tr align="center" bgcolor="red">
<th >Yillar</th>
<th Choy</th>
<th>Kofe</th>
</tr>
<tr align="left" bgcolor="brown">
<td>2001</td>
<td >35%</td>
<td>65%</td>
</tr>
<tr align="right" bgcolor="orange">
<td>2002</td>
<td >45%</td>
<td>55%</td>
</tr>
<tr align="center" bgcolor="yellow">
<td>2003</td >
<td >55%</td>
<td>45%</td>
</tr>
</table>

```

<table> elementi **align** va **border** atributlaridan boshqa atributlarga ham ega. Ular quyidagilardir:

1. **width**. Jadvalning enini oʻrnatadi. Ikki holatda oʻlchamini berish mumkin: foizlarda (`width="50%"`) va piksellarda (`width="250"`). Bu atribut berilmaganda jadvalning ichidagi maʼlumotlar oʻlchamiga qarab jadval chiziladi. Oʻlchamlarni piksellarda berish tavsiya etilmaydi. Chunki hamma foydalanuvchilar ham ekranning oʻlchamini har doim ham bir xilda ifodalamaydilar (800x600, 1024x768 kabi oʻlchamlar mavjud).

2. **border**. Jadval ramkasi qalinligini tanlash imkoniyatini beradi. Oʻlchamlar piksellarda belgilanadi (`border="3"`). Bu atribut berilmaganda jadval chegarasi oq rangda ifodalanadi (koʻrinmaydi).

3. **cellspacing**. Jadval ustunlari orasidagi masofani bildiradi. Oʻlchamlar piksellarda beriladi. Atribut qoʻllanilmaganda 1 ga teng deb qabul qilinadi.

4. **cellpadding**. Jadval qatorlari orasidagi masofani bildiradi. Oʻlchamlar piksellarda beriladi. Atribut qoʻllanilmaganda 1 ga teng deb qabul qilinadi.

5. **rules** va **frames**. Jadvalning ichki (`rules`) va tashqi (`frames`) chegaralari qanday chizilishi kerakligini bildiradi.

Jadvallar HTML standartlariga kirgandan boshlab dizaynlar yaratish professionallik darajasi bir necha marotaba ortdi. Jadval yacheykalarini boshqarib, ularga ixtiyoriy elementlarni – tasvirlarni, matnlarni, roʻyxatlarni, gipermuvojatlarni qayerda kerak boʻlsa oʻsha yerda ishlatish imkoniyati mavjud boʻladi. Shu tariqa butun web sahifalarni qamrab oluvchi jadvallarni yaratish yaxshi natija beradi. Yacheykalarni umumlashtirish, kerakli oʻlchamgacha kengaytirib, toraytirish, ular oraliqlarini belgilash orqali yuqorida aytilgan natijaga erishish mumkin.

Murakkab jadvallar tuzish

Oddiy jadval tuzish usullarini oʻrganib oldingiz. <TD> tegining hali siz bilmagan baʼzi qiziq oʻlchamlari mavjud boʻlib ularni qoʻshish orqali ajoyib natijalarga erishish mumkin. Aytaylik siz quyida keltirilgan jadvalga oʻxshash jadval tuzmoqchisiz (5.3-jadval).

5.3-jadval

№	Mahsulot nomi	Narhi (soʻm)
1	Qalam	100
2	Ruchka	150
3	Daftar	400
Jami: 650 soʻm		

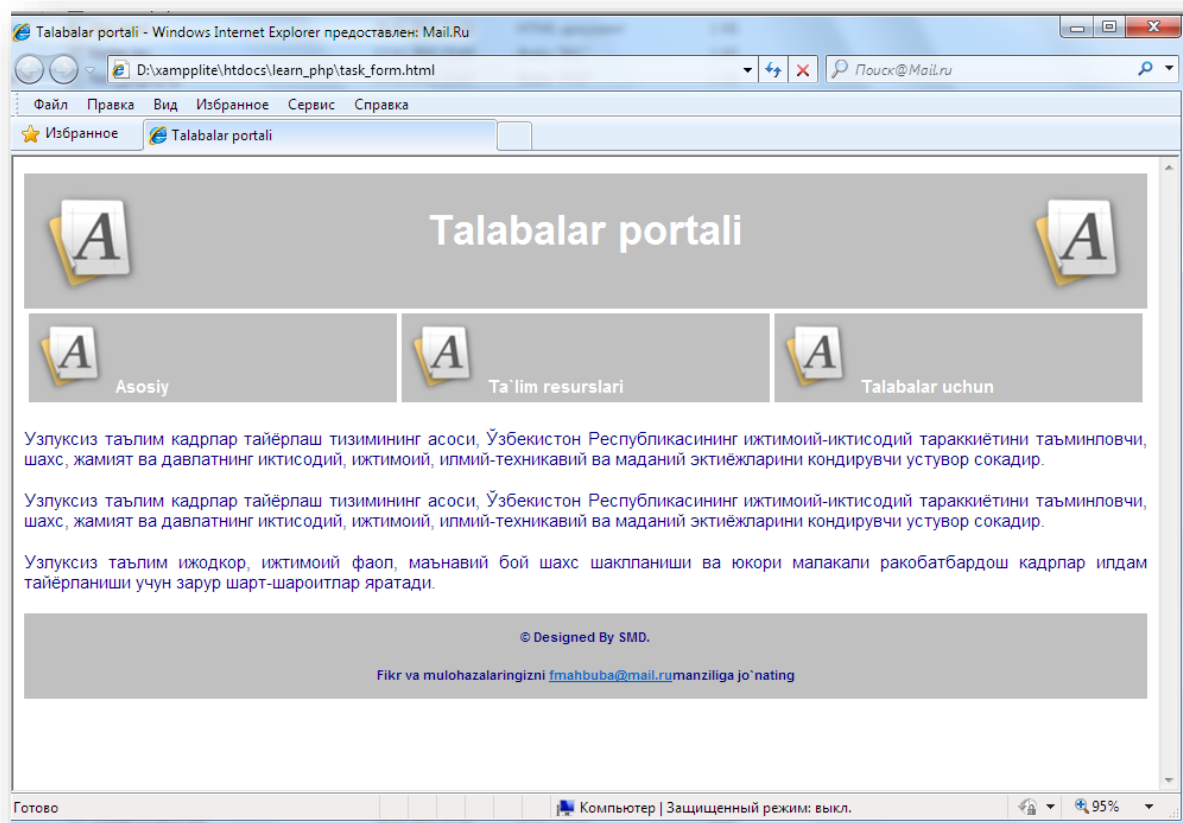
Eʼtibor bergan boʻlsangiz jadvalning soʻngidagi qator boshqalaridan farqli oʻlaroq bitta katakchadan iborat. Biz avval aytib oʻtganimizdek jadvaldagi barcha kataklar soni bir xil emas. Bu misolda oxirgi qatordagi uchta kataklar bir-biri bilan birlashtirilgan xolos. Buni <TD> tegining COLSPAN buyrugʻi yordamida amalga oshirish mumkin.

COLSPAN

Jadvaldagi bir qator yoki ustundagi bir nechta kataklarni birlashtiradi. Bu buyruq o‘lchamiga siz nechta katakchani birlashirmoqchi bo‘lsangiz, ularning sonini kiritishingiz lozim. Quyida, yuqoridagi jadvalda bu usul qanday amalga oshirilganini ko‘rsatib beruvchi misol keltirilmoqda. Bu misolda jadvalning aynan COLSPAN buyrug‘i qo‘llanilgan qismigina yozilgan:

```
<TR>  
<TD COLSPAN=3 BGCOLOR=red>  
<FONT COLOR=white>Jami: 650 so‘m</FONT>  
</TD>  
</TR>
```

Umumiy holda jadvallar ustunlar va qatorlardan iborat bo‘lib, ular ustida ishlash imkoniyati mavjuddir. Gazeta yoki jurnallarni varaqlab ko‘plab kataklarga bo‘linganligini ko‘rish mumkin. Xuddi shunday qilib web-sahifalarni ham kataklarga bo‘lish mumkin. Agar sahifa bir nechta bo‘laklarga bo‘linsa, u o‘quvchi tomonidan oson tushuniladi. Masalan, quyidagi ko‘rinishdagi web-sahifani yaratmoqchi bo‘laylik:



Buning uchun quyidagi kodni yozishga to‘g‘ri keladi:

```
<!-- page header -->  
<table width=«100%» cellpadding = 12 cellspacing =0 border = 0>  
<tr bgcolor = silver>  
<td align = left><img src = «logo.png»></td>
```

```

<td>
<h1>Talabalar portali</h1>
</td>
<td align = right><img src = "logo.png "></td>
</tr>
</table>
<!-- menu -->
<table width = "100%" bgcolor = white cellpadding = 4 cellspacing = 4>
<tr >
<td width = "25%">
<img src = "s-logo.png "> <span class=menu>Asosiy</span></td>
<td width = "25%">
<img src = "s-logo.png "> <span class=menu>Ta`lim resurslari</span></td>
<td width = "25%">
<img src = "s-logo.png "> <span class=menu>Talabalar uchun</span></td>
</tr>
</table>
<!-- page content -->
<p>Uzluksiz ta`lim kadrlar tayyorlash tizimining asosi, O`zbekiston Respublikasining ijtimoiy-iktisodiy tarakkiyotini ta`minlovchi, shaxs, jamiyat va davlatning iktisodiy, ijtimoiy, ilmiy-texnikaviy va madaniy ehtiyojlarini kondiruvchi ustuvor sokadir. </p>
<p>Uzluksiz ta`lim kadrlar tayyorlash tizimining asosi, O`zbekiston Respublikasining ijtimoiy-iktisodiy tarakkiyotini ta`minlovchi, shaxs, jamiyat va davlatning iktisodiy, ijtimoiy, ilmiy-texnikaviy va madaniy ehtiyojlarini kondiruvchi ustuvor sokadir.
</p>
<p>Uzluksiz ta`lim ijodkor, ijtimoiy faol, ma`naviy boy shaxs shakllanishi va yukori malakali rakobatbardosh kadrlar ildam tayyorlanishi uchun zarur shart-sharoitlar yaratadi.</p>
<!-- page footer-->
<table width = "100%" bgcolor = silver cellpadding = 12 border =0>
<tr>
<td>
<p class=foot>&copy; Designed By SMD.</p>
<p class=foot>Fikr va mulohazalaringizni <a href = "mailto:fmahbuba@mail.ru">fmahbuba@mail.ru</a>manziliga jo`nating </p>
</td>
</tr>
</table>

```

Xuddi shuningdek, menyular uchun uchta ustun hosil kilib, barcha ustunlarga 25% dan berib, bo‘lish mumkin. Asosiy sahifani o‘zini ham ikkiga bo‘lib, chap menyularni hosil qilish mumkin.

Jadvaldagi **colspan** atributi qatoridagi ustunlarni umumlashtirish imkoniyatini beradi. Bu yerda colspan=«n» yozuvi nechta qatorni umumlashtirish kerak bo'lsa, n shunga teng deb yoziladi.

Jadvaldagi **rowspan** esa bir nechta qatorlarni umumlashtirish imkoniyatini beradi. Nechta qator umumlashtirilishi kerak bo'lsa rowspan=«n» deb yoziladi.

Jadvalning yana quyidagi atributlari mavjud: **frame** va **rules**. Birinchi atribut **border>0** bo'lgan holatda tashqi chegarasining qanday bo'lishi kerakligi, ikkinchi atribut esa yacheykalarni ajratuvchi chegaralar qanday bo'lishi kerakligini bildiradi.

frame atributi qabul qilishi mumkin bo'lgan qiymatlar quyidagi jadvalda keltirilgan:

Atribut nomi	Vazifasi
Void	Chegara yo'q
Above	Yuqori chegara
Below	Pastki chegara
Nsides	Chap va o'ng chegaralar yo'q
Vsides	Yuqori va pastki chegaralar yo'q
lhs	Faqat chap chegara mavjud
Rhs	Faqat o'ng chegara mavjud

Rules atributi qabul qilishi mumkin bo'lgan qiymatlar quyidagi jadvalda keltirilgan:

Atribut nomi	Vazifasi
None	Chizik yo'q
Groups	<thead>, <tfoot>, <tbody> elementlari ishtirok etganda faqat qatorlar oralig'ida chiziq bo'ladi, <colgroup>, <col> elementlari ishtirok etganda faqat ustunlar oralig'ida chiziqlar bo'ladi.
Rows	Chiziklar faqat qator oraliqlarida
Cols	Chiziklar faqat ustun oraliqlarida
All	Hamma chiziklar bo'ladi



2.4. HTML tilida rasmlar joylashtirish teglari

Internet bilan ishlovchi dasturlar tasvirlarni (grafik elementlarni) ochish imkoniga ega bo'lishi bilan internetda mavjud sahifalarning deyarli barchasi tasvirli axborotlarni nashr eta boshladi. Siz yaratgan web-sahifa qiziqarli va chiroyli dizayn asosiga qurilgan bo'lsa uning o'quvchilari soni shubhasiz ortadi. Tasvirlar yordamida harakatlanuvchi tugmalar va Gif animatsiyalarni kiritish mumkin. Biroq tasvirlar bilan ishlashda ularni hajmi va formatiga e'tibor berish kerak. Hozirgi paytda internetga joylashtirilayotgan deyarli barcha sahifalarda ishlatilayotgan tasvirlar formati *.JPEG yoki *.GIF ni tashkil etadi. Bunga asosiy sabab, bu formatdagi tasvirlar hajmi juda kichik. Bu formatdagi tasvirlar ishlatilganda web-sahifa hajmi ortib ketmaydi va sizning web-sahifangizdan foydalanuvchi sahifani internetda ochilishini uzoq vaqt kutmaydi.

Agar siz boshqa formatdagi tasvirlardan web-sahifani bezashda foydalansangiz uning hajmi ortib ketadi va undan foydalanuvchilarga sahifani ochishda muammolar keltirib chiqaradi. Web-sahifani bezatishda uning dizayniga *.GIF formatdagi tasvirlardan foydalaning. Boshqa maqsadlarda *.JPEG dan foydalanishingiz mumkin.

 tegini qo'llash qoidalari

 tegi bilan bog'liq bo'lgan, quyida keltirilgan asosiy qoidalar bilan tanishib chiqing va o'z sahifangizni yaratishda ushbu qoidalarga doim rioya qilishga harakat qiling.

- **Tasvir nomi va joylashgan o'rnini yozganda harflarning bir hilda bo'lishini ta'minlash katta ahamiyatga ega.** Tasvir joylashgan papka nomi va tasvirning nomini kiritganda harflarni bir hilda yozilishi shart. Biror bir harfni tushirib qoldirish yoki hato yozilishi tasvirni ochishda hatoliklarni keltirib chiqaradi.
- **Tasvir yoki papka nomini kiritganda probel ishlatmang.** Web-sahifani yaratayotganingizda fayllar yoki papkalar nomini kiritganda probel (bo'sh joy) ishlatmang. Agar siz kiritgan nom ikki yoki undan ortiq so'zlardan iborat bo'lsa, probel o'rnida (pastgi chiziqcha) ishlatishingiz mumkin. Masalan: .../Chet_elga_sayohat/Ispaniya/Malaga_01.jpeg
- **Internetda ishlatiluvchi tasvir formatlari.** Hozirgi paytda internetda ikki hil formatdagi tasvirlar keng qo'llanilib kelinmoqda. Bu formatlar: GIF (bu formatdagi tasvirlar *.gif qisqartmasiga ega) hamda JPEG (bu formatdagi tasvirlar esa *.jpg yoki *.jpeg qisqartmasiga ega). Siz ham ushbu formatdagi tasvirlardan foydalaning. Boshqa formatdagi tasvirlarni aksariyat brauzerlar ocha olmaydi.
- **Har doim WIDTH va HEIGHT o'lchamlarini kiritishga harakat qiling.** Bu sizning web-sahifangizni ochilishini tezlashtiradi va tasvirlarni asl o'lchamda bo'lishini ta'minlaydi.
- **Tasvir hajmini sun'iy kichraytirish.** WIDTH va HEIGHT orqali tasvirning real o'lchamini o'lchamlarini kichraytirish mumkin. Ammo bu usuldan iloji boricha kamroq foydalanish kerak. Eng yaxshi usul tasvirni real o'lchamini grafik muharrirlar yordamida kichraytirishdir. Bu usul tasvirning ochilish vaqtini kamaytiradi.
- **Tasvir hajmini sun'iy kattalashtirish.** Tasvir o'lchami kattalashtirilganda yoki kichraytirilganda uning sifatida o'zgarish sodir bo'ladi. Agar siz o'lchamlarni

o'zgartirganda bo'yi va enini proporsionalligiga e'tibor bermasangiz, tasvir sifatini buzilishiga olib keladi. WIDTH va HEIGHT o'lchamlari orqali tasvir hajmini kattalashtirganda, real o'lchamni 2 marta, 3 marta va shu tariqa oshishiga erishish kerak. Masalan, tasvirning real o'lchami WIDTH = 100, HEIGHT = 200 ga teng bo'lsa, uni ikki marta kattalashtirganda WIDTH=200, HEIGHT= 400 ga teng bo'lishi shart. Bu usul tasvir sifatidagi salbiy o'zgarishlarni bir oz yumshatadi. Shuningdek, ayni holatni tasvir o'lchamini kichraytirishda ham qo'llash maqsadga muvofiq.

• **Har doim ALT o'lchamini qo'llang.**

Har doim ALTdan so'ng tasvirga aloqador qo'shimcha axborotni kiritishga harakat qiling. Buning asosiy sababi, siz joylashtirgan tasvir ochilmagan taqdirda, Internetdan foydalanuvchi siz kiritgan tasvir izohini o'qiydi va minimum axborotga ega bo'ladi.

Tarmoqda tasvirlar haqida gapirilganda aniqlangan ma'lum bir turdagi fayllar haqida gap ketadi. U raqamli kamera yoki skaner yordamida olingan rasm, tasvir, hattoki, grafik formatda ifodalangan matn bo'lishi ham mumkin. Bu yerda faylning aniqlangan formati muhim rol o'ynaydi. Ko'pchilik matn muharrirlari (Notepad, SimpleText) grafik fayllarni o'zlarida tasvirlay (ko'ra, ifodalay) olmaydilar. Buning uchun maxsus tasvirlarni ko'rish dasturi, grafik muharrirlar mavjud.

Tasvirlarni web-sahifaga joylashtirish juda oddiy. Buning uchun elementi kerak bo'ladi. Shuni tushunish kerakki, rasm sahifaga joylashtirilmaydi, balki rasmga sahifadan turib murojaat (rasmga yo'l) ko'rsatiladi. Rasm fayli esa kompyuterning qattiq diskida yoki tarmoqda joylashgan bo'lishi mumkin.

Tarmoqda tasvirlarning asosan JPEG (Joint Photographic Experts Group «Jpeg» deb o'qiladi), GIF (Graphics Interchange Format «gif» yoki «djif» deb o'qiladi) va PNG (Portable Network Graphics «ping» deb o'qiladi) kabi formatlari qo'llaniladi. Bundan tashqari PCX, TIFF va PICT kabi formatlar ham internetda uchraydi. Agar foydalanuvchida boshqa formatdagi tasvirlar bo'lib, uni web-sahifaga joylashtirmoqchi bo'lsa, u holda uni yuqoridagi formatlardan birida ifodalab olishi maqsadga muvofiqdir.

 elementi Web sahifaga tasvir qo'yish imkoniyatini beradi. Bunda tasvirning URL adresini ko'rsatish kerak bo'ladi. Tasvir fayli hamma fayllar saqlanayotgan kataloglarning birida bo'ladi yoki Internetda joylashgan bo'lishi mumkin. elementi quyidagi ko'rinishda yoziladi:

```

```

Agar tasvir internetdan yuklanishi rejalashtirilgan bo'lsa, u holda uning to'liq adresini yozish kerak:

```

```

Agar tasvir o'zimizning katalogda joylashgan bo'lsa, uning adresini to'liq ko'rsatish shart emas:

```
  

```

alt atributida sahifaga qo'yilayotgan tasvir ekranga yuklash imkoniyati bo'lganda shu tasvir nima haqida ekanligini bildiruvchi matn joylashadi. Bu matn tasvir chiqishi kerak bo'lgan, ammo ba'zi bir sabablarga ko'ra ekranda ko'rinmay turgan joyda yoziladi. Masalan:

```

```

```
joylashadi"/>
```

Matn va tasvirlarni sahifaga joylashtirishda quyidagi atributlardan foydalaniladi:

Nomi	Vazifasi
align="top"	Matn tasvirning yuqori chegarasi bilan tenglashtirib yoziladi.
align="middle"	Matn tasvirning o'rtasi bilan tenglashtirib yoziladi.
align="bottom"	Matn tasvirning quyi chegarasi bilan tenglashtirib yoziladi.

Matnlarni o'ngdan va chapdan tekislash imkoniyati ham mavjud. Buning uchun **align** atributida **left** yoki **right** yozuvini yozish kerak.

Qo'yilayotgan tasvirning o'lchamini ham berish mumkin. Buning uchun **width** va **height** elementlari ishlatiladi. Berilayotgan o'lcham piksellarda ifodalanadi. masalan:

```

```



2.5. HTML tilida gipermatn joylashtirish teglari

Web-sahifaga tasvir joylashtirishni o'zlashtirib olganingizdan so'ng, endigi navbatda havolalar haqida tushuncha olishning ayni mavrudi. Havolalar tufayli internet hozirgi mavqeiga erishdi. Havolalar yordamida ikki va undan ortiq (cheksiz) web-sahifalarni bir-biriga bog'lashingiz mumkin. Havola rus tilida «ssilka» deb ataladi. Havolaga bosish orqali siz ikkinchi sahifani ochishingiz mumkin. Havolalar yordamida web-sahifangiz uchun menyu tuzishingiz mumkin. Havolalar sizning web sahifangizdan foydalanuvchini sahifa bo'ylab tom ma'noda sayohat qilishi imkonini beradi va unga to'g'ri yo'lni ko'rsatib turadi.

Havolalar

Havola - bu HTML hujjatning boshqa HTML hujjat bilan bog'lovchi qismi. Unga kursorni bosishingiz bilan ikkinchi HTML hujjat ekranda namoyon bo'ladi. Shuningdek, havoladan (agar siz yaratgan bir sahifada juda ham ko'p matn kiritilgan bo'lsa) bitta sahifa ichida boshqa HTML hujjatni ochmasdan turib ham foydalanish mumkin. Sizga maslahatimiz, agar bitta sahifaga joylashtirmoqchi bo'lgan matn juda ham ko'p bo'lsa, uni iloji boricha turli bo'laklarga bo'lgan holda, bir nechta HTML faylga joylashtiring. Shunday qilsangiz, web sahifadan foydalanish oson kechadi.

Navbatdagi sahifalar bilan tanishib chiqqach siz havolalarni ishlatish usullarini va havola tuzish uchun qanday HTML teglardan foydalanish zarurligini bilib olasiz.

Havolalar tuzish usullari

Havolalarni ta'sir doirasi va ularni qaysi sahifa bilan bog'lash vazifasini <A> ochuvchi hamda yopuvchi teglari orasida belgilanadi. Ushbu teglar orasida uning o'lchamlari va kalit so'z joylashtiriladi.

Ushbu teg orasida nimalarni joylashtirish mumkin? Deyarli hamma narsani. Bu matn, tasvir, jadval va hatto harakatlanuvchi tasvir ham bo'lishi mumkin. Ulardan qaysi birini joylashtirish o'zingizga havola.

Bu bo'limda oldimizga qo'ygan maqsadimiz havolalar tuzish. Maqsadga erishishning eng oson usuli barchasini oddiy misollardan boshlash. Shunga amal qilgan holda sizga havola tuzishning eng oddiy misolini keltiramiz:

Bu sahifada mening hayotim bilan bog'liq tasvirlar joylashgan.

Natija mana bunday ko'rinishda bo'ladi:

Bu sahifada mening hayotim bilan bog'liq tasvirlar joylashgan.

Bu sahifada mening hayotim bilan bog'liq tasvirlar joylashgan. so'zini bosishingiz bilan *pictures.html* sahifasi ekranda namoyon bo'ladi. <A> tegining qo'shimcha o'lchami bo'lgan HREF sizni *pictures.html* sahifasiga olib boradi.

Tasvirlarni havola sifatida namoyon etish uchun <A> va teglari orasida tegi yordamida tasvir nimani qo'yish va u joylashgan papkaga yo'lni ko'rsating.

Havolalar - <A> tegining qo'shimcha buyruqlari tasnifi

Boshqa HTML teglari kabi <A> tegining ham qo‘shimcha buyruqlari mavjud. Quyida ushbu buyruqlarning tasnifi keltirilgan.

HREF

Bu buyruq yordamida havola qilinishi lozim bo‘lgan HTML hujjat nomi va uning joylashgan joyi belgilanadi. Bu buyruq tegining SRC buyrug‘i kabi bir hil vazifani bajaradi. tegining SRC buyrug‘i yordamida tasvir nomi va u joylashgan papkaga yo‘l ko‘rsatilsa, HREF yordamida HTML hujjat nomi va u joylashgan papkaga yo‘l tasvirlanadi.

TARGET

Bu buyruq HTML hujjatni aynan o‘sha oynada yoki alohida oynada yoki qaysi freymda ochilishini belgilaydi. TARGET buyrug‘ining qo‘shimcha o‘lchamlari: **_blank** – brauzerda siz ishlab turgan oynadan tashqari yangi oynada havolada ko‘rsatilgan hujjatni ochish.

_top - web sahifada freymlar ishlatilgan bo‘lsa, havolada ko‘rsatilgan hujjatni faol oynada freymlar strukturasi buzgan holda ochadi. Ochilgan oynada freym alomatlari mavjud bo‘lmaydi (freymlar haqida batafsil ma‘lumotni «Freymlar bilan ishlash bo‘limi»ni o‘qishingiz mumkin).

_parent - havolada ko‘rsatilgan hujjatni avval ochilgan oynada ochiladi. **_self** - havola qaysi freym yoki freymdan holi sahifada ishlatilgan bo‘lsa aynan o‘sha sahifada ochadi.

Bu sahifada mening hayotim bilan bog‘liq tasvirlar joylashgan.
 Bu sahifada siz men haqimda qiziqarli ma‘lumotlarni o‘qishingiz mumkin.

Natija mana bunday ko‘rinishda bo‘ladi:

Bu sahifada mening hayotim bilan bog‘liq tasvirlar joylashgan. Sahifada siz men haqimda qiziqarli ma‘lumotlarni o‘qishingiz mumkin. Ushbu havolalardan biriga bosib, siz havolada belgilangan HTML hujjatdagi ma‘lumotlarni o‘qishingiz mumkin.

Havolalar tuzishning maxsus hollari

Havolalar yordamida nafaqat boshqa HTML hujjatni ochish, balki mavjud bir hujjat ichida ham ichki havolalarni qo‘llash mumkin. Agar siz yaratayotgan sahifadagi matn haddan tashqari ko‘p bo‘lsa va uni mavzu bo‘yicha bo‘lishi imkoni mavjud bo‘lmagan taqdirda siz sahifa ichiga ma‘lum kalit so‘zlardan foydalanib havolalar menyusini yaratishingiz mumkin. Bunday ichki havolalar menyusini sahifadagi matnni o‘qish jarayonini tezlashtiradi va o‘quvchi o‘zi uchun kerakli ma‘lumotni butun matn ichidan topib olishi oson kechadi.

Ichki havola <A> tegining NAME buyrug‘i yordamida amalga oshiriladi.

NAME

<A> tegi tarkibida yoziladi va HTML hujjatdagi matn tarkibidagi ma‘lum kalit so‘zga bog‘lanadi.

HTML hujjatning biror bir joyiga havola qilish uchun havola qilinayotgan nuqtani <A> tegining NAME buyrug‘i yordamida belgilab olish zarur. Masalan, sahifa yakunidan uning boshiga qaytish uchun HTML hujjat boshlanishida quyidagi kodni yozamiz:

Soʻngra sahifa yakunida matn yakun topgan joydan soʻng mana bu kodni teramiz:

Yuqoriga qaytish

Agar siz sahifani brauzerda ochib «Yuqoriga qaytish» havolasi ustida kursorni bossangiz, brauzer yana matn boshiga «#Top of Page» nuqtasiga qaytadi. "#Top of Page" havola murojat qilayotgan nuqta nomi. Teg kodlarini terishda hatolikka yoʻl qoʻysangiz, bu katta hatolik emas. Faqatgina havola ishlamaydi xolos.

Shuningdek ichki havolani boshqa HTML fayldagi matnning biror bir nuqtasiga ham bogʻlash mumkin. Aytaylik sayohat.HTML Frantsiya boʻlimiga murojat qilish lozim. Buning uchun quyidagi usuldagi kod yoziladi:

 Frantsiyaga sayohat

Yuqoridagi misoldan koʻrinib turganidek, avval HTML hujjat nomi (agar u boshqa papkada joylashgan boʻlsa, u holda faylga boʻlgan yoʻl, papka nomi) kiritiladi va # simvolidan soʻng havola murojat qilayotgan nuqta nomi yoziladi.

Navbatdagi misolda HTML teglar qanday yozilganiga eʼtibor bering:

<HTML>

<HEAD>

<TITLE>Yevropa boʻylab sayohat </TITLE>

</HEAD>

<BODY>

<P ALIGN="center"> Yevropa davlatlari boʻylab sayohatdan olgan taasurotlarim</P>

<P>Germaniya</P>

<P>Italiya</P>

<P>Frantsiya</P>

<P>Kirish</P>

Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn

Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn

<P>Germaniya </P>

<P>Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn</P>

<P>Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn</P>

<P>Yuqoriga</P>

<P>Italiya</P>

<P>Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn</P>

<P>Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn</P>

<P>Yuqoriga</P>

<P>Frantsiya</P>

<P>Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn</P>

<P>Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn Matn</P>

<P>Yuqoriga</P>

</BODY>

</HTML>

Gipermurojaatlar web-saytlar bo‘ylab harakatning asosi hisoblanadi. Murojaatni tanlaganda foydalanuvchi brauzer oynasiga yuklanuvchi yoki yordamchi dasturni ishga tushiruvchi qanaqadir URL bilan bog‘langan adresga «tushib» qoladi. Ba‘zan gipermurojaat natijasi e-mail yoki FTP serverga yo‘llanma beruvchi yangi web-sahifani ochilishiga olib keladi. Foydalanuvchi murojaatni tanlab olishi uchun web-dizayner uni yaratishi kerak.

Gipermurojaat yaratish uchun <a> (anchor, yakor) elementidan foydalaniladi. U o‘zida yo‘llanma beruvchi URL adresni ko‘rsatuvchi href atributi bilan to‘ldiriladi. Shuning uchun gipermurojaatni yaratish uchun URL adresni aniqlab olish kerak.

URL bu brauzer adreslar qatorida yoziladigan web-sahifa yoki xizmatning to‘liq adresidir. Ko‘pincha bu adres gipermurojaatning ustiga kursor olib kelinganda brauzerning pastki qatorida ko‘rsatiladi.

Gipermurojaatni yaratishda agar Internetdagi xizmat yoki adresdan foydalanmoqchi bo‘lsak, albatta uning to‘liq adresini ko‘rsatish shart. Agar o‘zimizda bor bo‘lgan web-sahifalardan gipermurojaatlar yaratmoqchi bo‘lsak ba‘zi bir ishni osonlashtiruvchi holatlar mavjud:

1. Faraz qilaylik biz <http://www.tdpu.uz> asosli kataloglar tuzilishini yaratmoqchimiz.

2. Bu katalogning ichida images, about, fakultet kabi kataloglar joylashtirmoqchimiz.

3. fizmat katalogida (fakultet katalogi ichida) kurs1.html sahifasini yaratdik.

4. Endi about katalogida joylashgan contact.html sahifasiga murojaat qo‘ymoqchimiz. Buning uchun quyidagicha yoziladi:

```
.././about/contact.html
```

5. Afsuski, bu yozuv <http://www.tdpu.uz/about/contact.html> yozuvi kabi qulay bo‘lmagan yozuvdir. Bunday uzun murojaatlarni yozganda turli xil xatolikka yo‘l qo‘yish mumkin. Buning oldini olish uchun <base> elementi ishlatiladi. Bu element yordamida asosiy baza sifatidagi katalog tanlanadi:

```
<head>
<base href="http://www.tdpu.uz" />
</head>
```

Bu element yordamida yuqoridagi murojaatni oddiygina qilib «about/contact.html» ko‘rinishida yozish mumkin. <base> elementi Internetdagi adreslarga qo‘yilgan murojaatlarga xalaqit bermaydi.

HTML hujjatda murojaatlarni yaratish uchun yakor (<a>,) elementidan foydalaniladi. Bu teglar murojaatni bildiruvchi so‘zni o‘z ichiga oladi. Murojaat quyidagi ko‘rinishda yoziladi:

```
<a href="Murojaat_manzili">Murojaat matni</a>
```

Demak, «Ma'lumotlar» so‘zi bilan murojaat about katalogidagi about.html sahifaga murojaat qo‘yish uchun quyidagicha yozish kerak bo‘ladi:

```
<a href="http://www.tdpu.uz/about/about.html">Ma'lumotlar </a>
<a href="about/about.html" >Ma'lumotlar </a>
```

Ichki murojaatlar joriy sahifaning o'ziga qo'yiladi. Masalan, biror fandagi biror mavzuga bag'ishlangan sahifa yaratayotgan bo'laylik. Mavzudagi mavjud hamma rejalarga murojaatlarni shunday qo'yish mumkinki, murojaatni tanlaganda rejadagi ma'lumotlar ekranda paydo bo'ladi. Bu ishni qilish oddiy. YA'ni o'tilishi kerak bo'lgan hamma abzatslarni belgilab chiqish, undan keyin o'sha belgilarga yakor (<a>,) ni qo'yib chiqish kerak. Misol qaraylik:

```
<h2 align="center">Reja </h2>
<p><a href="#q1">1. HTTP</a></p>
<p><a href="#q2">2. HTML</a></p>
```

Endi murojaat qilinayotgan abzatsni belgilab olamiz:

```
<p><a name=q1>HTTP bu...</p> yoki
<p><a name=q2>HTML bu...</p>
```

Bundan tashqari boshqa web-sahifadagi biror abzatsga ham murojaat o'rnatish mumkin:

```
<p>Bu <a href=" ../index.html#q1">savol</a><p>ga javob bering
Tasvirlarga ham gipermurojaatlarni qo'yish mumkin:
```

```
<a href="pedagog/index.html" ></a>
```

Tasvirlarning o'lchami har xil bo'lishi mumkin, har qanday tasvirga gipermurojaatlarni qo'yish mumkin.

URL shunday universal va moslashuvchanki undan tarmoqdagi har xil murojaatlar (e-mail, FTP, Gopher, Usenet yangiliklari, Telnet sessiyalari) dan foydalanish mumkin. Bu narsa bitta sahifaning o'zida turli xil murojaatlardan foydalanib, universal sayt yaratish imkoniyatini beradi. Masalan, dasturni yuklash bilan birga fodalalanuvchi o'sha dastur haqidagi yangiliklarni Usenet yangiliklar serveri murojaatidan foydalanib tanishishi mumkin.

Havolalarning qo'shimcha imkoniyatlari

Havolalarni nafaqat HTML fayl yoki hujjat ichidagi kalit so'zga, balki FTP serverlarda saqlanayotgan fayllar va E-mail (elektron pochta)ga ham o'rnatish imkoniyati mavjud.

FTP havolalar

FTP serverlarda saqlanadigan fayllarni yuklash uchun, FTP havolalar tuziladi. FTP serverlarda turli fayllar yoki dasturlar internet orqali yuklash uchun saqlanadi. Aytaylik siz o'z sahifangizda biror - bir qiziqarli dastur, tasvir yoki musiqa haqida ma'lumot yozdingiz va uni yuklab olish uchun fayl saqlanayotgan FTP server manzilini havolada ko'rsatmoqchisiz. U holda fayl joylashgan FTP serverga havola quyidagi tarzda ifodalanadi:

```
<A HREF="ftp://ftpsitenam.com/programs/program.exe"> Faylni yuklash</A>
```

FTP protokoli kompyuterlar o'rtasida fayllar almashishda qo'llaniladi. FTP murojaatlarini hosil qilish uchun kerakli server nomini bilish kifoya:

```
<a href="ftp://ftp.microsoft.com/">Microsoft
```

kompaniyasining FTP sayti
 Zip langan
programma

Ikkinchi misoldan ko‘rinib turibdiki, kerakli dasturni birato‘la yuklab olish mumkin. Bunda brauzer olinayotgan dasturni qattiq diskka haqiqatan ham saqlash kerak yoki kerakmasligini foydalanuvchidan so‘raydi.

Faylni yuklash

Faylni yuklash so‘zi ostida kursorni bosish bilan, brauzer dasturni foydalanuvchi kompyuteriga saqlash yoki ochish haqidagi axborotni chiqaradi.

E-MAIL havola

E-mail (Elektron pochta) manzillarni faollashtirish uchun havola tuzish mumkin. E-mail manzil ko‘rsatilgan havolani bosish bilan kompyuterga o‘rnatilgan elektron pochta bilan ishlovchi dastur faollashadi va unda havolada ko‘rsatilgan elektron pochta manzili ko‘rsatilgan yangi oyna hosil bo‘ladi va siz ko‘rsatilgan manzilga xat jo‘natishingiz mumkin.

E-mail manzilni havola sifatida belgilash uchun HREF buyrug‘i o‘rnida mailto: elektron pochta manzili yoziladi. Misol uchun:

Mening elektron pochtaga ushbu o‘quv qo‘llanma haqidagi fikr - mulohazalaringizni yozing!

Natija mana bunday ko‘rinishda bo‘ladi:

Mening elektron pochtaga ushbu o‘quv qo‘llanma haqidagi fikr-mulohazalaringizni yozing!

Gopher. Gopher matnlar yordamida iyerarxik ko‘rinishdagi ma‘lumotlarni brauzerda ifodalashdir. Hozirgi kunda Gopher saytlari juda ham kam miqdorda ishlatiladi. Ularga bo‘lgan murojaatlarni qo‘yish:

Kongress kutubxonasi

Usenet. Usenet yangiliklar guruhi – Internet konferensiyalar guruhidir. Uning nomi yangiliklar bo‘lgani bilan bu guruhning asosi har bir qatnashuvchi ishtirok etishi mumkin bo‘lgan savol-javoblar, muhokamalardan iboratdir. Bu guruhga sizning saytingizda qandaydir muammo muhokama qilinishi kerak bo‘lganda murojaat qilish mumkin. Yoki foydalanuvchilar tomonidan sizga shunaqa ko‘p savollar beriladiki, sizning ularga javob berishga imkoniyatingiz yo‘q, shunda bu guruhga murojaat qilishingiz mumkin, qayerdadir sizga yordam beradigan inson topiladi:

Starrek ga bagishlangan
Usenet konferensiyasi

Telnet. Telnet serveriga murojaat – dasturiy ta‘minot bilan ta‘minlangan uzoqdagi kompyuter bilan aloqa o‘rnatish uchun mo‘ljallangan murojaatdir. Telnet texnologiyasini birorta ham zamonaviy brauzerlar qo‘llamaydi, shuning uchun bu murojaatni tanlaganda qo‘shimcha dasturlardan foydalanishga to‘g‘ri keladi:

Telnet serveri bilan aloqa
o‘rnatish

Murojaatlarni yaratishda ba'zi bir kichik ayyorliklardan foydalanish ham mumkin. Masalan, murojaatni tanlaganda yangi oyna ochilishi uchun **target** atributidan foydalaniladi. Bu atribut qo'llanilganda natija yangi oynada hosil bo'ladi.

```
<a href="http://www.tdpu.uz" target="_blank"> TDPU sayti</a>
```

Agar saytdagi barcha murojaatlar yangi oynadan ochilishini hohlasangiz uni <base> elementi ichida joylashtirganingiz ma'qul:

```
<base href="http://www.tdpu.uz" target="_blank" />
```

Natijada sizning web-sahifangizdaga hamma gipermurojaatlar yangi oynada hosil bo'ladi.

Havolalar yozishning asosiy qoidalari

Quyida havola tuzishda bilishingiz lozim bo'lgan oddiy qoidalar yozilgan. Ularni diqqat bilan o'qib chiqing va eslab qolishga harakat qiling. Havola tuzishda bu qoidalarga amal qilsangiz, siz tuzgan havolalar hatoliklarsiz ishlaydi.

- **Harflarning bir hilda yozish va fayl yoki papka nomini to'g'ri ko'rsating.** Havola tuzilganda eng katta e'tiborni siz murojaat qilmoqchi bo'lgan faylga olib boruvchi yo'l aniq va to'g'ri ko'rsatilishi lozim. Bunda papka va fayl nomiga katta e'tibor qarating. Ularning nomida arzimagan hatolikka yo'l qo'ysangiz havola ishlamaydi. Aytaylik biror bir harfni tushib qolishi yoki bosh harf o'rniga kichik harf bilan yozilsa hatolik yuz berishi muqarrar. Shu bois, havola tuzishda aniqlik katta ahamiyatga ega.

- **Fayl va papka nomini kiritganda probel ishlatmang.** Web sahifa yaratish jarayonida fayl va papkalar nomi ikki va undan ortiq so'zdan iborat bo'lsa so'zlar orasida va oxirida probel ishlatmang. Probel ishlatilgan fayl yoki papkaga havola murojaat qilganda hatolik sodir bo'ladi.

- **Kalit so'zlarga havola tuzish.** Biror hujjat ichidagi kalit so'zga havola tuzganda, eng avvalo kalit so'zni <A> tegining NAME buyrug'i yordamida belgilab olish lozim. So'ngra shu tegning HREF buyrug'i orqali u joylashgan fayl nomi (agar kalit so'z boshqa faylda joylashgan bo'lsa) va kalit so'z manzilini ko'rsating HREF="fayl nomi.HTML# kalit so'z".

- **FTP server va E-MAIL uchun maxsus havola.** FTP serverda joylashgan faylga havola tuzganda 'FTP:/' bilan boshlanuvchi manzilni yozing. E-MAIL havola tuzganda esa "mailto: e-mail manzil"ni yozing.



2.6. Freymalar bilan ishlash imkoniyatlari

Freym - brauzer oynasidagi mustaqil to'g'ri to'rtburchak shaklidagi maydon bo'lib, u o'z ichiga boshqa alohida mustaqil HTML hujjatlarni birlashtira oladi.

Endi siz har bir oynasida alohida mustaqil HTML hujjatlarni ochish imkonini beruvchi freymnlarni tuzish usullari bilan tanishib chiqasiz. Freymlar brauzerdagi yangi oyna emas, balki bu alohida oyna yoki ramka bo'lib, u brauzerning ochilib turgan oynasi ichida aks ettiriladi. Bu bo'limda, bir freymdan ikkinchi freymnlarni bir-biriga havolalar yordamida bog'lash usullarini tahlil qilib chiqamiz.

Freymlar tuzish

Uzoq vaqtlar web-sahifalar yaratuvchilari butun sayt hattoki portal sahifalari bo'ylab harakatning umumiy sistemasini topishga harakat qildilar. Freymlar kirib kelishi bilan bog'liq eng asosiy muammo har doimdagidek brauzerlarning o'zaro tug'ri kelmasligi bo'ldi. Ko'pgina brauzerlar freymnlarni «tanimasdi». Freymlar spetsifikatsiyasi HTML standartlariga 1997 yilda yaratilgan HTML 3.2 versiyasidan boshlab qo'llanila boshlandi. U vaqtlarda freymlar Nestcape foydalanuvchilari orasida ommaviylashgan edi. HTML 4.0 dan boshlab brauzerlar freymnlarni «tushuna» boshladilar.

Freymlar bitta brauzer oynasiga bir nechta turli URL adreslarga ega bo'lgan mustaqil web-sahifalarni yuklash imkoniyatini beradi. Bu vazifani bajarish juda oddiy. Buning uchun **<body>** elementi **<frameset>** elementi bilan almashtiriladi. Bu konteynor sahifada freymlar hosil kiluvchi `<frame />` uchun mo'ljallangan. Freymlar ichida qaysi freymga chiqishi ko'rsatilgan alohida gipermurojaatlar bo'lishi mumkin.

Demak sahifada freymlar hosil qilish uchun ikkita **<frameset>** va **<frame />** elementlari ishlatiladi ekan:

```
<html>
<head>
</head>
<frameset>
</frameset>
</html>
```

`<frameset>` konteynori bir o'zi hech narsa qila olmaydi. Ekranda freymlar hosil bo'lishi uchun `<frame />` va `<noframes>` elementlari ishlatilishi kerak. Birinchisi ekranda freymlar hosil bo'lishi uchun ishlatilsa, ikkinchisi aksincha freymnlarni olib tashlash uchun ishlatiladi:

```
<frameset>
<noframes>
```

`<p>` Uzoq vaqtlar web-sahifalar yaratuvchilari butun sayt hattoki portal sahifalari boylab harakatning umumiy sistemasini topishga harakat qildilar. `</p>`

```
</noframes>
</frameset>
```

<noframes> elementi yordamida umuman freymlarsiz sahifa yaratish ham mumkin.

<frame /> elementiga misol:

```
<frameset>
<frame src= "index.html" />
<frame src= "viewer.html" />
</frameset>
```

<frame /> elementi biror-bir freymga itiyoriy web-sahifani chaqirish uchun ishlatiladi. Sahifaning URL adresi **src** atributi yordamida beriladi. Adreslar gipermurojaatlardagidek to'liq yoki qisqartirilgan holda berilishi mumkin.

Biror bir freymlarga yangi sahifalarning yuklanishi ikki bosqichdan iborat bo'ladi:

- ❖ Birinchidan, freymga aniq nom berilishi kerak;
- ❖ Ikkinchidan, ko'rsatilgan freymga yuklanishi uchun yuqorida berilgan nomdan foydalanish kerak.

Siz internetdan foydalanish jarayonida freymlar ishlatilgan web-sahifani ko'rgansiz. Bu sahifada uchta freymdan foydalanilgan.

Ko'rib turganingizdek bu sahifa uchta freymdan iborat ustunlarga bo'lingan. Chap ustunda joylashgan freymda navigatsion menyu, yuqoridan o'ngdagsida reklama va o'ngdan pastdagi freymda esa web sahifa asosiy matni o'rin oladi. Chap ustunda joylashgan havolalarga bosish bilan, havolalari hujjat o'ngdan pastda joylashgan freymda ochiladi.

Ushbu freymlar qo'llanilgan sahifaning umumiy tuzilishini ko'rib chiqsak:

<FRAMESET> - freymlar tuzilishini belgilovchi ochuvchi teg (freymset)

<FRAME> - birinchi freym

<FRAME> - ikkinchi freym

</FRAMESET> - freymsetni yopuvchi teg

Sxemada ko'rinib turganidek, har gal <FRAMESET> tegi yordamida freymlar tuzilishini belgilash lozim. Shundan so'ng bevosita freymlar <FRAME> tegiga navbat keladi. Oxirgi navbatda </FRAMESET> freymset yopuvchi teg yoziladi.

<frame /> elementi quyidagi parametrlarga ega:

- ❖ **Noresize**. Freymlar o'lchamini o'zgartirishni man qiladi.
- ❖ **Frameborder**. Faqat 1 yoki 0 qiymatlardan bittasini qabul qiladi. Agar 1 qiymatni qabul qilsa, freym chegarasi mavjud, 0 qiymat qabul qilsa, freym chegarasi yo'q.
- ❖ **Scrolling**. **yes**, **no** yoki **auto** qiymatlaridan faqat bittasini qabul qila oladi. **Yes** – prokrutka bor, **no** – prokrutka yo'q, **auto** – agar ehtiyoj bo'lsagina prokrutka qo'yiladi.
- ❖ **marginwidth**, **marginheight** – bu atributlar gorizont va vertikal maydonlarga mo'ljallangan.
- ❖ **Longdesc**. Mazkur freym haqidagi ma'lumotlar saqlanuvchi URL adresni o'zida saqlaydi.

Ichma-ich freymlar. Agar ekranda gorizont va vertikal freymlarni birdaniga hosil qilmoqchi bo'lsak, ichma-ich freymlardan foydalanishimiz mumkin.

<frameset> elementi ikkita **cols** va **rows** atributlariga ega. Ular yordamida <frameset> sahifalarni bir nechta qatorlar yoki ustunlardan iborat freymlarga ajratadi:

```
<frameset cols= "25%, 75%">
```

```
</frameset>
```

Natijada web-sahifa ikkita ustunli freymga ajraydi.

<frameset> elementi yordamida sahifada hohlaganicha ustun va qator freymlarini hosil qilish ham mumkin. Ularning o'lchamini doim ham foizlarda ifodalash shart emas. O'lchov birligi sifatida piksellarni ham olish mumkin :

```
<frameset rows= "10, 200, 300, 100">
```

Natijada to'rtta qator freymlari hosil bo'ladi.

Bundan tashqari o'lchamlarni ifodalashda «*» ni ham ishlatish mumkin. Bunda «*» ga qancha o'lcham qolsa hammasi tegishli bo'ladi:

```
<frameset cols= "200, 300, *">
```

Foizlarda ifodalangan o'lchamlar uchun ham «*» ni ishlatish mumkin:

```
<frameset rows= "25%, 35%, *">
```

Masalan tepada to'liq satrli freym, uning pastida ikki ustunga ajratilgan freymlarni hosil qilaylik. Birinchidan bizga ikki qatorli freym kerak bo'ladi. Ikkinchidan ikkinchi qator freymni ikki ustunga ajratish kerak:

```
<frameset rows= "100, *">
```

```
<frame src= "banner.html" scrolling= "no" noresize = "noresize" />
```

```
<frameset cols= "25%, 75%">
```

```
<frame src= "index.html" />
```

```
<frame src= "viewer.html" marginwidth= "5" marginheight= "5"/>
```

```
</frameset>
```

Freymlarning klassik variantini ko'rib o'tamiz: brauzer oynasi uchta qismdan tashkil topgan bo'lsin: logo.html, menu.html, asosiy.html. Ularning joylashish o'rni turlicha bo'lishi mumkin, masalan,

LOGO
ASOSIY
MENU

LOGO	
MENU	ASOSIY

L	M	ASOSIY
O	E	
G	N	
O	U	

LOGO	ASOSIY
MENU	

Quyidagicha variantdagi freymni yaratamiz:

LOGO
ASOSIY
MENU

```
<html>
```

```
<head>
```

```
</head>
```

```
<frameset rows="100,*,150">
```

```
<frame src="logo.html">
```

```
<frame src="asosiy.html">
<frame src="menu.html">
</frameset>
</html>
```

Koddagi rows="100,*,150" yozuvi, birinchi qatorning uzunligi 100 piksel, uchinchi qatorning uzunligi 150, ikkinchi qator esa qolgan maydonni egallashini anglatadi. rows o'rnida cols ishlatilsa brauzer oynasini qatorlarga emas, balki ustunlarga bo'lish kerakligini bildiradi.

Qator va ustunlarning o'lchamlari nafaqat piksellardan balki, foizlarda ham berilishi mumkin:

```
<frameset rows="15%,75%,10%">
```

Shuni yodda tutish kerakki, barcha o'lchamlarning yig'indisi 100% ga teng bo'lishi kerak.

Endi boshqa ko'rinishdagi freymlarni kodini yozamiz.

LOGO	
MENU	ASOSIY

Masalan,

```
<html>
<head>
</head>
<frameset rows="100,*">
<frame src="logo.html">
<frameset cols="150,*">
<frame src="menu.html">
<frame src="asosiy.html">
</frameset>
</frameset>
</html>
```

Oxirgi variantdagi freym kodini yozamiz:

LOGO	ASOSIY
MENU	

```
<html>
<head>
</head>
<frameset cols="100,*">
<frameset rows="100,*">
```

```
<frame src="logo.html">
<frame src="menu.html">
</frameset>
<frame src="asosiy.html">
</frameset>
</html>
```

Freymlarni yuqorida keltirilganlardan boshqa qo‘shimcha imkoniyatlari ham mavjud. Masalan, biror freymga sahifani yuklash yoki freymlarni yangi oynaga ochish.

Target atributi quyidagi qiymatlarni qabul qilishi mumkin:

- ❖ **_self**. Murojaat qo‘yilgan freym o‘zida hujjat ochilishini ta’minlaydi.
- ❖ **_top**. Bu qiymatni qo‘llab, shu oynaning o‘zida freymlarni o‘chirib hujjatni yuklash mumkin.

- ❖ **_blank**. Hujjatni yangi oynada ochilishini ta’minlaydi.

<iframe> elementi <frame /> elementi bilan to‘g‘ridan-to‘g‘ri bog‘langan, lekin <frameset> elementiga umuman aloqasi yo‘q. <iframe> elementi HTML hujjatda ichki mustaqil freymlar yaratish imkoniyatini beradi. <iframe> elementi ixtiyoriy sahifada <body> ning ichiga joylashishi mumkin. Uning vazifasi sahifada boshqa bir hujjatni <frame /> <frameset> da ko‘rsatgani kabi ko‘rsatishdir.

Freymlar - oddiy misol

Xullas, freymlar bilan ko‘rsatilgan oddiy misolni ko‘rib chiqamiz. Ishni boshlashdan oldin, oddiy HTML formatida ikkita hujjat yarating. Birinchi hujjatga "Frey-1" ikkinchisiga esa "Frey-2" deb yozing va ularni frem1.HTML va frem2.HTML deb nomlang. Bu ikkita fayllar bizning misolimizda ikki freymlar uchun asosiy manba bo‘lib xizmat qiladi. Endi esa HTML formatida keyingi hujjatni yarating.

```
<HTML>
<HEAD>
<TITLE>freymlar</TITLE>
</HEAD>
<FRAMESET COLS=200,*>
<FRAME NAME=left SRC=frame1.HTML>
<FRAME NAME=right SRC=frame2.HTML>
</FRAMESET>
<NOFRAMES>
Sizning brauzeringiz freymlarni tushunmaydi.
</NOFRAMES>
</HTML>
```

Demak siz ham o‘zingizning birinchi sahifangizni freymlar yordamida yaratdingiz. Endi shu yaratgan sahifangizni frameset.HTML deb nomlab oldin yaratilgan frame1.HTML va frame2.HTML fayllar turgan papkada saqlang va frameset.HTML faylini o‘z brauzeringizda oching.

Birinchi navbatda, e’tibor bergan bo‘lsangiz HTML formatidagi freymlardan iborat sahifada <BODY> tegi mavjud emas. Bu esa juda muhimdir. Agar siz <BODY> tegini ishlatsangiz u holda, hamma <FRAMESET> va <FRAME> teglari

brauzeringiz tomonidan o‘chiriladi. Undan tashqari <NOFRAMES> tegi shunday holatga mo‘ljallangan-ki, bu holatda sizning brauzeringiz freymlarni qo‘llab-quvvatlamaydi.



Savol va topshiriqlar:

1. HTML nima?
2. Teg nima?
3. Brauzer dasturi nima? Qanday brauzer dasturlarini bilasiz?
4. HTML hujjatning asosiy strukturasi tushuntirib bering?
5. HTML tilida rasm qo‘yish tegi va uning atributlarini tushuntiring?
6. HTML tilida jadval qo‘yish tegi va uning atributlarini tushuntiring?
7. HTML tilida havola qo‘yish tegi va uning atributlarini tushuntiring?
8. HTML tilida freym qo‘yish tegi va uning atributlarini tushuntiring?
9. HTML tilida tartiblangan ro‘yxat tuzishni aytib bering?
10. HTML tilida tartiblanmagan ro‘yxat tuzishni aytib bering?

III BOB. DREAMWEAVER DASTURIDA WEB-SAYTLAR YARATISH

Dreamweavering bu nafaqat interfeysning tashqi ko‘rinishi yangilangani, balki radikal texnologik yangilangan, gipermatnli belgilash tili imkoniyatlari kengaygan. Unda Web-sahifa maketlarini yaratish uchun yangi ashyolar paydo bo‘ldi. Jumladan, tizimga Flash ob‘yektlarni qo‘yish buyruqlari qo‘shilgan va Web-uzellar bilan boshqarish va turli jamoaviy ishlarni loyihalashda rivojlanishga imkon beradi. Endi Dreamweaver bilan batafsil tanishib chiqamiz.

Belgili kodlar bilan ishlash vositalari sezilarli darajada rivojlandi. Dastur ixtiyoriy murakkablikdagi dastlabki matn bilan ishlash uchun kuchli bazaga ega bo‘lgan gipermatnli dasturlashning to‘laqonli qobig‘idir. Asosiy imkoniyatlarini sanab chiqamiz:

1) Hujjatning oynasi endi har xil rejimlarda ishlashi mumkin. Design View (Режим планировки) rejimida, foydalanuvchi HTML sahifa maketini taxminan shunday ko‘rinishda ko‘rish mumkin. Code View (Режим разметки) rejimida esa hujjatning oynasida belgili matn sahifasining barcha deskriptorlari, senariylari va kommentariyalari tasvirlanadi. Dasturning bu rejimlari orasida kichik koordinatsiya mavjud. Kodlarni o‘zgartirish maket modifikatsiyasiga, vizual rejimda rejalashtirish yoki formatlashga o‘zgartirish kiritish imkonini beradi;

2) Dreamweaver tizimi qobig‘iga barcha zaruriy qurilmalarga ega bo‘lgan to‘laqonli kodlar tahrirlagichidan iborat dastur integratsiya qilingan. Unda HTML hujjatlarining dastlabki matnlari bilan ishlashni engillashtiruvchi vositalar: kiritilgan teglarning sarlavhalari, turli kodlarni rangli belgilash, tez qidirib topish vositalari aks etgan. Sintaktik noto‘g‘ri loyihalarni aniqlash, uzun qatorlarni avtomatik ko‘chirish, avtokorreksiya rejimi yo‘lga qo‘yilgan. Dastur vositalari yordamida nafaqat gipermatnli sahifalarni, balki matnli ko‘rinishga ega har qanday hujjatlarni, dasturlar, senariylar, oddiy matnlar, XML belgili hujjatlar va boshqalarni tahlil qilish mumkin.

3) Gipermatnlarni belgilash kodlari bilan ishlashni ma‘lumot paneli ancha soddalashtiradi. Bu dastur ichiga qurilgan interfaol ma‘lumotnoma gipermatnli belgilash tilining barcha standart loyihalarini va u bilan bog‘langan vositalarni: diskreptorlar, atributlar, ierarxik stildagi o‘ziga xosliklarning deklaratsiyalari (Cascading Style Sheet), Java Script tilining sintaktik loyihalarini qisqa va lo‘nda shaklda tavsiflaydi.

Tajribadan ma‘lumki xatto eng tajribali, HTML tilini chuqur biladigan Web-ustasi ham bu til standartiga kiruvchi yuzlab kodlar va atributlarni eslab qolishga qodir emas. Ma‘lumotlar paneli deskriptorni aniq va ravshan tavsiflab beradi, uning atributlari hamda ko‘rib chiqish dasturining turli versiyalari bilan mos tushishi haqida ma‘lumot beradi.

4) Java Script senariylar

Java Script dasturlash tilida yozilgan senariylardagi yashirin xatolarni aniqlash imkonini beradi. Bu vosita nafaqat hozirgi vaqtda web-dasturlash sohasida qo‘llaniladigan ko‘p sonli analizatorlar uchun qiyin bo‘lmagan, sintaksis xatolarni aniqlashga qodir. Kodlarni tuzatish vaqtida dastur ichiga qurilgan nazorat nuqtalari va

qadamba-qadam bajariladigan dastur ko'rsatmalari hamda o'zgaruvchilarning holatini nazorat mexanizmlariga ega.

Doimo sezilarli potentsialga ega bo'lgan vizual loyihalash tahrirlagichining uskunalari yana bir vosita bilan boyitildi. Bu gipermatnli sahifalarni rejalashtirish va yaratish uchun mo'ljallangan maxsus dastur ishchi rejimidir. U sahifani aks ettirishning maxsus rejimi (Layout View) va ikkita uskuna ko'rinishida amalga oshirilgan bo'lib, ular yordamida maket elementlarini chizish mumkin.

Tahrirlagichning oldingi versiyalari vektorli animatsiyalashgan tasvirlar sohasida birinchilar qatorida bo'lgan Flash dasturi bilan juda yaxshi o'zaro aloqada bo'lishi bilan ajralib turgan. Dasturning joriy versiyasiga animatsiyalashgan ob'yektlarni generatsiya qilishning ikkita asosiy buyrug'i integratsiya qilingan. Endi bevosita dastur oynasida interfaol tugma yoki «jonli matn» yaratish mumkin. Bunga o'xshash ob'yektlarni kiritishning texnik tomoni o'ta soddalashtirilgan va loyihalashtirishdan ko'ra ko'proq anketalashtirishga o'xshab ketadi. Foydalanuvchidan Flash – texnologiya sohasida chuqur bilimlar talab qilinmaydi, u faqat muloqot oynasi sohasini to'ldirishi kerak. Zamonaviy dasturlarning ko'pchiligi animatsiyalashgan Flash – grafikani aniq aks ettiradi. Bu tipdagi ob'yektlar a'lo darajada masshtablanishi bilan ajralib turadi va tarmoq bo'ylab uzatishda ham diskda saqlashda juda kam resurs talab qiladi. Dasturga kiritilgan yangiliklarni qisqacha sanab o'tamiz:

- iyerarxik stildagi o'ziga xosliklar (CSS) bilan ishlash vositalari sezilarli darajada takomillashtirilgan.

- Adobe firmasi tomonidan ishlab chiqilgan Fire woks grafik muharriri bilan Dreamweaver dasturining integratsiyasi yanada rivojlantirildi.

- yangi Assets (resurslar) palitrasi yaratildi. U dasturda bitta WEB – tugunning bir necha sahifasi tomonidan birgalikda ishlatiladigan turli xildagi ma'lumotlarning omborxonasi vazifasini bajaradi. Bular tasvirlar, animatsiya roliklari, ranglar, senariylar, kutubxonalar va boshqalar bo'lishi mumkin. Saqlanayotgan resurslarni turli sahifalarning talab qilinadigan sohalarida joyini o'zgartirish (yoki surish) yo'li bilan joylashtirish mumkin.

Site Report (Отчет о сайте) buyrug'i loyiha holati haqidagi batafsil hisobotni hosil qiladi. Hisobot o'ziga ko'p sonli giperbog'lanishli hujjatlarni biriktiradi:

- tahrirlagichga versiyalarni va dasturiy ta'minotni yangilashni boshqarishni nazorat qilishning etakchi dasturiy vositasi bo'lgan Microsoft Source Safe paketli integratsiya vositalari qo'shilgan. Bu narsa Dreamweaver tizimini Internet uchun murakkab loyihalarni ishlab chiquvchi jamoalar uchun asosiy uskuna sifatida qo'llanilishiga asos bo'ladi.

- dasturning bu versiyasi WEBDav bayonnomasini to'laqonli qo'llab-quvvatlaydi. Bu versiyalarni boshqarish va taqsimlangan ishni tashkil qilish bo'yicha kengaytirilgan imkoniyatlarga ega bo'lgan klassik bayonnomasidir.

- interfeysni standartlashtirish va sayqal berish bo'yicha katta ish qilindi. Adobe firmasi ko'pgina dasturlarning tashqi ko'rinishi noturg'un palitralarning yagona kontseptsiyasiga asoslanadi. Muharrirning oxirgi versiyasida palitralarning ko'rinishi standartlashtirilgan. Ularning deyarli hammasi tarkibiy majmuaviy palitralarga

birlashish imkoniyatiga ega bo'ldilar. Oldingi versiyalarning ro'yxatga olishdagi ba'zi kamchiliklar bartaraf etildi.

- klaviatura kombinatsiyalari bir shaklga keltirildi. Ular web-dizayn uchun Adobe firmasi tomonidan ishlab chiqilgan, barcha dasturlash vositalari uchun umumiy bo'lgan ichki standartga moslashtirilgan. Ichiga qurilgan boshqaruv tizimi yordamida tugmalarning har qanday tuzilishini o'zgartirish mumkin.



Savol va topshiriqlar:

1. Dreamweaver dasturi nima maqsadda qo'llaniladi?
2. Dasturning qanday yangi imkoniyatlari mavjud?
3. Dastur avvalgi web-muharrirlaridan nimasi bilan farq qiladi?

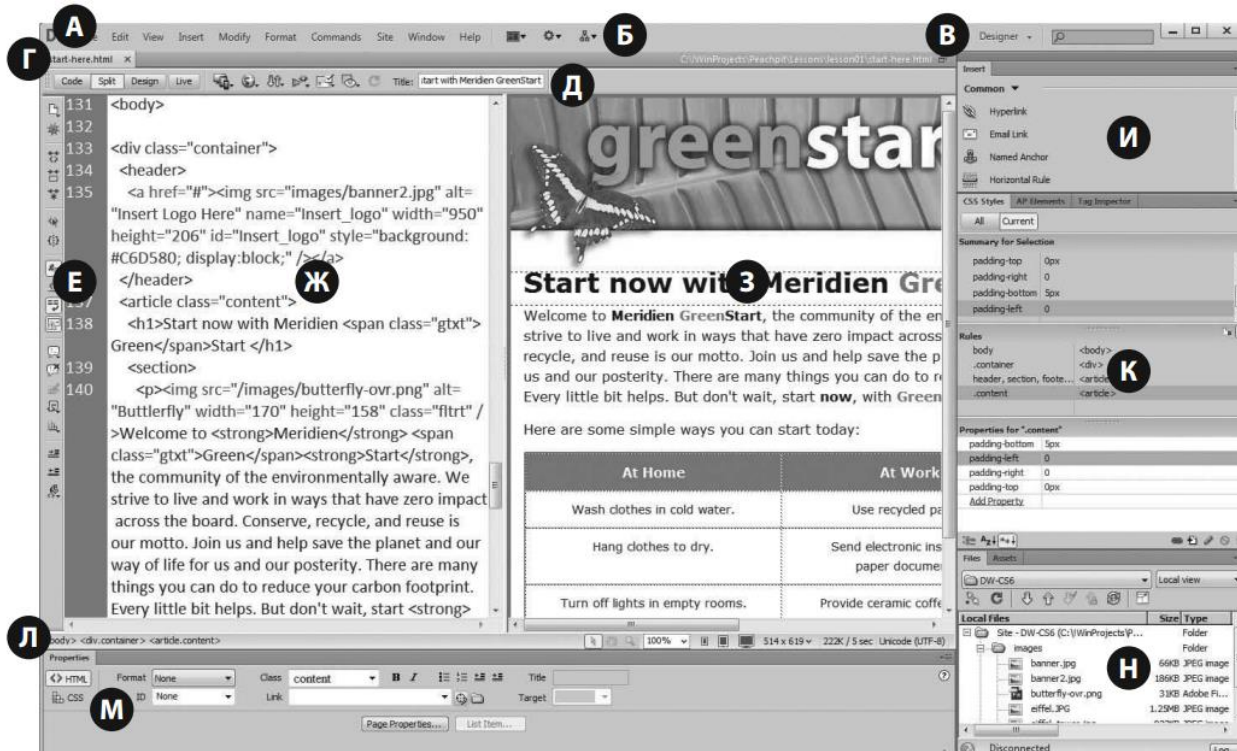


3.1. Dreamweaver dasturning asosiy oynalari, menyusi va komponentlari

Interfeys deganda informatikada dasturdagi turli komponentlar va foydalanuvchi o'rtasidagi o'zaro aloqani ta'minlovchi vositalar va qoidalar yig'indisi tushuniladi. Dreamweaver paketi taklif etadigan muomala usuli zamonaviy foydalanuvchi uchun odatiy bo'lib qolgan grafik oynali interfeys printsiplariga asoslanadi. Dreamweaver paketi o'sha sinfdagi boshqa dasturlar bilan taqqoslaganda ancha aniq, moslashuvchan va shu bilan bir paytda kuchli interfeys hisoblanadi. Palitralar, ko'p sonli klaviatura kombinatsiyalari va menyu qatori uning asosiy vositalari hisoblanadi. Barcha operatsiyalar to'xtatilgach ekranda qoladigan oynalarni palitralari (Palette) yoki panellar (panel) deb atash qabul qilingan. Ekranning maxsus ajratilgan pozitsiyalarida emas, balki ixtiyoriy joyida joylasha oladigan oynalar ko'chib yuruvchi palitralar (floating plette) deb ataladi. Dreamweaver dasturi muloqot oynalarining ko'pchiligi ko'chib yuruvchi palitralar ko'rinishida ishlaydi. Palitralar va panellar orasidagi kichik farqlar dasturni tavsiflash uchun sezilarli ahamiyatga ega emas. Shuning uchun kelgusida bu atamalarni sinonimlar sifatida qo'llaymiz.

Dreamweaver — gipermatnlarni belgilash tili (HTML) uchun nisbatan keng tarqalgan muharrirlardan biridir. Uning mashhurligi uchun jiddiy asos mavjud. Dastur o'zida kodlarni tahrirlash va dizayn uchun mo'ljallangan ko'plab uskunalarni taqdim qiladi. Kod bilan shug'ullanuvchilarga Code (Код) taqdim qilish oynasi muhitida joylashgan turli yangilanishlar, yaratuvchilarga esa dasturning ASP, PHP, Cold Fusion va JavaScriptsingari tillarni qo'llab-quvvatlashlari yoqadi. Loyihalovchilarga esa «nimani ko'rsangiz, shuni olasiz» (WYSIWYG, what you see is what you get) kontseptsiyasiga mos ravishda matn va grafika aniq joyda paydo bo'ladi. Bunda brauzerlarda oldindan ko'rishni amalga oshirish uchun ortiqcha vaqt sarlab o'tirilmaydi. Endigina ish boshlagan veb-yaratuvchilarga esa dasturning kuchli interfeysi qo'llashda osonligi bilan muhim. Siz qanday toifa foydalanuvchilarga

mansublingizdan qat'iy nazar Dreamweaver ilovasi bilan murosaga kelish talab qilinmaydi⁸.



Dreamweaver dasturi interfeysi foydalanuvchi tomonidan sozlanadigan ko'plab panellar va uskunalar tarkib topgan. Ular bilan tanishib chiqish uchun ko'p vaqt ketmaydi:

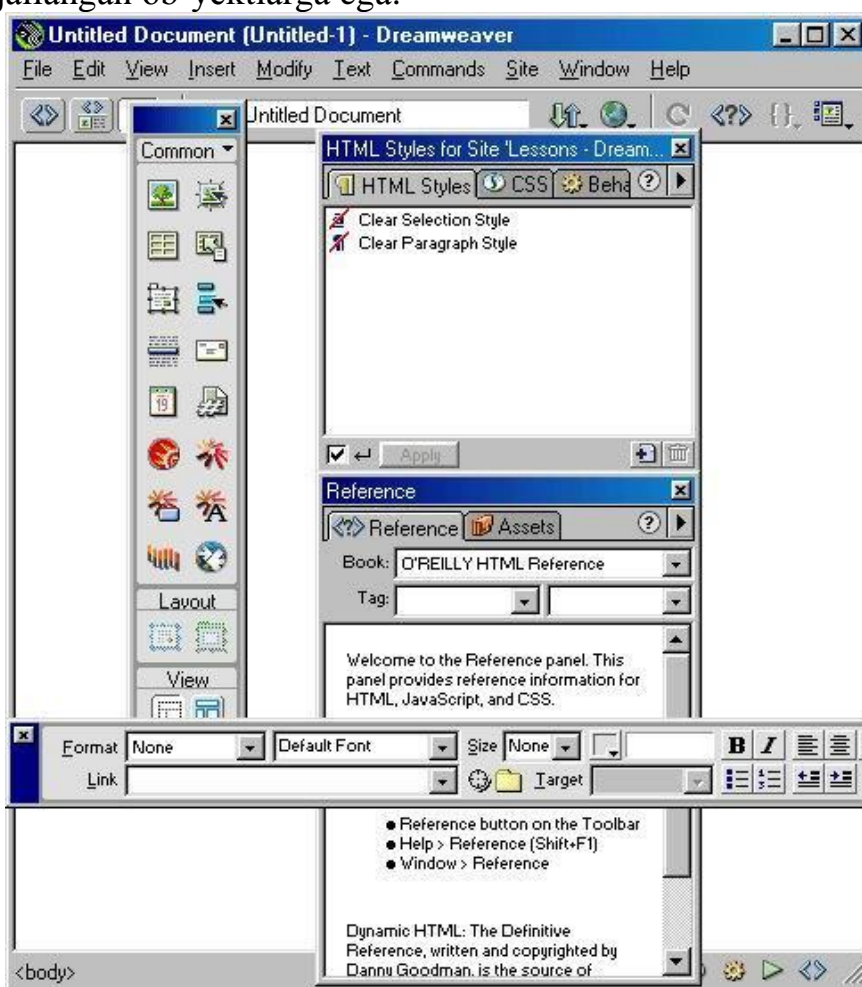
- A. Menu qatori,
- B. Ilovalar paneli,
- B. Ishchi fazo menyusi,
- Г. Hujjat sarlavhasi,
- Д. Document uskunalar paneli,
- E. Coding uskunalar paneli,
- Ж. Code taqdim etilishi,
- 3. Design taqdim etilishi,
- И. Insert paneli,
- K. CSS Styles paneli,
- Л. Teglar jamlanmasi,
- M. Xususiyatlar paneli,
- H. Files paneli.

Agar siz dasturni bunday ko'p imkoniyatlari uni judayam og'irlashtirishi kerak, sekin va boshqaruv yomon bo'ladi deb o'ylasangiz adashasiz. Dreamweaver ilovasida panellarni ko'chirish orqali o'z xohishingiza ko'ra osongina sozlab olish mumkin. Ularni aks ettirish yoki bekitish va har qanday kombinatsiyada joylashtirib ajoyib ishchi muhitni yaratish mumkin.

⁸ Adobe Dreamweaver CS6 Professional. Официальный учебный курс; [пер. с англ. М. А. Райтмана]. – М. : Эксмо, 2013. – 496 с. : ил. + 1 CD-Rom. – (Официальный учебный курс).

Deyarli barcha palitralar birlashish xususiyatiga ega. Bu har xil palitralarni bitta muloqot oynasida joylashtirish mumkinligini bildiradi. Birlashtirish amali turli mavzu yoki buyruqlarga tegishli boshqaruv vositalariga ega bo'lgan ko'p funktsiyali terma palitralarni olish imkoniyatini beradi. Birlashish xususiyatiga ega bo'lgan palitralarni (ular dasturda ko'pchilikni tashqil etadi) modulli (dockable) palitralar deb ataymiz. Bir nechta alohida modullardan iborat murakkab palitralar terma (tabbed) palitralar deb ataladi.

8.1-rasmda to'rtta ochiq muloqot oynali dasturning tashqi ko'rinishi ko'rsatilgan. Boshqaruvning eng talab qilinuvchi vositasi so'zsiz, Properties inspector (xossalar nazoratchisi) palitrasi hisoblanadi. Uning yordamida HTML sahifalarida ob'yektlarni formatlash va HTML sahifasidagi ob'yektlarni tahrirlash bo'yicha ko'plab amallar bajariladi. Behaviores (rejimlar) palitrasi – bu modul ko'chib yuruvchi palitraga misol bo'ladi. Dreamweaver dasturi turli loyiha masalalarini hal qilishga mo'ljallangan ob'yektlarga ega.



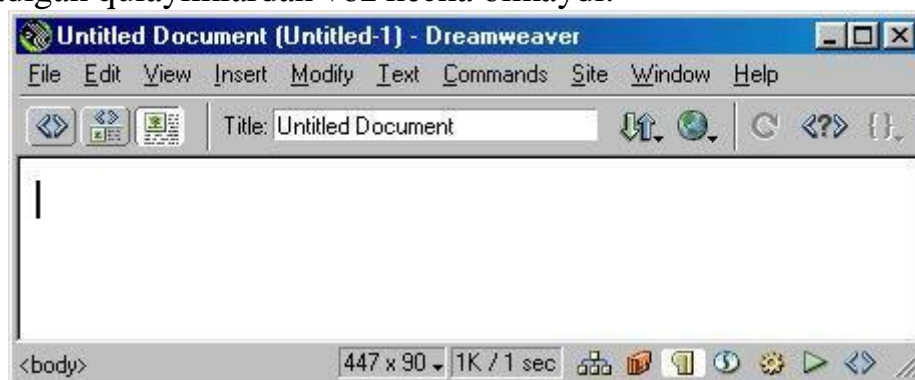
8.1-rasm. Dastur oynasining tashqi ko'rinishi.

Rasmda ikkita oddiy modulli Layers va History palitralardan iborat bo'lgan terma palitraga misol keltirilgan. Objekts palitrasi tashqi ob'yektlar: rasmlar, jadvallar, qatlamlar, freymalar va boshqa ob'yektlarni qo'yish uchun mo'ljallangan. Rasmda ko'rsatilgan boshqa barcha boshqaruv elementlari hujjat oynasining ajralmas tashkil etuvchi qismi hisoblanadi. Texnik tizimlarning ishonchli ishlashi, dasturiy mahsulotlarni interfeyslarining qulayligi asosida to'g'ri tashkil qilingan.

Dreamweaver paketi formatlash va tahrirlashning deyarli barcha asosiy buyruqlari uchun amalga oshirishning bir necha xil usullarini taklif etadi. Keng formatli ekranlarning egalari barcha buyruqlarni palitralar va panellar yordamida bajarishlari mumkin. Bundan mahrum foydalanuvchilarga esa dastur menyu qatori buyruqlarini va klaviatura tugmalari kombinatsiyalaridan foydalanishadi.

Hujjat oynasi

Hujjat oynasi - bu gipermatnli hujjatlar yaratiladigan ish maydoni hisoblanadi (8.2-rasm). U HTML sahifasi o'z ichiga olgan matnlar, rasmlar, jadvallar, shakllar va boshqa ob'yektlarni saqlaydi. Dreamweaver, WYSIWYS sinfiga kiruvchi tahrirlagich dastur hisoblanadi. Bu uzun atama inglizcha What you see is what you get (siz nimani ko'rayotgan bo'lsangiz shuni olasiz) iborasining bosh harflaridan tuzilgan. Bu narsa HTML sahifasining hujjat oynasida aks ettirilishi uning yanada ommaviyrog ko'rib chiqish dasturlari – Microsoft Internet Explorer va Netscape Navigator brauzerlarida ifodalanishidan uncha farq qilmasligini bildiradi. HTML kodlari bilan to'g'ridan-to'g'ri ishlash yuqori professionallik hisoblanishiga qaramay, hatto eng tajribali WEB –dizayner ham grafik interfeys va WYSIWYG rejimida hujjat yaratishda beradigan qulayliklardan voz kecha olmaydi.



8.2-rasm. Hujjat oynasi.

Ko'pchilik Windows ilovalaridan farqli ularoq Dreamweaver dasturi bitta dastur oynasida bir necha hujjatlar bilan ishlay olmaydi. Tanlab olingan parametrlarni o'rnatishdan kelib chiqib yangi hujjat dasturning alohida oynasiga joylashtiriladi yoki joriy oynada avval ochilgan hujjat o'rnini egallaydi. Bunday ish rejimida dastur oynasi va hujjat oynasi orasida deyarli farq yo'q. Shuning uchun ham biz kelgusida bu nomlarni sinonimlar sifatida ishlatamiz.

Ish rejimlari

WYSIWYG rejimida gipermatnli hujjat yaratish oddiy matn muharriri bilan ishlashdan ko'p farq qilmaydi.

Matnni kiritish va formatlashning bazaviy amallari, ob'yektlarni joylashtirish buyruqlari, navigatsiya va tanlash texnikasining barcha amallari Dreamweaver dasturida so'nggi bir necha yil davomida vizual muhitda ustunlik qilish usulida amalga oshirib kelinmoqda. Foydalanuvchi loyihalash echimlarini HTML tili vositalari tomonidan qanday yo'lga qo'yilishi haqida o'ylab o'tirmasdan tahrirlash va formatlashning odatdagi vositalaridan foydalanib ishlashi mumkin. Dreamweaver paketi dastlabki matnni avtomatik ravishda yaratadi, unda hujjatning ma'lumot bilan to'ldirilishi va ro'yxatga olinishi bilan gipermatnni belgilash tilining kodlari yoziladi.

Yaratuvchanlik ishining katta qismi WYSIWYG rejimida bajarilishi mumkinligiga qaramay ba'zan to'g'ridan-to'g'ri kodlashga murojaat qilishga to'g'ri keladi. Buning bir nechta sababi bor. Asosan, bu belgilash kodlarini juda ehtiyotkorlik bilan sozlash zarurligidir. Bundan tashqari HTML tili standartida yuzdan ortiq deskriptorlar haqida eslatib o'tiladi, ularning har biri bir nechta atributlar va modifikatorlarga ega bo'lishi mumkin. Gipermatnli belgilashning eng ko'p talab qilinadigan vositalari Dreamweaver tizimi interfeysi doirasida joylashtirilgan.

Design taqdim etilishi. Design (Дизайн) rejimida taqdim qilishda sahifani tahrir qilish interfeysi brauzer dasturida ko'rinadigan holatga o'xshash aks etadi. Design (Дизайн) rejimiga o'tish uchun Document (Документ) uskunalar panelidan Design (Дизайн) nomli tugmasini bosib.

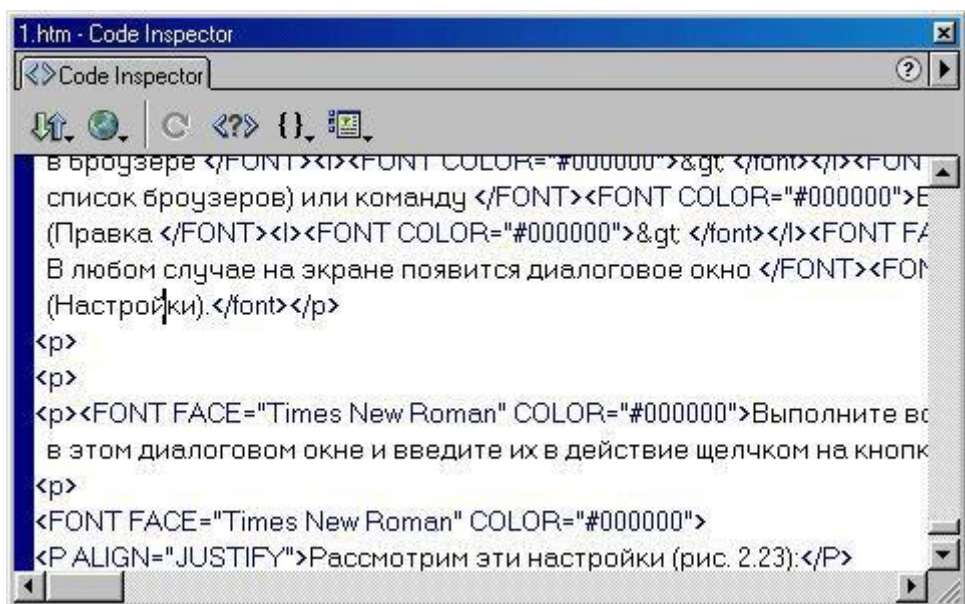
Code (Код) rejimi — bu Dreamweaver dasturining ishchi muhiti bo'lib, to'raligicha HTML-kodni tahrirlashni aks ettirib beradi. Code (Код) rejimini faollashtirish uchun Document (Документ) uskunalar panelidan Code (Код) nomli tugmasini bosib.

Split (Разделение) rejimi o'zida murakkab ishchi muhitini taqdim qiladi. Uning yordamida bir vaqtning o'zida ham bezak berishga va ham kodga ishlov berish imkoniyati paydo bo'ladi. Bir oynada qilingan o'zgarishlar boshqasiga o'sha zahoti aks etadi. Split (Разделение) rejimini o'chirish yoki taqdim qilish uchun Document (Документ) uskunalar panelidagi Split (Разделение) nomli tugmani bosib. Yangi yassi panelli monitorlar afzalliklarini qo'llash uchun Dreamweaver dasturi odatda ishchi fazoni vertikal bo'yicha bo'lib beradi⁹.

Dreamweaver dasturining hamma versiyalari, birinchisidan boshlab, foydalanuvchiga belgilash kodlarini to'g'ridan-to'g'ri tahrirlash imkonini beradi. Bu ish uchun qanday vositalar qo'llaniladi?

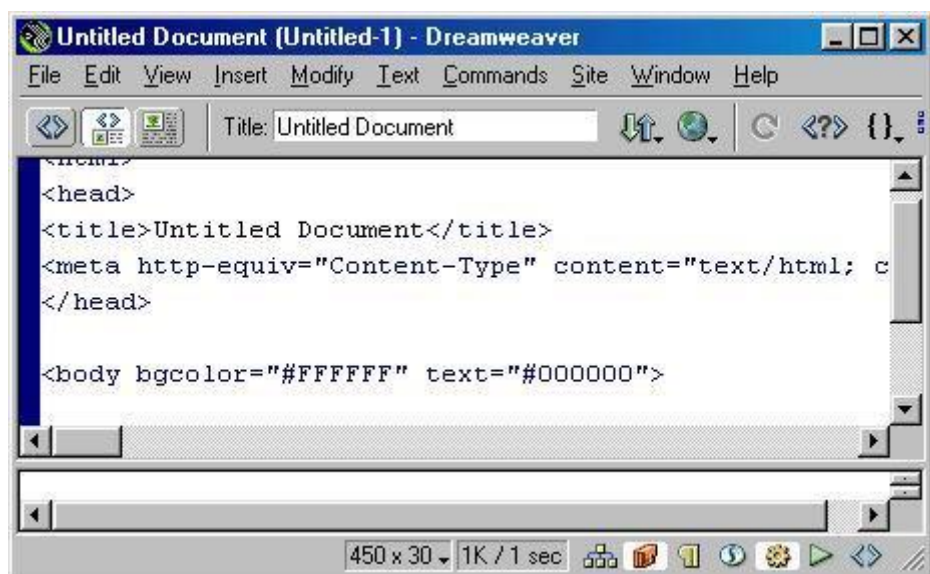
Birinchidan, bu maxsus ichiga qurilgan Code Inspector (kodlar inspektori) tahrirlagich bo'lib, u Dreamweaver dasturini eng birinchi versiyalarida ham mavjud edi va bu oxirgi versiyalarda vaqt davomida faqatgina dizayniga o'zgartirishlar kiritildi. Ikkinchidan, foydalanuvchi HTML tiliga tashqi tahrirlagichlarni ulash imkoniyatiga ega. Turli dasturlar: universal vazifali (masalan, Bloknote) oddiy matn tahrirlagichlaridan boshlab, to murakkab gipermatnli hujjatlar bilan ishlashga mo'ljallangan (Allaire Home Style) yuqori darajali dastur vositalarigacha bu rolni bajarishi mumkin. Dasturda juda qulay yangilik – hujjat oynasining ish rejimi paydo bo'ladi. Hujjat oynasi uchta rejim yoki ko'rinishlardan birida bo'lishi mumkin.

⁹ Adobe Dreamweaver CS6 Professional. Официальный учебный курс; [пер. с англ. М. А. Райтмана]. – М. : Эксмо, 2013. – 496 с. : ил. + 1 CD-Rom. – (Официальный учебный курс).



8.3-рasm. Hujjatning belgili rejimida ko‘rinishi.

8.3-рasm. Belgilash rejimidagi hujjat oynasi Code and View (belgilash va rejalashtirish rejimi). Bu aralash rejim bo‘lib, unda hujjat oynasi ikkita qismga bo‘linadi. Bir qismda gipermatnli belgilash kodlari aks etadi, boshqasida dastur sahifani barcha ob‘yektlar va formatlash atributlari bilan rejalashtirishni ko‘rsatadi. 8.4- rasmda bo‘sh sahifali ajratilgan hujjat oynasi ko‘rsatilgan.



8.4 - rasm. Bo‘sh sahifali hujjatning aralash rejimda ko‘rinishi.

Ish rejimini o‘zgartirishning bir necha usuli bor. Birinchidan, bu menyu qatori buyruqlari: View > Code, View – Design, View – Code and Design. Ikkinchidan, uskunalar panelida bir rejimidan boshqasiga tez o‘tish imkonini beruvchi tugmalar bor. Va nihoyat, Ctrl + Tab tugmalar kombinatsiyasi yordamida rejalashtirish rejimidan belgilash rejimiga va aksincha, yoki ajratilgan oynaning bir qismidan boshqasiga o‘tish mumkin.

Ajratilgan hujjat oynasi qismlarining o‘lchamlarini o‘zgartirish mumkin. Buning uchun ko‘rsatkich yordamida ajratuvchi chiziqni ko‘chirish kerak bo‘ladi.

Oxirgi paytlarda dasturchilar orasida «dastlabki matn» atamasi paydo bo'ldi. Yuqori saviyali dasturlash tilida yozilgan dastur ishlab chiqishning dastlabki bosqichlarida shunday deb ataladi. Bu atama gipermatnli hujjatning kodli qismi va ko'rinadigan obrazi o'rtasida mavjud bo'lgan munosabatlardan tashkil topadi. Hujjatning HTML tilida yozilishini dastlabki matn deb ataymiz.

Menyu qatori

Menyu qatori – bu Windowsning ixtiyoriy ilovasining zaruriy atributidir. U buyruqlar yordamida, muloqot oynalariga, tugmalarga yo'l ochadi, ular yordamida hujjat yaratish, dasturni sozlash, vizuallashtirish jarayonini boshqarish kabi dasturning barcha funktsional imkoniyatlarini o'z ichiga olgan samarali ishlar bajariladi. Menyu qatorining har bir bandi vazifasi jihatidan bir-biriga yaqin kiritilgan menyu va kichik menyular ierarxiasini hosil qiluvchi buyruqlar guruhini birlashtiradi. Bu bandlarning qisqacha tavsifini keltiramiz.

- File menyusi (Файл) fayllar bilan bog'liq buyruqlarni, gipermatnli hujjatlar shablonlari bilan ishlash, eksport qilish, import qilish va brauzer dasturlarida ko'rish buyruqlarini o'z ichiga oladi.
- Edit (Правка) menyusiga tahrirlash va qidirish bilan bog'liq buyruqlar kiradi.
- View (Вид) menyusi interfeysning yordamchi elementlarini va gipermatnli hujjat ob'yektlarining ko'rinishlari ustida ishlovchi buyruqlarni o'z ichiga oladi.
- Insert (Вставка) menyusiga har-xil ob'yektlarni import qilish buyruqlari kiradi: rasmlar, jadvallar, shakllar, qatlamlar, maxsus belgilar va boshqa ob'yektlarni import qilish.
- Modify (ИЗМЕНИТЬ) menyusi o'z ichiga gipermatnli hujjat ob'yektlari atributlarini o'zgartirish buyruqlarini oladi.
- Text (Текст) menyusi matnli fragmentlarni hosil qilish va ularni formatlash bilan bog'liq buyruqlarni o'z ichiga oladi.
- Commands (КОМАНДЫ) menyusi gipermatnli hujjatlarni avtomatik qayta ishlash buyruqlarini o'z ichiga oladi.
- Menyu qatorining Site (Сайт) bo'limi sayt bilan ishlash vositalari va gipermatnli hujjatlar bilan bog'liq murakkab tizimlar vositalarini o'z ichiga olgan.
- Window (Окно) menyusi dasturga tegishli palitralar, panellar va muloqot oynalarini o'rnatish va aksincha olib tashlash bilan bog'liq buyruqlarni o'z ichiga oladi.
- Help (Справка) menyusida dasturdan foydalanish haqidagi interfaol ma'lumotlarni chaqirish, elektron qo'llanmani chaqirish kabi buyruqlar joylashgan.

Uskunalar paneli

Uskunalar paneli dasturning asosiy dispetcherlik pulti vazifasini bajaradi (8.5 - rasm). U eng ko'p ishlatiladigan buyruqlar va tahrirlash resurslarini chaqirish tugmalarini birlashtiradi. Bu ixchamgina panel kiritilgan menyu va kichik menyular tizimi navigatsiyasi bilan bog'liq sarf-harajatlarni anchagina qisqartiradi.



8.5-rasm. Uskunalar paneli.

Unda joylashgan barcha boshqaruv elementlarini sanab o‘tamiz (chapdan o‘ngga):

- Code View (Режим разметки) - gipermatnli hujjatning faqat belgili kodlari tasvirlanadi (dizayni tasvirlanmaydi).
- Code and Design View (Режим разметки и планировки). Tugmasi bosilganda hujjat oynasi ikki qismga ajraladi: birinchi oynada belgili kodlar beriladi, ikkinchi oynada esa gipermatnli sahifaning dizayni, ya’ni tashqi ko‘rinishi beriladi.
- Design View (Режим планировки). Dasturda ishlashning vizual – WYSIWYG rejimiga o‘tish tugmasi.
- Title (Заголовок) – bu matnli maydon hujjatning sarlavhasini kiritishga mo‘ljallangan.
- File Management (Управление файлами) – tarmoqda ma’lumot almashish uchun xizmat qiladigan fayllar bilan ishlash buyruqlaridan tarkib topgan menyuni ochish tugmasi.
- Preview/Debug in Browser (Просмотр/отладка в браузере) – tugmasi gipermatnli hujjatni ko‘rish va undagi Java Script kodlarini tanlangan brauzerda ishlash yoki ishlamasligini tekshiruvchi buyruqlaridan iborat menyuni ochadi.
- Refresh Design View (Обновить страницу) - bu buyruq berilgandan so‘ng, hujjatning belgili matnlarida bo‘lgan barcha o‘zgarishlar loyihalash rejimiga uzatiladi. Ana shu vaqtda kodlarni interpretatsiyasi va sintaksik tekshiruv boshlanadi. Xato fragmentlari sariq rangda ko‘rsatiladi. To‘g‘rilar esa sahifada tasvirlanadi.
- Reference (Справка) – tugmasi elektron ma’lumotli qo‘llanmani ekranga chiqarib beradi. Bu qo‘llanmada gipermatnli belgilash tili deskriptorlari ifodalangan.
- Code Navigation (Навигация по кодам) – tugmasi Java Script tilida yozilgan belgili kodlarni tuzatishni boshqaruv buyruqlaridan tarkib topgan menyuni ochadi.
- View Options (Параметры отображения) – tugmasi hujjat oynasining qo‘shimcha elementlari vizualizatsiyasini boshqaruvchi buyruqlar menyusini ochadi. Bu menyuga asosiy menyu qatoridagi View menyusining bir necha buyruqlari kiradi.

Sahifalar o‘lchamlarini belgilash

Virtual to‘plam hech qanday tartib-qoidalarga va standartlarga bo‘ysunmasdan, tarqoq holda rivojlanadi. Bu esa ishlab chiquvchilar va dasturchilar uchun ko‘p sonli qiyinchiliklar manbai bo‘lishi «Internet almashuvi»ning juda tez o‘sishi sabablaridan biridir. Aslida sahifaga yoki saytni ishlayotganda muallif o‘z asarini turli kompyuter tizimlarida aks etishi mumkinligini yodda tutishi kerak. Agar u tashrif buyuruvchilarini (foydalanuvchilarini) yo‘qotishni istamasa, u holda har xil o‘lchamli va har xil texnologiyalarni qo‘llovchi (elektron-nur trubka bazasida bo‘lishi shart emas) monitorlarda hujjatning aniq va qisqa ifodalanishi haqida o‘ylab ko‘rish kerak.

Onlaynli hujjatlarni ko‘rib chiqishda brauzer oynasi o‘lchamlarini o‘zgartirishga majbur qiluvchi boshqa sabablar ham bor. Har bir savodli inson o‘qish tezligi boshqa sabablardan tashqari qator uzunligiga ham bog‘liqligini tushunadi. Agar qatorlar biror me’yordan sezilarli farq qilsa o‘qilganning qabul qilinishi

murakkablashadi. Psixologik tadqiqotlar ko'rsatishicha, matn qatorining optimal uzunligi alfavit simvollarini soni bilan uzviy bog'liq. Qulay o'qish zonasini aniqlash uchun alfavit o'lchamini 1,5 dan 2 gacha qiymatni qabul qiladigan koeffitsientga ko'paytirish kerak. Bu narsa rus tilida matn qatorlari 70 simvoldan uzun bo'lmasligini bildiradi. Ko'p foydalanuvchilar bunday formulaning mavjudligi haqida o'ylab ko'rmaydilar va xaddan tashqari uzun qatorlar paydo bo'lishining oldini olish uchun brauzer oynasining o'lchamlarini cheklab qo'yadilar.

Dreamweaver paketini yaratuvchilari dasturni turli o'lchamdagi monitorlarda hujjat qanday ko'rinishga ega bo'lishini tasavvur qilish imkonini beruvchi maxsus vosita bilan ta'minlab qo'yganlar.

Holat qatorining o'rta qismida (8.2-rasm) hujjat oynasining joriy o'lchamlari piksellarda chiqariladigan maydon joylashgan. Birinchi son-kenglik qiymatini, ikkinchi son balandlikni o'lchamini ifodalaydi. Agar maydonning ixtiyoriy nuqtasida sichqoncha tugmasini bosilsa, u holda Windows va Macintosh platformalarida hisoblash tizimlarining eng ommaviy konfiguratsiyalari uchun ettita standart o'lchamga ega bo'lgan muloqot oynasi chiqadi. O'lchamni belgilash uchun oynaning tegishli qatorini tanlash yetarli.

Menyuning birinchi bandi hujjat sohasining 592 pikselga teng bo'lgan kengligini belgilaydi va sohaning joriy balandligini o'zgarishsiz qoldiradi. Edite Sizes (O'lchamlarni to'g'rilash)ning oxirgi bandi hujjat oynasining o'lchamlarini tanlash va ularni standartlar ro'yxatiga kiritish imkonini beradi. Ekranida Preferences (Sozlashlar) muloqot oynasi paydo bo'ladi, unda Width (kenglik, eni) maydonida kenglik, Hight (balandlik) maydonda esa soha balandligi belgilanadi. Yangi standartning matnli tavsifi Description (tavsiya) maydonida ko'rsatiladi.

Barcha o'lchamlar dastur oynasining xizmat elementlarini: o'lchov chiziqlari va menyu qatorini inobatga olmasdan hisoblab chiqiladi. Faqatgina hujjat va uning ob'yektlari uchun ajratilgan ishchi soha hisobga olinadi.



Savol va topshiriqlar:

1. Qator uzunligining sahifani o'qishdagi ahamiyatini tushuntirib bering?
2. Dasturda sahifalar o'lchamlari qanday o'zgartiriladi?
3. Sahifa o'lchamlarini dastur oynasi qismlaridan qaysi biri tasvirlab turadi?
4. Interfeys nima?
5. Dastur oynasi qanday qismlarga bo'linadi?
6. Hujjat oynasi nima?
7. Menyu qatori menyulari va ulardagi buyruqlar vazifalarini sanab bering?
8. Uskunalar panelida qanday uskunalar bor va ularning vazifalari nimalardan iborat?
9. Dasturda qanday ish rejimlari mavjud?
10. WYSIWYG rejimida ishlashning qanday afzalliklari bor?

Palitra va tez yuklanish paneli

Tez ishga tushirish palitrasi (Launcher) – bu asosiy muloqot oynalari va dasturlar panellarini ekranga chiqarish uchun mo'ljallangan yordamchi, xizmatchi

palitra. Palitrani aks ettirish uchun Windows > Launcher buyrug'ini bajarish kerak. Bu buyruqning takroran bajarilishi palitrani ekrandan olib tashlaydi.

Palitraning Launcher (tez ishga tushirish) tugmasini bosish natijasida ekranga shu nomdagi muloqot oynasi chiqadi. Agar ochilgan palitra ekranning oldingi qavatida joylashgan bo'lsa, buyruq tugmasini takroran bosish uni yopilishiga olib keladi. Agar faol palitra hech bo'lmaganda qisman boshqa ob'yektlar bilan to'silgan bo'lsa, u oldingi qavatga o'tadi.

Palitra ettita eng ko'p talab qilinadigan dastur komponentlarining tez ishga tushirish tugmalarini birlashtiradi:

Site (Сайт) - katta saytlarni boshqarish uchun muloqot oynasi, murakkab kesishuvli bog'lanishlarga ega gipermatnli hujjatlar tizimi uchun.

- **Assets** (Библиотека) - bu nom harfma-harf tarjimada faollar yoki fondlar ma'nolarini beradi. Bu panel dasturda kutubxona rolini o'ynaydi. U gipermatnli hujjatlar dizayniga tegishli elementlarni o'zida saqlaydi. Bu elementlarga: tasvirlar, ranglar, tashqi bog'lanishlar, Flash – roliklari, shablonlar va boshqalar kiradi.

- **HTML Styles** (Стили HTML) – bu palitra matnli qator va abzatslar uchun HTML belgili stillar bilan ishlashga mo'ljallangan.

- **CSS Styles** (Каскадные листы стилей) - bu palitra stillarni kaskadli sahifalar bilan boshqarish uchun xizmat qiladi. Keyinchalik bu jumlaning «ierarxik stillar spetsifikatsiyasi» ko'rinishida ham qo'llaymiz.

- **Behaviors** (Режимы) - palitrasi gipermatnli sahifalarni interfaoligini boshqarish vositalarini o'zida birlashtirgan. Senariy sahifani ko'rish jarayonida ro'y beradigan hodisalar asosida quriladi. Bu jarayonning chizmasida ko'plab interfaol effektlar yig'iladi: oddiy animatsiyalashgan tugmalardan tortib ma'lumotlar omborini qayta ishlashning murakkab protseduralarigacha.

- **History** (Предыстория) – palitrasi foydalanuvchining HTML hujjatni tahrirlashdagi xatti-harakatlarini yozib boradi va saqlaydi. Tahrirlash jarayonidagi operatsiyalarni bekor qilish imkonini beradi va ayrim zanjirli ketma-ketliklarni buyruq ko'rinishida saqlaydi.

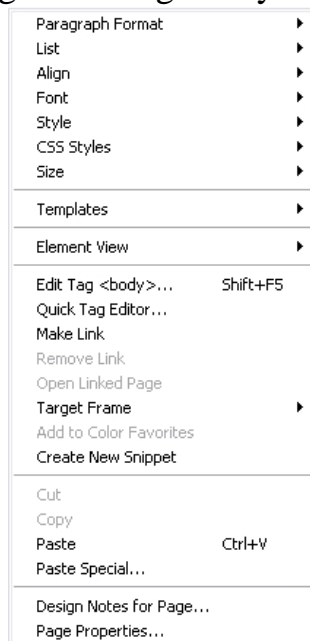
- **Code Inspector** (Инспектор кода). Dasturning eng muhim palitralaridan biri. HTML kodining ichki tahrirlagichi bo'lib Dreamweaverning grafikli rejimiga o'zaro bog'langan.

Tez ishga tushirish palitrasining o'ng tomondagi pastki burchagida to'g'ri burakli tugma joylashgan. Bu vosita palitraning yo'nalishini o'zgartirishga mo'ljallangan. Tez ishga tushirish palitrasini o'zgartirish mumkin. Buning uchun Edit > Preferences buyrug'ini berish kerak. Dasturning asosiy sozlashlariga ega bo'lgan muloqot oynasi ochiladi. Unda Panels (panellar) kategoriyasini tanlab oling va palitra ichidagi Show in Launche (tez ishga tushirish palitrasida ko'rsating) bo'limida beriladigan vositalar yordamida o'zgartiring. Holat qatorining o'ng qismida (8.2-rasm) joylashgan.

Kontekstli menyu

Kontekstli menyu Windows operatsion tizimining xatto boshlovchi foydalanuvchilari uchun ham yaxshi tanish. Boshqaruvning bu standart usuli ushbu operatsion tizimi boshqaruvida ishlovchi ko'pchilik ilovalar (dasturlar) tomonidan ko'p qo'llaniladi. Uning mohiyati sodda. Sichqonchanning o'ng tugmasini hujjat

oynasining ob'yekti ustida bosilsa, ekranga ob'yekt turiga bog'liq bo'lgan mazmundagi menyu chiqadi. Bu menyuda shu ob'yektga tegishli yoki ushbu holatda talab qilinishi ko'proq ehtimolligi bor bo'lgan buyruqlar to'plangan.



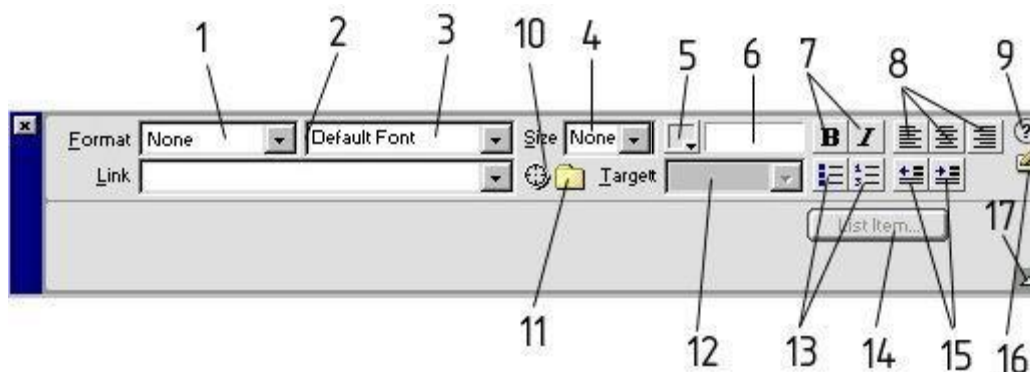
8.6-rasm. Kontekstli menyu.

Dreamweaver dasturining barcha versiyalarida kontekstli menyu mavjud. Kontekstli menyuning o'lchami ya'ni buyruqlari safi kengaydi, qo'shimcha darajalarga ega bo'ldi va endi har qanday holatda chaqirilishi mumkin. 8.6-rasmda oyna ixtiyoriy bo'sh nuqtasida sichqonchanning o'ng tugmasi bosilsa dastur chiqarib beradigan kontekstli menyuga misol keltirilgan.

Property Inspector palitrasi

Property Inspector (Инспектор свойств) palitrasi - bu dasturning asosiy uskunasidir. HTML sahifasida joylashgan ob'yektlar atributlarini o'zgartirish ishlarining asosiy qismini uning yordamida bajariladi. (8.7-rasm).

Property Inspector palitrasi ko'chib yuruvchi palitradan iborat bo'lib, uning tarkibi bajarilayotgan operatsiyaga, tanlangan ob'yekt turiga va kursorning holatiga qarab, mos ravishda o'zgarib turadi. Palitrani ekranga chiqarish uchun quyidagi Window > Properties (Окно > Свойства) buyruqni berish yoki CTRL+F3 tugmalar kombinatsiyasini bosish kerak. Bu buyruqni qaytadan berish palitrani ekrandan olib tashlaydi.



8.7-rasm. Property Inspector palitrasi.

Palitraning asosiy maydonlari va tugmalarini sanab o‘tamiz. Palitraning shu boshqaruv elementlari matnni formatlash va gipermatnli bog‘lanishlarni hosil qilishga tegishli vazifalarni o‘z ichiga oladi:

- Format (Формат) ro‘yxati matnning standart strukturali tiplaridan birini tanlash imkonini beradi (1).
- Link (Ссылка) ro‘yxati ushbu hujjatga tegishli giperbog‘lanishlar ro‘yxatini o‘zida saqlaydi. Har bir bog‘lanishni bir necha marta qo‘llash mumkin. Ya’ni bir marta tuzilgan bog‘lanishni, boshqa faol xududlarga ham takror qo‘llash mumkin (2).
- Matn fragmentini shriftini belgilash imkonini beradi. Bu ro‘yxatning punktlari shriftlar to‘plamidan iborat bo‘lib ular qo‘llanilishiga qarab joylashgan (3).
- Size (Размер) ro‘yxati shrift o‘lchamini tanlash uchun mo‘ljallangan (4).
- Standart ranglar palitrasini chaqirish tugmasi. Bu palitra tanlangan matnli fragment rangini tanlash imkonini beradi (5).
- Tanlangan rangning kodi tasvirlangan yoki kiritiladigan maydon (6).
- Shriftni tanlash tugmasi. Ko‘rinishi va funksiyasi standart boshqa dasturlardagidek (7).
- Matnni tekislash tipini tanlash tugmalari (8).
- Interfaol yordamni chaqirish tugmasi (9).
- Marker-ko‘rsatkich. Uni tegishli fayl yoki simvolga surish bilan gipermatnli bog‘lanish hosil qilinadi (10).
- Select File (Выбрать файл) muloqot oynasini ochish tugmasi. Fayllar tizimida harakatlanish va gipermatnli bog‘lanishlarni asosi bo‘lib xizmat qiladigan hujjatni tanlash (11).
- Target (Назначение) ro‘yxati gipermatnli bog‘lanishda ko‘rsatilgan hujjatni ekranda yuklanish usulini tanlash imkonini beradi. SHu bo‘limda havolali hujjat chiqish oynasi yoki freymni belgilanadi (12).
- Ko‘pchilik tahrirlagich dasturlarda standart ko‘rinishda, matnli va HTML, ro‘yxatlarni belgilash va nomerlash tugmasi (13).
- List Item (Раздел списка) tugmasi List Properties (Свойства списка) muloqot oynasini ochish uchun ishlatiladi. Bu oyna barcha tipdagi ro‘yxatlarni taxrirlashga mo‘ljallangan (14).
- Ikkita tugma, matnli fragmentlarni gorizontal surilishini boshqaradi (15).
- Kodlarni ekspress-tahrirlash oynasini chaqirish tugmasi (16).

- Expander arrow (Расширение) palitra o'Ichamlarini o'zgartirish uchun xizmat qiladi. Uning yordamida Property Inspector (Инспектор свойств) palitrasini ochish va yopish mumkin (17).

Property Inspector paneli yordamida bajariladigan buyruqlarning ko'pchiligi qayta ishlanayotgan ob'yektlar: rasm, matn yoki jadvallarga tegishlilari paydo bo'ladi. Xossalar panelida harakatlanish uchun TAB tugmasini bosish yoki kerakli qismga sichqonchani chap tugmasini bosish kerak. Tanlangan bo'limda kerakli ko'rsatkichni kiritib, ENTER tugmasini bosish uni o'rnatilishi uchun yetarli.

Property Inspector panelining ko'plab boshqaruv elementlari o'zida matnli yoki raqamli maydonlar kombinatsiyasi va bekiladigan ro'yxatlaridan iborat. So'ralgan qiymatni klaviaturadan terish yoki menyudan tanlash yo'li bilan tegishli parametrlarni o'rnatish uchun eng qulay usul hisoblanadi.

Dreamweaver o'zining bir qancha qo'shimcha funktsiyalari bilan foydalanuvchilarga manzur bo'ldi: birinchidan, Dreamweaver barcha palitralari erkin holatda turadi. Ularni qaerga surib qo'ysangiz o'sha joyda joylasha oladi. Bu esa monitoring ish maydoni xududini boshqaruvning xizmatchi elementlari va hujjatning oynasiga eng optimal bo'lish imkonini beradi. Ikki monitorli tizimda ishlaydigan foydalanuvchilar barcha yordamchi oyna va palitralarni bitta monitorga joylab, ikkinchisida esa hujjatlar bilan ishlashi mumkin. Ikkinchidan, dastur palitralari birlashish xususiyatiga ega. Uchinchidan, barcha ochiq palitralarni dastur ish stolidan olib tashlovchi buyruq mavjud. Bu operatsiyalarni F4 tugmasini bosish bilan ham bajarish mumkin. Bu tugmani takror bosish esa yana tiklash imkonini beradi.

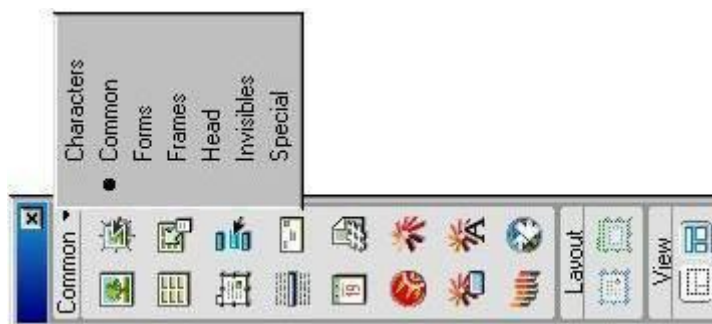


Savol va topshiriqlar:

1. Tez yuklanish palitrasi nima?
2. Tez yuklanish palitrasi qanday tugmalarni o'z ichiga oladi?
3. Kontekstli menyu nima va uning vazifalari nimadan iborat?
4. Property Inspector palitrasi qanday vazifalarni bajaradi?

Ob'yektlar palitrasi

Faqatgina eng oddiy gipermatnli hujjatlar o'zida havola bo'yicha boshqa hujjatlarga o'tuvchi buyruqlar biriktirilgan gipermatnli fragmentlarni birlashtiradi. Ammo ko'pchilik zamonaviy HTML sahifalari multimedialashib bormoqda. Ya'ni ularning tarkibiga quyidagi tashqi ob'yektlar kiradi: multimediali fayllar, ijro etish vositalari, kartinkalar, animatsiya fayllari va boshqalar. Objects (Палитра объектов) palitrasi ana shunday tashqi ob'yektlarni yaratish va qo'yish uchun xizmat qiladi. Bu palitrani ekranga chiqarish uchun Window > Objects (Окно > Объекты) buyrug'ini bajarish kerak. Xuddi shu buyruqni klaviaturadan CTRL+F2 tugmalar kombinatsiyasini bosish bilan ham berish mumkin. Bu buyruqni takror berish palitrani ekrandan olib tashlaydi.



8.8-rasm. Ob'yektlar palitrasi.

Palitra ettita nomdagi qismlardan iborat. Bular: Characters, Common, Forms, Frames, Head, Invisibles, Special.

- **Characters** (Специальные символы). Klaviaturada uchramaydigan maxsus simvollarini qo'yish (savdo markasi, kopirayt va boshqalar).
- **Common** (Общие) – bo'limi eng ko'p qo'llaniladigan ob'yektlarni birlashtiradi. Bu jadvallar, rasmlar, gorizontal chizg'ich, joriy kalendaridagi sanalar va boshqalar.
- **Forms** (Формы). Formalar va elementlar, ularning tashkil qiluvchilari: yoqib-o'chiruvchilar, ro'uxatlar, matn maydonlari va boshqalar.
- **Frames** (Фреймы). Freymlar tizimini yaratish. Har xil ko'rinishdagi freymlar qismlaridan tarkib topgan freymlar tizimini hosil qilish. Ya'ni yaratilayotgan saytni sahifasini freymlarga bo'lib olish.
- **Head** (Заголовок) - gipermatnli hujjatning sarlavha qismiga tavsif berish.
- **Invisibles** (Скрытые объекты). HTML sahifani brauzer dasturlarida namoyish qilishda, ko'rinmaydigan yashirin ob'yektlarni joylash. Bu palitraning eng kichik qismi bo'lib, u faqat uchta tipdagi ob'yektlarni qo'shib beradi: simvollar, izohlar va senariylar.
- **Special** (Специальные объекты) - bu bo'lim o'zida maxsus ob'yektlarni qo'yish tugmalarini birlashtiradi. Bularga appletlar, ActiveX ob'yektlari va boshqa tashqi to'ldiruvchilarni (plug-ins) kiritish mumkin.

Palitraning pastki qismida ikki guruhga bo'lingan to'rtta tugma joylashgan: Layout (Компоновка) va View (Вид). Ular maketlash rejimiga o'tish va sahifaning strukturasi tezda yaratishga imkon beradi. Bu elementlar bilan qisqacha tanishib o'tamiz:

- **Draw Layout Cell** (Создание макетной ячейки) - maketli katakcha yaratish uskunasini faollashtiradi.
- **Draw Layout Table** (Создание макетной таблицы). Maketli jadval yaratish uskunasini faollashtiradi.
- **Standard View** (Стандартный режим). Hujjat bilan ishlashning standart rejimiga o'tish.
- **Layout View** (Режим макетирования) - Hujjatni maketlash rejimiga o'tish.

Ushbu bo'limni nomi palitra sarlavhasi qatori ostida tasvirlanadi. Bo'limni ochish uchun shu nomni sichqoncha chap tugmasi bilan bosing. Ochilgan menyudan foydalanuvchi kerakli qismni tanlashi mumkin. Palitraning ayrim bo'limlarida hamma tugmalar ham tasvirlanmaydi. Masalan, Common (Общие) bo'limi yigirmaga yaqin tugmani o'z ichiga oladi, ularning hammasi ham palitra oynasiga sig'maydi. Boshqa tugmalarni ko'rish uchun prokrutkadan foydalanish (prokrutkani vazifasini

palitraning chetida joylashgan strelkali tugmalar bajaradi) yoki muloqot oynasining o'lchamini kattalashtirish kerak.

Aksariyat hollarda ob'yektlarni qo'yish uchun qo'shimcha parametrlarni kiritishga to'g'ri keladi. Bunday holatlarda ekranda foydalanuvchi tanlashi yoki ma'lumot kiritishi mumkin bo'lgan muloqot oynasi paydo bo'ladi. Bu oynaga foydalanuvchi ob'yektning kerakli ob'yektlarini tanlashi yoki kiritishi kerak.



Savol va topshiriqlar:

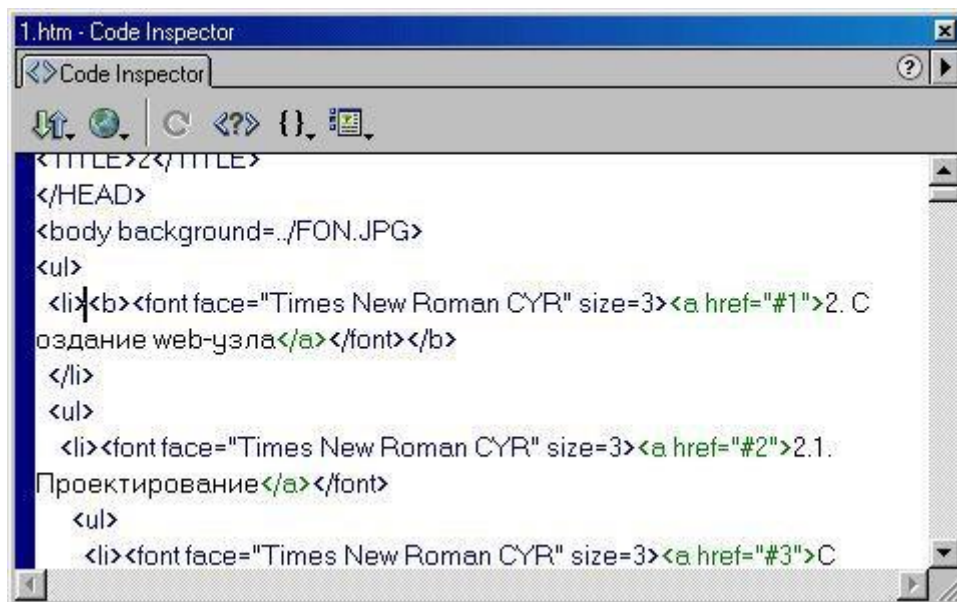
1. Ob'yektlar palitrasi nima vazifani bajaradi?
2. Ob'yektlar palitrasini ekranga chiqarish yo'llarini sanab bering?
3. Ob'yektlar palitrasi qanday qismlarga bo'linadi va ularning vazifalari nimalardan iborat?

Kodlar nazoratchisi bilan ishlash

Ko'p sonli HTML tahrirlagichlari sinfi orasida Dreamweaver dasturi gipermatnli hujjatlar kodlari bilan ishlash imkoniyatlari jihatidan keskin farq qiladi. Foydalanuvchining HTML kodlarni qo'lda tahrirlashdan qutilishi eng katta yutuq va tezkorlikni ta'minlab berdi.

Kodlar bilan to'g'ridan-to'g'ri ishlash, yashirin xatolarni identifikatsiya qilish, ortiqcha kodni bartaraf etish, gipermatnning tanlangan fragmentiga tezkor o'zgartirish kiritish, hujjatning strukturasi yoki uning formatlanishini optimallashtirish imkoniyatlari mavjud.

Dreamweaver paketi tarkibida gipermatnli hujjatlarni belgilash kodlarini ko'rish va tahrirlashga mo'ljallangan maxsus vositasi mavjud. Bu vosita Code Inspector (Инспектор кода) deb nomlanadi. Uni quyidagi usullar bilan o'rnatish mumkin: birinchidan, Window > Code Inspector (Окно > Инспектор кода) buyrug'ini berish yo'li bilan, ikkinchidan, bu F10 tugmasini bosish yo'li bilan. Bu buyruqlar takror bosilganda Code Inspector yana ish stolidan yo'qoladi.



8.9-rasm. Kodlar nazoratchisi.

8.9-rasmda Kodlarni nazorat qilish oynasi ko'rsatilgan. Palitrada kodlarning belgilanishini tanishtirishni View Options (Параметры отображения) tugmasi yordamida ochiladigan buyruqlar boshqaradi. Bu buyruqlarni ko'rib chiqamiz:

- Word Wrap (Перенос по словам). Kod qatorini ko'chirish rejimi. Agar u yoqilgan bo'lsa, HTML tili bayonnomasi tahrirlagich oynasining o'lchamiga etganda avtomatik ravishda yangi qatorga o'tadi.
- Line Numbers (Номера строк) - nomerlashni o'rnatish buyrug'i. Agar bu rejim o'rnatilgan bo'lsa, har bir qator tartib bilan nomerlanadi.
- Highlight Invalid HTML (Пометка некорректных кодов). Bu buyruq sintaktik xatoga ega konstruktsiyalarni Property Inspector panelida va hujjatning oynasida sariq rangda ajratish rejimini faollashtiradi. Shundan so'ng Property Inspector xato haqida xabar beradi va ayrim xollarda uni tuzatish uchun ko'rsatmalar beradi.
- Syntax Coloring (Раскраска синтаксических конструкций). Bu rejimda belgili kodlar turli xil ranglarda ajratiladi. Bu esa hujjatning kiritiladigan matni va belgili kodlarini tez ajratib olishga imkon beradi. Bundan tashqari Dreamweaver dasturi HTML tilining har bir deskriptori uchun alohida rang belgilash imkoniyatiga ega. Ranglarni belgilashning dastur asosiy ko'rsatkichlarini Code Colors (Цвета кода) bo'limida amalga oshiriladi.
- Auto Indent (Автоматический отступ) buyrug'i kodlar nazoratchisi oynasiga kiritilgan belgilash diskreptorlari uchun avtomatik joy qoldirish rejimini faollashtiradi.

Kodlar nazoratchisi oynasining o'ng qismida uchburchak formada tugma joylashgan. U hujjatning mavjud matnlariga xizmat qiladigan buyruqlardan iborat menyuni chaqiradi. Bu buyruqlarni sanab o'tamiz:

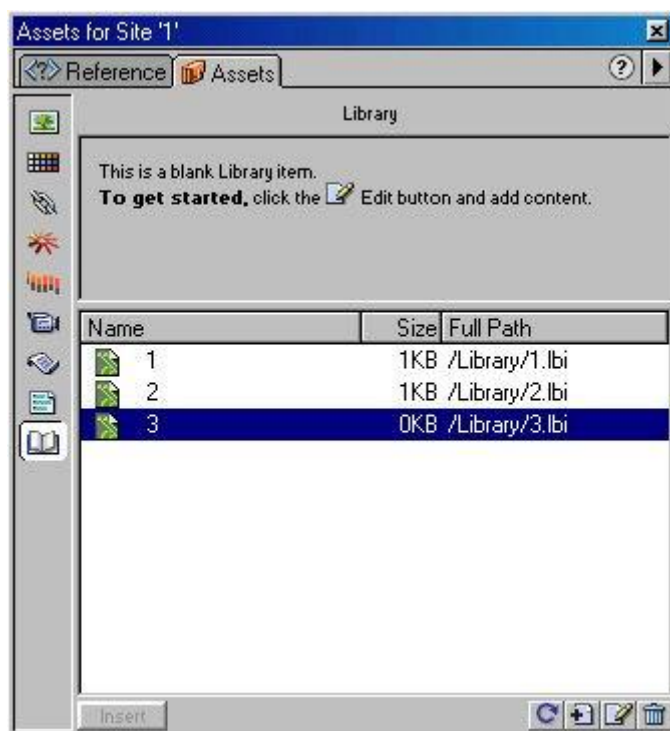
- Find and Replace (Найти и заменить). Qidirish va almashtirish vazifasini bajaradi.
- Find Next (Найти далее). Avval qidirilgan namunani mavjud parametr asosida qidirish. Bunda qidirish va almashtirish muloqot oynasi chiqmasdan to'g'ridan-to'g'ri amalga oshiriladi.

-
- Reference (Справочное руководство). HTML, JavaScript va CSS manbalari asosidagi ma'lumotli qo'llanmani chaqiradi.
 - Set Breakpoint (Вставить контрольную точку). Matn kursori pozitsiyasida nazorat nuqtasini belgilaydi.
 - Remove All Breakpoint (Удалить все контрольные точки). Hujjatning mavjud matnidan barcha nazorat nuqtalarini olib tashlaydi.
 - Cut (Вырезать). Tanlangan kodli fragmentni buferga qirqib olish.
 - Soru (Копировать). Tanlangan kodli fragmentni nusxasini buferga olish.
 - Paste (Вставить). Buferga olingan fragmentni qo'yish.

Kodlar nazoratchisini matn muharriri uchun biroz g'ayri oddiy holda muvaffaqiyatli qo'llash mumkin. Dreamweaver dasturi ASCII simvollarini qo'llash bilan yozilgan fayllarni, masalan oddiy matnli fayllarni yoki dasturning dastlabki matnini muvaffaqiyatli o'qiydi. Bunday hujjatlar Open (Открыть) buyrug'i yordamida ochiladi va kodli rejimda ishlaydigan oyna hujjatiga yuklanadi. Bunda dasturning ba'zi resurslariga kirish mumkin bo'lmay qoladi.

Dasturning kutubxonasi

Kutubxonalar - bu gipermatnli sahifaning dizayniga tegishli ko'p marta qo'llaniladigan elementlarni saqlash vositasidir. Bu elementlarga: tasvirlar, bog'lanishlar, matn bo'laklari va boshqalar kiradi. Dreamweaver dasturida esa kutubxonaning ishlash printsipti biroz boshqacha. Dastlab kutubxonaga saqlashning bitta elementi hisoblangan, ob'yekt yoki ob'yektlarga tegishli gipermatnli belgilash fragmentlari saqlanadi. Dreamweaver dasturida kutubxona elementlari maxsus Library papkasida saqlanadi. Bu papka saytning asosiy papkasi ichida joylashadi. Har bir Web-uzel o'zining standart elementlari va ularni saqlash papkasini tashkil qiluvchi shaxsiy kutubxonasiga ega. Ular bilan ishlash uchun palitraning Assets (Ресурсы) bo'limidan foydalaniladi (8.10-rasm). Kutubxonalar bilan boshqarish ishlarining asosiy qismini palitraning Assets (Ресурсы) bo'limi buyruqlari yordamida amalga oshiriladi.



8.10-rasm. Palitraning Assets bo‘limi.

Dreamweaverda kutubxonalarning asosiy imkoniyatlarini sanab o‘tamiz:

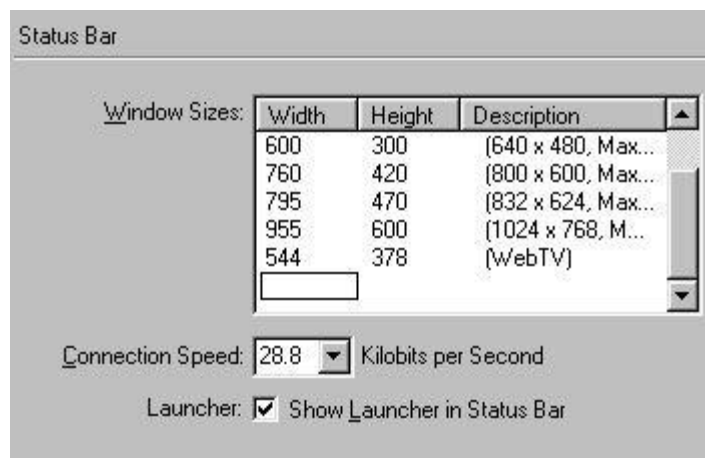
- Kutubxona elementi deb <body> deskriptori orasida yotgan har qanday fragmentni aytish mumkin;
- Kutubxona elementlari qo‘llanilishida cheksiz imkoniyatlarga ega. Ularni gipermatnli sahifalarga hech qanday tahrirlashsiz qo‘yish mumkin;
- Kutubxona elementlariga o‘zgartirish kiritilsa, ularni boshqa gipermatnli sahifalar va hujjatlarda qo‘llashda saqlanib qoladi.
- Sahifalardagi kutubxona namunasi va uning nusxalari orasidagi aloqani o‘zishga ruxsat etiladi.
- Dastur alohida saytlar o‘rtasida kutubxona elementlarini almashish buyruqlariga ega.
- Kutubxonalar dinamik xususiyatga ega bo‘lgan ob‘yektlarni saqlash xususiyatiga ega.
- Vaqt diagrammalari (timeline) va stillarning kaskadli sahifalari kutubxonada saqlanishi mumkin emas.

Holat qatori va uni o‘rnatish

Status Bar (Строка состояния) bu kichik bo‘lim dasturning holat qatori ko‘rinishi ko‘rsatkichlarini belgilaydi (8.11-rasm). Bu element hujjat oynasining pastki qismida joylashgan. Odatda bu qatorda joriy oyna o‘lchami haqidagi ma’lumotlar va uning yuklanishi uchun ketadigan vaqt ko‘rsatiladi. Undan tashqari holat qatorining o‘ng qismini tez yuklanish paneli egallaydi:

- Window Sizes (Размеры окна). Gipermatnli sahifalar har xil o‘lchamdagi oynalarda tasvirlanishi mumkin. Window Sizes (Размеры окна) paneli standart o‘lchamlar ro‘yxatini to‘ldirish uchun mo‘ljallangan. Barcha tahrirlash amallari klaviaturadan shu panelning tegishli maydonlariga kiritish yordamida bajariladi.

- Connection Speed Kilobits Per Second (Скорость соединения, Кбит/с). Bu maydonga server bilan bogʻlanishning taxminiy tezligi kiritiladi. Bu joriy sahifani taxminiy yuklanish vaqtini belgilaydi. Odatda 28,8 Kbit/s tezlik oʻrnatilgan boʻladi.



8.11-rasm. Status Bar boʻlimi.

- Show Launcher in Status Bar (Панель быстрого запуска в строке состояния) bu bayroqcha tez yuklanish panelini holat qatorida chiqishini boshqaradi.



Savol va topshiriqlar:

1. Dreamweaver dasturida kodlar bilan ishlashning qanday imkoniyatlari bor?
2. Kodlar nazoratchisini oʻrnatish usullarini sanab bering?
3. Kodlar nazoratchisi nima va uning vazifalari nimalardan iborat?
4. Dasturning kutubxonalari nima?
5. Dasturning kutubxonalari qanday vazifalarni bajaradi?
6. Holat qatori nima?
7. Holat qatori qanday oʻrnatiladi?
8. Holat qatori nimalardan tashkil topgan?



3.2. Dreamweaver dasturida web-sahifa yaratish

Dreamweaver dasturining foydalanuvchilari gipermatnli hujjatlarning koʻplab tahrirlash operatsiyalarini toʻgʻridan-toʻgʻri HTML deskriptorlari yordamida bajarishi mumkin. HTML tili haqida biz yuqorida batafsil maʼlumot oldik. Bu tilning elementlarini deskriptorlar yoki teglar deb nomlash qabul qilingan. Ingliz tilidan tag (yorliq, belgi va boshqa maʼnolarni beradi). Deskriptorlarni saqlash uchun ASCII belgilari qoʻllaniladi. Shuning uchun har qanday gipermatnli hujjatni belgilashni har qanday oddiy matn muharrirlarida bajarish mumkin. Buning uchun xatto Bloknot dasturi ham yetarli. HTML tilining kodlari hujjatning matnli massivlari qismi hisoblanadi.

Endi deskriptorlar bilan tanishib chiqamiz. Deskriptorlar ikki xil bo‘ladi. Yakka va juft deskriptorlar. Juft deskriptorlar ayrim xollarda konteynerlar deyiladi. Hujjatga tegishli matnlar va uning belgili kodlarini ajratish uchun, HTML tili kodlari burchakli qavslarda yoziladi. Masalan,
 yoki <p>. Juft deskriptorlar (konteynerlar) o‘z ichiga deskriptorlar juftligini, ochiladigan (boshlang‘ich) va yopiladigan (oxirgi) deskriptorlarni oladi. Bu juftliklar <tag> </tag> ko‘rinishida yozilib, o‘z qiymatini juftlik orasidagi hujjat fragmentiga o‘tkazadi. Boshqacha aytganda konteyner juftliklari orasiga hujjat fragmenti yoziladi. Deskriptorlar ana shu fragment uchun ishlaydi xolos. Yakka deskriptorlarda yopiladigan teg bo‘lmaydi. Masalan, <hr> deskriptori gorizontaal chiziq hosil qiladi,
 - esa matnli qatorni ifodalaydi, eng muhim yakka deskriptorlardan biri maxsus kodi hisobalanadi. U rasmlarni qo‘yiladigan holatini va joylashuvini belgilaydi.

Endi Dreamweaver dasturida shablon asosida yangi hujjat hosil qilamiz. Yangi hujjatning bo‘sh (to‘ldirilmagan) bo‘lishi hammamizga ma‘lum. Quyida bunday sahifaning Dreamweaver dasturidagi gipermatnli belgilari keltirilgan:

```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#FFFFFF">
</body> </html>
```

Bundagi deskriptorlarning vazifalari bilan HTMLda tanishib chiqdik. Undan tashqari <meta> deskriptorining charset atributi internetda tanlangan simvollarini namoyishini ta‘minlovchi kodirovkani ifodalaydi. Odatda yangi shablondagi (ISO-8859-1) kodli to‘plam kiril simvollarini ifodalay olmaydi.

Deskriptorlarni kiritish qoidalari

Belgilash kodlarini to‘g‘ridan-to‘g‘ri kiritish hujjatning faqat maxsus rejimida amalga oshiriladi. Bu rejimga o‘tishning bir nechta imkoniyatlari bor. Ulardan ayrimlarini ko‘rib chiqamiz:

- Window > Code Inspector buyrug‘ini bering. Xuddi shu buyruqni F10 tugmachasini bosish yordamida ham berish mumkin;
- Uskunalar panelining chap qismida joylashgan Code View (Режим разметки) tugmasini bosib kodlarni tasvirlash rejimiga o‘tish mumkin;
- Uskunalar panelidagi chapdan ikkinchi Code and Design View (Режим разметки и планировки) tugmani bosish bilan hujjat oynasini ikki qismga bo‘lish mumkin.

Kodlar nazoratchisida joriy hujjatning kodlari tasvirlanadi. Belgilashda qo‘llaniladigan ko‘pchilik deskriptorlar alohida rangda tasvirlanadi. Deskriptorlarning ranglari dasturning asosiy ko‘rsatkichlariga kiradi. Ranglarni foydalanuvchining xohishiga qarab o‘zgartirish mumkin. Agar foydalanuvchi hujjat ustida biror tahrirlash ishlarini olib bormoqchi bo‘lsa u holda F5 tugmasini bosish yoki View > Design View (Вид > Обновить) buyrug‘ini berish kerak. Bu buyruq hujjatning kodlarni belgilash rejimidan tahrirlash rejimiga o‘tkazadi. Tahrirlash

rejimidagi ob'yektlarni qo'yish, formatlash buyruqlari va har qanday o'zgarishlar darhol kodlar inspektoriga ko'chadi. Agar tahrirlash rejimida biror ob'yekt yoki fragmentni belgilasangiz, kodlarni belgilash rejimida ham shu ob'yekt yoki fragmentga tegishli kodlar belgilanadi. Bu esa dasturda ikki yoqlama ishlash imkoniyatini ta'minlab beradi.

Kodlardan nusxa olish

Dreamweaverda boshqa HTML tahrirlagichlari bilan gipermatnli belgilashdagi kodlarni almashish imkoniyati mavjud. Buning uchun Windows operatsion tizimining ko'pchilik dasturlarida mavjud bo'lgan, belgilangan fragmentdan nusxa olish va qo'yishning standart buyruqlaridan foydalaniladi. Aytaylik HomeSite dasturida ochilgan gipermatnli hujjatning belgilash kodlaridan nusxa olishimiz kerak:

- Nusxa olish uchun Edit > Copy (Правка > Копировать) yoki CTRL+C tugmalar kombinatsiyasidan foydalanamiz;
- Buferga olingan nusxani qo'yish uchun Edit > Paste (Правка > Вставить) yoki CTRL+V tugmalar kombinatsiyasidan foydalanamiz.

Matn muharrirlarida tanlangan fragmentni sichqon yordamida surish yo'li bilan ko'paytirish imkoniyati mavjud. Bu imkoniyat Dreamweaver dasturida ham ishlaydi. Buning uchun CTRL tugmasini bosgan holda belgilangan fragmentni sichqon ko'rsatkichi bilan kerakli joyga surib ko'paytirish mumkin.



Savol va topshiriqlar:

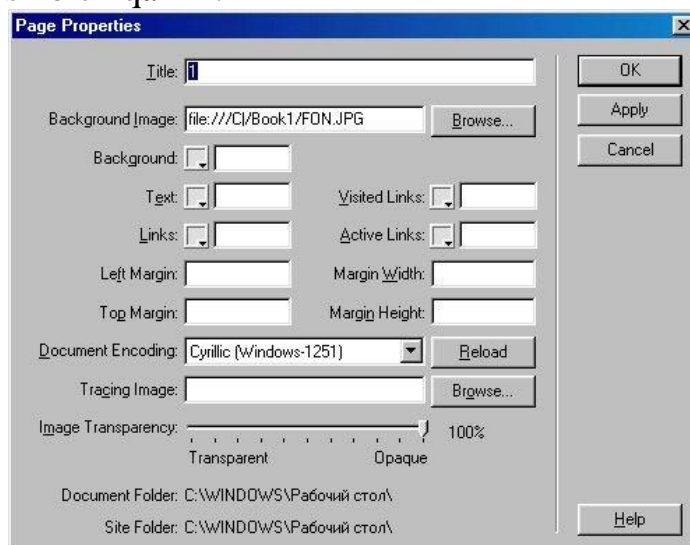
1. HTML so'zining lug'oviy ma'nosi qanday ?
2. Deskriptorlar nima?
3. Deskriptorlarning qanday ko'rinishlari mavjud?
4. Deskriptorlarni kiritishni tushuntirib bering?
5. Kodlardan nusxa olish ketma-ketligini aytib bering?

Hujjatning o'lchamlarini belgilash

Dasturda yaratilgan har qanday yangi hujjat bo'm-bo'sh oq rangdagi fonli sahifadan iborat bo'ladi. Hujjatda ish boshlashdan oldin uning bir qancha muhim parametrlarini kiritib olish kerak. Bular ish maydonining o'lchami, fon rangi, hujjat sarlavhasi, havola ranglaridir. Bu parametrlarni ish jarayonining ixtiyoriy vaqtida ham o'zgartirish mumkin. Sanalgan barcha parametrlarni yagona Page Properties (Свойства страницы) muloqot oynasi yordamida kiritiladi (8.19-rasm). Bu muloqot oynasini bir necha yo'llar bilan ochish mumkin:

- Dastur menu qatoridan Modify > Page Properties (Изменить > Свойства страницы) buyrug'ini berish bilan;
- CTRL+J tugmalar kombinatsiyasini bosish bilan;
- Kontekstli menyudan Page Properties (Изменить > Свойства страницы) buyrug'ini berish bilan.

Endi Page Properties (Свойства страницы) muloqot oynasining boshqaruv elementlari bilan tanishib chiqamiz:



8.19- rasm. Page Properties muloqot oynasi.

- **Title** (Заголовок). Sahifaning sarlavhasini kiritish uchun xizmat qiladi.
- **Background Image** (Фоновый рисунок). Hujjatga fon sifatida qo'yiladigan rasm ko'rsatiladi.
- **Background** (Фоновый цвет). Sahifaning fon rangini tanlash imkoniyatini beradi.
- **Text** (Текст). Shriftning rangini tanlash imkoniyatini beradi.
- **Links** (Ссылки). Bu bo'limda giperhavolalar rangi va grafikli havolalar ramkasi rangi belgilanadi.
- **Visited Links** (Просмотренные ссылки). Brauzerda ko'rilgan havolalar rangini tanlash.
- **Active Links** (Активные ссылки). Faol havola rangini belgilash.
- **Left Margin** (Левое поле). Sahifaning chap tomonidan qoldiriladigan maydonni pikselda belgilaydi.
- **Top Margin** (Верхнее поле). Sahifaning yuqori qismidan qoldiriladigan maydonni pikselda belgilaydi.
- **Margin Width** (Ширина полей). Sahifa maydoni enini belgilaydi.
- **Margin Height** (Высота полей). Sahifa maydoni bo'yini belgilaydi.
- **Document Encoding** (Кодировка документа). Kiril simvollaridan iborat sahifalar tuzish uchun juda muhim ko'rsatkich. Chunki uning yordamida qanday simvollarini brauzer dasturlarida tasvirlab berish ko'rsatiladi. Rus alifbosini faqat quyidagi to'rtta kodirovka tasvirlab bera oladi xolos ISO-8859-5, MacCyrillic, KOI8-R va Windows-1251.
- **Reload** (Повторная загрузка). Tugmasi yordamida hujjatning grafik tahrirlash ishlari bilan bog'liq oxirgi saqlangan ko'rinishni tiklash imkonini beradi.
- **Tracing Image** (Трафаретное изображение). Sahifaning dizayni bilan ishlashda namuna sifatida ishlatiladigan tasvirlarni ifodalaydi. Bu tasvirlar sahifani ko'rish rejimida ko'rinmaydi.

• **Image Transparency** (Прозрачность изображения). Iz qoldirish uchun tanlab olingan rasmning shaffoflik darajasi belgilanadi.

• **Document Folder** (Папка документа). Hujjatning qaysi katalogda joylashganini ko'rsatib turadi.

• **Site Folder** (Корневая папка узла). Tegishli Web-uzelning o'zak papkasiga to'liq yo'l ko'rsatadi.

Yuqoridagi ko'rsatkichlar tanlangandan so'ng, barcha o'zgarishlarni saqlab, ularni ko'rish uchun File > Preview in Browser (Файл > Просмотр в браузере) buyrug'i beriladi.

Gorizontal va vertikal chizg'ich

Hujjat bilan ishlashda o'lchash va etalon vazifalarini bajaradi. Chizg'ichga qarab ob'yektlarning o'lchamini boshqa ob'yektlarga nisbatan joylashuvini ko'rish mumkin. Boshqaruvda chizg'ichlar maxsus buyruq yordamida o'rnatiladi. Buning uchun View > Rulers (Вид > Линейки) buyrug'ini berish kerak. Bu menyuning punktlarini sanab o'tamiz.

• **Show** (Показать) buyrug'i chizg'ichni o'rnatib beradi. Bu buyruqni - CTRL+ALT+R tugmalar kombinatsiyasi yordamida ham bajarish mumkin. Buyruqni takror berish chizg'ichni ekrandan olib tashlaydi.

• **Pixels** (Пиксели), **Inches** (Дюймы), **Centimeters** (Сантиметры) - buyruqlari o'lcham birligini tanlash imkonini beradi.

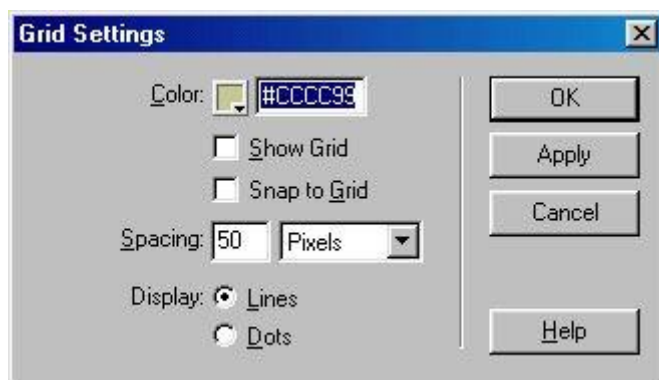
• **Reset Origin** (Восстановить начало координат). Koordinatalar boshini sahifaning ixtiyoriy nuqtasi bilan bog'lash mumkin. Buning uchun chiziqlar kesishuvida joylashgan krest shaklidagi markerni hujjatning ixtiyoriy joyiga olib o'tish yetarli. Ushbu buyruqning bajarilishi sanoq nuqtasini dastlabki vaziyatga qaytaradi.

Montaj to'ri

Montaj to'ri - hujjat elementlarini tekislash va joylashtirish uchun mo'ljal olish vositasidir. Hujjat elementlariga: matn, rasmlar, jadvallar va boshqalar kiradi. Sahifani rejalashtirishning eng asosiy vositasidir. Loyihaning dizayn bilan bog'liq ishlari tugagach to'rni olib tashlash mumkin. To'rni yana ekranga chiqarish uchun View > Grid > Show Grid (Вид > Разметочная сетка > Показать сетку) buyrug'i beriladi yoki CTRL+ALT+G tugmalar kombinatsiyasi bosiladi. Olib tashlash uchun shu buyruq takror beriladi.

View > Grid > Edit Grid (Вид > Разметочная сетка > Настроить сетку) buyrug'i 8.20- rasmda ko'rsatilgan muloqot oynasini ekranga chiqaradi. Uning tarkibiy qismi bilan tanishib chiqamiz:

- **Color** (Цвет) - to'r rangini tanlash imkonini beradi.
- **Show Grid** (Показать сетку) – to'rni o'rnatish yoki olib tashlash.
- **Spacing** (Расстояния) – o'lchash birligi va chiziqlar orasidagi bo'linish qadamini belgilaydi.
- **Display** (Отображение) - to'r chiziqlari ko'rinishini belgilaydi.



8.20-rasm. Montaj to‘rini o‘rnatish.



Savol va topshiriqlar:

1. Hujjat o‘lchamlarini belgilash uchun qaysi buyruqdan foydalaniladi? Bu buyruqning mos kombinatsion tugmalarini ayting?
2. Page Properties muloqot oynasi qismlari vazifalarini aytib bering?
3. Chizg‘ich va uning vazifalari nimalardan iborat?
4. Montaj to‘ri nima va uning vazifalari nimalardan iborat?

Dreamweavering ro‘yxatlar bilan ishlash imkoniyatlari

Ro‘yxatlarni nomerlash deb – bo‘lim yoki qismlar ketma-ketligini tartibga soluvchi va ularning har biri navbatdagi yangi nomer bilan tartiblanishini tushunish mumkin. HTML tilida ro‘yxatlarni hosil qilish bilan biz yuqorida batafsil tanishib chiqdik. Bundan tashqari tartiblangan ro‘yxatlar asosida murakkab ierarxik ro‘yxatni tuzish mumkin. Bunda murakkab ro‘yxatning qismlari bo‘lib quyi (oddiy) ro‘yxatlar olinadi.

Nomerlangan ro‘yxatlar quyidagi yo‘l bilan hosil qilinadi.

1. Hujjatning bir nechta tizimli bo‘lagini oling. Bunday bo‘laklarga abzatslar, har qanday standart sarlavhalar yoki boshqa ro‘yxatlarning bo‘limlarini kiritish mumkin:

1. Dreamweaver
2. Homesite
3. Hotmetal Pro

Text > List > Ordered List (Текст > Список > Нумерованный список) buyrug‘ini bering. Bu buyruqni CTRL+F3 tugmalar kombinatsiyasi yordamida ham berish mumkin.

Belgili ro‘yxatlarni yaratish

Belgili ro‘yxatlar deb – ma‘lum belgi ostida ketma-ket bo‘limlarni sanash tushuniladi. Bunday ro‘yxatlarni HTMLda hosil qilish bilan biz yuqorida tanishib chiqdik.

Brauzerlarning oxirgi versiyalarida ro‘yxatlarni belgilash uchun type atributi bor. Bu atributni quyidagi qiymatlarini qo‘llash mumkin: circle (aylana), disk (disk), square (kvadrat). Ya‘ni ro‘yxatning har bir bo‘limi shu belgilar bilan belgilanadi.

Bundan tashqari kichik xajmdagi rasmlarni ham ro'yxatning maxsus belgisi sifatida qo'llash mumkin.

Betartib ro'yxatlarni yaratish tartibi quyidagicha:

1. Hujjatning bir nechta bir-biriga qo'shni strukturali bloklarini tanlang. Text > List > Unordered List buyrug'ini bering. Bu buyruqni CTRL+F3 tugmalar kombinatsiyasi yordamida ham bajarish mumkin.

- Toshkent
- Samarqand
- Buxoro
- Xorazm



Savol va topshiriqlar:

1. Ro'yxatlarni belgilash deganda nimani tushunasiz?
2. HTML tilida ro'yxatlar qaysi deskriptor yordamida hosil qilinadi?
3. Belgili ro'yxatlar qanday hosil qilinadi?

Matnlarni formatlash. Shriftlar

Bizga boshqa dasturlardan belgilangan matnni yoki butun hujjatdagi matnni shriftini o'zgartirish ma'lum. Endi ana shu jarayonni HTML tilida qanday bajarilishini ko'rib chiqamiz. Bu vazifani HTMLda maxsus deskriptor yordamida bajariladi. Uning qiymatlari bo'lib bir nechta shriftlar nomi ishlatilishi mumkin, masalan: Shriftni belgilash .

Shriftlarni belgilashni texnik tomoni esa matn muharrirlaridan hech ham farq qilmaydi:

1. Matnni belgilang yoki matn kursorini hujjatning shrifti o'zgarishi kerak bo'lgan joyiga qo'ying;
2. Text > Font (Текст > Шрифт) buyrug'ini bering va kerakli shriftni tanlang. Xuddi shu ishni Property Inspector (Инспектор свойств) palitrasi yordamida ham bajarish mumkin.

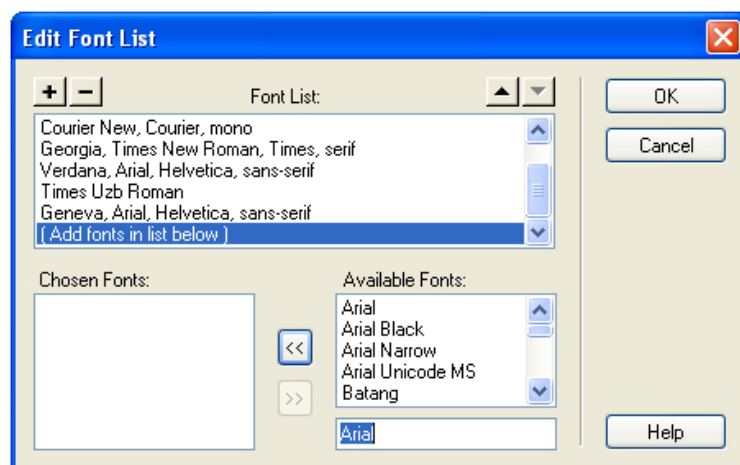
Tanlangan shriftning nomi shriftlar ro'yxatining nomi hisoblanadi.

Shriftlar ro'yxatini tahrirlash

Shriftlar to'plami ro'yxatiga foydalanuvchi o'z xohishiga qarab yangi shriftlar qo'shishi yoki qo'llanilmaydigan shriftlarni olib tashlashi ham mumkin:

1. Text > Font > Edit Font List (Текст > Шрифт > Редактировать список шрифтов) buyrug'ini bering. Bu buyruqni CTRL+F3 tugmalar kombinatsiyasini bosish bilan ham bajarish mumkin. Buyruqni berish natijasida Edit Font List muloqot oynasi hosil bo'ladi.

2. Bu oyna yordamida foydalanuvchi o'ziga kerakli shriftlar ro'yxatini qaytadan tuzib olishi mumkin.
3. Ishni tugallashdan oldin muloqot oynasida OK tugmasi bosiladi.



8.21-rasm. Shriftlar to'plamini tahrirlash muloqot oynasi.

Muloqot oynasining tugma va maydonlari bilan tanishib chiqamiz:

- Font List (Список шрифтов). Muloqot oynasining bu bo'limida ayni vaqtdagi ro'yxatga tegishli shriftlar berilgan. Endi qo'shiladigan shriftlar esa ro'yxatning eng oxiriga joylashadi.
- Chosen Font (Выбранные шрифты). Tanlangan shriftlar nomi.
- Available Fonts (Наличные шрифты). Ushbu hisoblash tizimida o'rnatilgan shriftlar ro'yxati. Bu shriftlardan birortasini tanlangan shriftlar to'plamiga qo'shish uchun, chapni ko'rsatish yo'nalish tugmasini bosish kerak.
 - To'plamdagi tanlangan shriftni o'chirish uchun «←» belgili tugmani, to'plamga yangi shrift qo'shish uchun «+» belgili tugmani bosish kerak.
 - Yuqoriga va pastga qaragan strelkalar esa tanlangan shriftni ro'yxatdagi holatini o'zgartirish, ya'ni yuqoriga va pastga yurish imkonini beradi.

Matn o'lchamini belgilash

HTMLda matn o'lchamini qanday belgilash bilan tanishib chiqamiz. Buning uchun `` deskriptorida `size` atributi qo'llaniladi. Masalan, `O'lchamni belgilashning absolyut tizimi `. Bundan tashqari `<body>` deskriptori yordamida ham `<basefont size="qiymat">` shriftni belgilash mumkin. Bunda sizening qiymatiga birdan to ettigacha bo'lgan har qanday qiymatni qo'yish mumkin.

Kattalashadigan o'lchamlarda esa `size` atributi qiymati oldiga + (kattalashtirish) yoki - (kichiklashtirish) belgisi qo'yiladi. Nisbiy tizimda shriftni o'lchamini o'zgartirishga misol keltiramiz:

```
<font size="+1">Nisbiy tizimda o'lcham</font>
```

Shrift o'lchamini quyidagi yo'l bilan o'zgartirish mumkin:

1. Matnni belgilang yoki kiritiladigan joyga kursorni qo'ying.
2. Absolyut o'lchamni belgilash uchun Text > Size (Текст > Размер) buyrug'ini bering yoki nisbiy tizimida o'lchamni berish uchun Text > Size Change (Текст >

Изменить размер) buyrug'ini bering. Ikkala buyruqni ham Property Inspector (Инспектор свойств) palitrasidan berish mumkin. Palitraniing Size (Размер) ro'yxatida shrift o'lchamini o'zgartirishning barcha imkoniyatlari ko'rsatilgan.

3. Absolyut yoki o'sish o'lchamini tanlang.

Matnlarga rang berish

Agar matnga biror rang berilmagan bo'lsa, u holda Windows operatsion tizimidagi yoki brauzerda tanlangan rang tasvirlanadi. Ko'pincha bu rang qora rang bo'ladi. Butun massivning simvollariga rang hujjatning asosiy ko'rsatkichlarida belgilanadi. Ko'pincha hujjat yaratib olingach, bu kabi ko'rsatkichlar kiritilib olinadi. Buning uchun yuqorida ko'rganimizdek Modify > Page Properties (Изменить > Свойства страницы) buyrug'ini berish kerak (CTRL+J). Har qanday simvollar to'plamiga alohida rang berish mumkin. Ranglarni saqlash uchun deskriptorida color atributi ishlatiladi. Masalan, yashil-ko'k rangini belgilovchi kod quyidagicha yoziladi:

```
<font color="#66FFFF">bu moviy rangning kodi</font>
```

Shuni esda tutish kerakki HTML tilida rang o'n oltilik sanoq tizimi kodlari yordamida belgilanadi. Palitraniing ayrim elementlari o'zining nomiga ega, masalan green (ko'k), red (qizil) va boshqalar.

Dasturda ranglarni tanlashni Property Inspector (Инспектор свойств) palitrasidan foydalanib ham uskunaviy yo'l bilan belgilash mumkin. Bu jarayon xuddi boshqa matn muharrirlarida bajarilgandek boradi.

Matnli maydonlar

Matn maydoni – sahifaning asosiy qismi hisoblanadi. Chunki har qanday Web-sahifaning asosiy belgilovchi qismi matnlar hisoblanadi.

Matnli maydonlar uch xil ko'rinishda bo'ladi (8.22-rasm):

- Single line (Однострочное). Qisqa xabar yoki javoblarni kiritish uchun xizmat qiladi.
- Multi line (Многострочное). Ixtiyoriy uzunlikdagi xabarni kiritish uchun mo'ljallangan.
- Password (Пароль). Parolni kiritish uchun mo'ljallangan.

Matnning uchala ko'rinishi ham bitta buyruq yordamida qo'yiladi. Ularni tiplarga ajratish esa Property Inspector (Инспектор свойств) palitrasi yordamida amalga oshiriladi. Buning uchun shu nomlardagi uchta qayta ulagich xizmat qiladi.



8.22-rasm. Matn maydonlarining tiplari.

Single line matn maydoni

Bitta qatordan tarkib topgan oddiy matn maydoni. U `<input>` deskriptori yordamida belgilanadi. Quyidagi misolda shunday ob'jektning ifodalovchi belgisi kod qatori keltirilgan:

```
<input type="text" name="maydon_nomi">
```

bu yerda `type` atributi maydonning tipini bildiradi, `name` atributi esa ob'jektning nomini bildiradi.

Oddiy matn maydoni bilan bog'liq ko'rsatkichlar bilan tanishib chiqamiz:

- **TextField (Текстовое поле).** Palitrani bu bo'limi matn maydonining nomini belgilaydi. Bu yerda maxsus belgilardan foydalanish, bir nomni takrorlash va maydonni bo'sh qoldirish ham mumkin emas.
- **Char Width (Ширина в символах).** Matn maydonining enini simvollar soni bo'yicha belgilash. Odatda bu o'lcham 20 ta simvolni tashqil qiladi.
- **Max Chars (Максимальный объем).** Matn maydoniga kiritiladigan simvollarining maksimal soni.
- **Init Val (Начальное значение).** Bu maydonga qarab ko'rish dasturi sahifani birinchi yuklashdan keyin olib chiqadigan matn qatorini kiritish mumkin.

Password matn maydoni

Kiritilgan simvollarining oynada aks etmasligini ta'minlash maqsadga muvofiq bo'lgan holatlar ham mavjud. Ko'pincha bunday zarurat maxfiy axborotlarni, masalan, parollar yoki kalit so'zlarni kiritishda yuzaga keladi.

Password (Пароль) matn maydoniga kiritilgan ma'lumot faqat yulduzchalar ko'rinishida tasvirlanadi. Matn maydoni bunday xususiyati bilan boshqa tipdagi matn maydonlaridan keskin farq qilib turadi.

Multi Line matn maydoni

Katta hajmdagi matnli ma'lumotlarni kiritish uchun Multi Line (Многострочное) tipdagi maydonlar ishlatiladi. Bunday tipdagi ob'ektlarni yaratish uchun, sahifaga matn maydoni qo'yish kerak, masalan Insert > Form Objects > Text Field (Вставка > Объекты форм > Текстовое поле) buyrug'i yordamida. Xossalar paneli ochish va bu palitradan Multi Line (Многострочное) qayta ulagichni

oling. Natijada, bir qatorli matn maydoni ko'p qatorli matn maydoniga aylanadi. Uning uzunligi 18 simvoldan, balandligi esa 3 qatordan iborat.

Bu tipdagi ob'ektlar ishlash uchun yana Property Inspector (Инспектор свойств) palitrasiga murojat qilamiz:

- TextField (Текстовое поле) - ko'p qatorli maydonning nomi.
- Char Width (Ширина в символах) - matn maydonining simvollarda belgilanadigan kengligi.
- Num Lines (Число строк) - matn maydonining qatorlar soni.
- Int Val (Начальное значение) - maydonning boshlang'ich qiymati.
- Wrap (Перенос по словом) – ko'p qatorli matn maydonida matnni yangi qatorga ko'chirish imkoniyatlarini belgilaydi.



Savol va topshiriqlar:

1. Matnlarni formatlash jarayonida shriftni belgilash uchun qaysi deskriptordan foydalaniladi?
2. Shriftlar ro'yxatini tahrirlash ketma-keligini aytib bering?
3. Matn o'lchamini belgilash uchun qaysi deskriptordan foydalaniladi?
4. Matnlarga rang berish ketma-keligini aytib bering?
5. Qanday matn maydonlarini bilasiz?
6. Single line matn maydoni nima va u qanday hosil qilinadi?
7. Password matn maydoni nima va u qanday hosil qilinadi?
8. Multi Line matn maydoni nima va u qanday hosil qilinadi?



3.3. Dreamweaver dasturida Web-uzel yaratish

Jahon o'rgimchak uyasining (World Wide Web) barcha informatsion manbalarini ikkita katta sinfga bo'lish qabul qilingan: sahifalar va uzellar yoki saytlar. Biz yuqorida web – sahifalar ta'rifi bilan tanishdik.

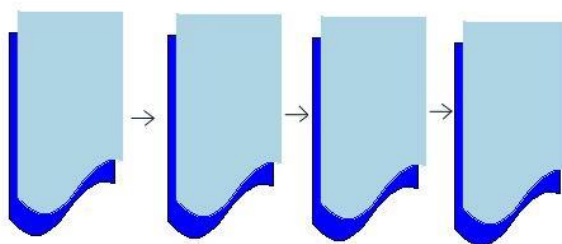
Web-uzellar ya'ni saytlar – deganda o'ziga sahifalarni birlashtirgan tizim tushuniladi. Boshqacha aytganda uzul deganda minglab hujjatlarni o'zida birlashtirgan, ko'psonli bog'lanishga ega, multimediali murakkab tizim tushuniladi.

Dreamweaver paketi - bu ko'p qirrali dastur. Uni alohida sahifalar yaratish uchun oddiy gipermatnli hujjatlar tahrirlagichi sifatida ham ishlatish mumkin. Bundan tashqari dastur har qanday murakkablikdagi Web-uzellarni ham yarata oladi.

Chiziqli struktura

Kompyuter paydo bo'lgunga qadar barcha chop etiladigan: badiiy kitoblar, qo'llanmalar, ma'lumotnomalar, texnik ko'rsatmalar chiziqli strukturaga misol bo'la olar edi. Bugungi kunga kelib bularni multimediali qo'llanmalar va elektron ko'rsatmalar ko'rinishida qo'llanmoqda. Chiziqli zanjir elektron magazinda buyurtma

berishni rasmiylashtirish qadamlari, ayrim tarmoq manbalariga kirish uchun ro'yxatdan o'tish yoki rasmiylashtirish ketma-ketliklarini tushunish mumkin.



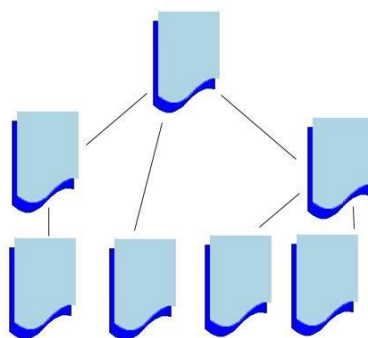
8.12-rasm. Chiziqli struktura.

Chiziqli strukturaga kirish nuqtasi bo'lib birinchi sahifa xizmat qiladi. Bu bosh sahifa odatda index.htm yoki main.htm deb nomlanadi. Bu esa kitob yoki jurnallardagi muqova rolini bajaradi. Bu sahifa bir vaqtning o'zida ham namoshiybop, ham tanishtiruv sahifasi, ham ma'lumotli sahifa bo'lmog'i lozim. Qidiruv mashinalari har bir sahifani alohida indekslaydi. Shuning uchun barcha ichki sahifalarni bosh sahifaga o'tkazadigan harakatlanish tugmalari bilan to'ldirish kerak.

Ierarxik struktura

Saytlar yaratishning zanjirli printsipli sahifalar joylashuviga va ular orasidagi bir-biriga o'tishlarga qat'iy cheklashlar qo'yishga asoslanadi. Ierarxik bo'ysunish printsipli bo'yicha qurilgan tugunlar ancha katta moslashuvchanlikka ega. Ko'plab korxonalar, tashkilotlar Ierarxik struktura chizmasi asosida qurilgan. Bu esa ierarxik strukturaning kamchiliklari va yutuqlari yaxshi o'rganilganidan darak beradi. Bu strukturani ba'zan daraxtsimon struktura ham deyilib, bitta bosh o'zak (butun saytga kirish nuqtasi hisoblangan) sahifaga ega bo'ladi.

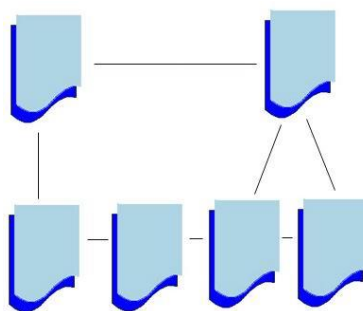
Har qanday ichki sahifa informatsion manba hisoblanib bosh sahifa bilan mantiqiy bog'langan bo'ladi.



8.13-rasm. Ierarxik struktura.

Ko'p bog'lanishli struktura

Ko'p bog'lanishli strukturada hech qanday cheklanishlar bo'lmaydi. Ya'ni har bir sahifa boshqa barcha sahifalar bilan bog'langan bo'lishi kerak. Katta saytlarda tashrif buyuruvchilarning mo'ljal olishi uchun maxsus indeksatsiya tizimlari yoki saytning aylanish kartasi ifodalanishi kerak.



8.14-rasm. Ko'p bog'lanishli struktura.

Materialning taqsimoti

Materialning Web-uzelda bo'limlar bo'yicha taqsimoti mantiqiy ravishda amalga oshirilishi kerak. Har bir sahifaga faqat belgilangan predmet yoki mavzudagi material joylanishi kerak. Tanlangan gipermatn ostida aynan shu mavzudagi hujjat borligi foydalanuvchining ishonchini yanada orttiradi. Ayrim hollarda buning aksini, ya'ni tanlangan gipermatnga xos bo'lmagan hujjatning ochilishi ham ko'rish mumkin.

Saytni qismlarga bo'lishda ham ma'lumotning katta-kichikligiga, ma'lum mavzu yoki predmetga xosligiga qarab bo'lish ham maqsadga muvofiqdir. Yuqorida aytilgandek biror sahifada bir predmet yoki mavzuga xos bo'lgan matnlar bilan birga, rasmlar, jadvallar, animatsiyalar va boshqa ko'rinishdagi materiallarning ham shu bo'limga xosligi katta ahamiyatga ega. Ma'lumotlarning mos kelmasligi foydalanuvchilarning ishonchini yo'qotib, ularning sonini keskin kamaytirib yuboradi. Bunday holatga tushib qolmaslik uchun Web-dizaynerdan qunt bilan ishlashni talab qiladi. Agarda tayyorlanayotgan saytdagi materiallar Web-dizaynerga notanish mavzuda bo'lsa, u holda o'sha sohadagi biror etuk mutaxassis bilan hamkorlikda ishlashga to'g'ri keladi.



Savol va topshiriqlar:

1. Web-uzel nima?
2. Chiziqli struktura nima?
3. Ierarxik struktura nima?
4. Ko'p bog'lanishli struktura nima?
5. Uzellar yaratishda materiallarni taqsimoti qanday tartibda amalga oshiriladi?

Web-saytlarni internetga joylash uchun manzil berish

Saytlarga internetda manzil berish uchun URL (Uniform Resource Locator) deb nomlanadigan ibora qo'llaniladi. URL-internetga murojaat qilishning eng oddiy usuli bo'lib, u manzilni ifodalaydi. URL manzildan ixtiyoriy sondagi foydalanuvchi bir vaqtning o'zida foydalanishi mumkin. URL manzili qat'iy format asosida yoziladi (8.15-rasm).

Протокол: // хост :порт / путь / файл #фрагмент

8.15-rasm. URL manzili strukturasi.

Manzilni to'liqroq tushunish uchun quyidagi misolni ko'rib chiqamiz:

<http://www.pedagog.uz/index.php>

bu URL manzili tarkibiy qismlari quyidagicha:

http – resursdan foydalanishda gipermatn bayonnomasi ishlatilayotganini bildiradi.

www.pedagog.uz - ma'lumotlar joylashgan Internet sahifa nomini bildiradi.

Index.php – faylning kompyuterdagi to'la nomini bildiradi.

Manzilning asosiy tashkil qiluvchilarini ko'rib chiqamiz:

- **Протокол** – bayonnoma. Internet – mijozlari va serverlarining bog'lanishi usulini tavsiflovchi qism. Odatda http (HyperText Transfer Protocol) gipermatnli uzatish maxsus bayonnomasidan foydalaniladi. Manzil kichik lotin harflari bilan yoziladi (<http://>).

- **Имя сервера.** Bunda joylanadigan ma'lumotlarni saqlaydigan hisoblash tizimining (bu hisoblash tizimini server deb nomlash qabul qilingan) nomi ko'rsatiladi. Serverga berilgan ta'rifda ham – tarmoq ishini nazorat qiluvchi kompyuter deb aytilganligi bejiz emas. Server nomi domen nomi kabi www bilan yoki usiz yozilishi mumkin. Masalan, www.macromedia.com. Boshqacha usulda nomni IP – adres ko'rinishida ifodalash mumkin. Masalan, 199.227.52.143.

- **Порт.** Port raqamini ko'rsatish bitta server ichidagi jarayonni adresatsiya qilish uchun talab etiladi. Biroz vaqtdan keyin port ishlatiladi va uning raqamini tushirib qoldirish mumkin.

- **Путь поиска.** Kataloglar va ostki kataloglar zanjiri, faylning joylashgan o'rnini belgilaydi. Bu zanjirning elementlari «/» simvoli bilan ajratiladi.

- **Файл.** Faylning nomi uning kengaytmasi bilan birga yoziladi (faqat kengaytma so'ralgan platformada). Agarda faylning nomi tushirib qoldirilgan bo'lsa, u holda Web-brauzer ko'rsatiladigan faylni qidiradi. Brauzer reaksiyasi va uning keyingi harakatlari faylning nomini kengaytirishga bog'liq bo'ladi. Masalan, grafik fayllari GIF yoki JPG ko'rish dasturiga tasvirlarning yuklanishini boshlab beradi. Arxiv fayllar (ZIP, RAR va boshqalar) bilan uyg'unlashgan kengaytirishlar – mijoz kompyuterida faylning saqlanishini boshlab beradi.

- **Фрагмент.** URL adresining fakultativ qismi. U # (panjara) maxsus simvoli bilan boshlanadi. Bu belgi gipermatnli hujjatning lavhasini ajratib ko'rsatuvchi maxsus belgidir. Tarkibga qo'shib qo'yish simvoli nomining kiritilishi butun sahifaga emas, balki uning ma'lum bir qismiga murojaat qilish imkonini beradi.

http bayonnomasi bo'yicha bog'lanish Internet tarmog'ida eng ko'p tarqalgan usul. Internetdagi ob'yektlarning manzillari aloqaning boshqa uslublarini va bayonnomalarini o'z ichiga olishi mumkin. Ulardan bir nechtasini sanab o'tamiz.

8.1-jadval.

Bayonnoma nomi	Yozish qoidasi	Qo'llanish sohasi
FTP	ftp://	Fayllarni uzatish bayonnomasi. FTR-serveri bilan bog'lanish uchun qo'llaniladi
File	file://	Kompyuter yoki lokal tarmoqda berilgan fayllarga bog'lanish bayonnomasi
Gopher	gopher://	Gopher serveriga bog'lanish bayonnomasi. Hozirgi vaqtda

		qo'llanilmaydi
JavaScript	javascript:	JavaScript tilida yozilgan dasturni chaqirish
Mailto	mailto:	Elektron pochta manzili
Newt	news:	Telekonferentsiyalarga bog'lanish bayonnomasi
Telnet	telnet:	kompyuterga bog'lanish bayonnomasi

Internetning bugungi ommaviylikini tushuntiruvchi sabablardan biri HTTP bayonnomasi imkoniyatlarining kengligidir. O'zidan avvalgi avlodlarining ba'zi bayonnomalarini u butunlay yutib yuboradi, ba'zilarini esa shunchaki siqib chiqaradi va ko'rgazmali grafik interfeys yordamida qo'llab-quvvatlash yo'li bilan yaxshiroq imkoniyat yaratadi.



Savol va topshiriqlar:

1. URL – nima?
2. Bayonnoma nima?
3. Server nomi nima?
4. Port nima?
5. Fayl nima?
6. Fragment nima?

Lokal uzal yaratish. Uzel kartasi

Dreamweaver dasturida saytlar va sahifalar yaratishning har xil yo'llari bor. Bulardan saytlarni yaratish bo'yicha eng mashhur texnologiyani ko'rib chiqamiz. Buning uchun quyidagi ketma-ketlikni bajaramiz:

- saytning lokal versiyasini yaratish;
- lokal versiyani Internet tarmog'iga joylash.

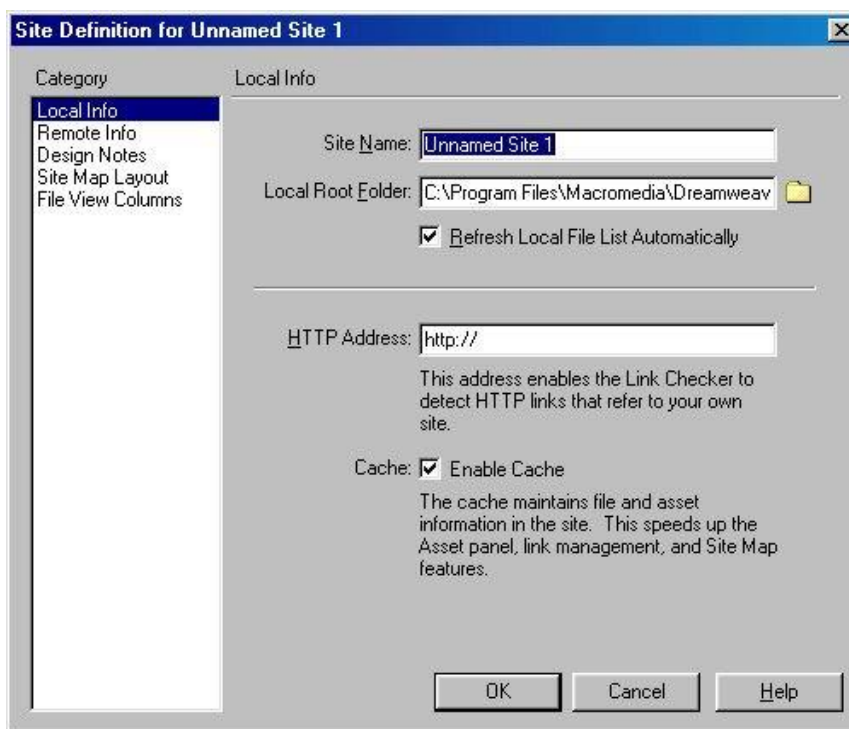
Lokal uzal - bu foydalanuvchi yoki foydalanuvchilarning kompyuterida joylashgan haqiqiy virtual Web-uzelning versiyasidir. Lokal uzal tarmoqdan ko'rinmaydi, unga faqat mualliflarigina bog'lana olishlari mumkin.

Lokal uzalning ma'lumotlari diskdagi belgilangan katalog ichida saqlanadi. Bu katalog vazifasini diskdagi har qanday katalog bajarishi mumkin. Yangi lokal uzal hosil qilish uchun quyidagi ketma-ketliklarni bajaring:

1. Site > New Site (Сайт > Создать) buyrug'ini bering. Ekranda 8.16-rasmda ko'rsatilgan Site Definition (Определение сайта) muloqot oynasi hosil bo'ladi.
2. Oynaning chap qismidan Local info (Информация о локальном узле) bo'limini tanlang. Maydonning qiymatini kiriting.
3. Remote Info (Сведения об удаленном сервере) kategoriyasiga sayt saqlanayotgan server haqidagi ma'lumotlarni kiriting.
4. Muloqot oynasining yuqorida sanalgan maydonlar to'ldirilishi shart. Site Map Layout (Планировка карты узла), Design Notes (Комментарии разработчика) va File View Columns (Колонки отображения файловых атрибутов) maydonlari to'ldirilishi shart emas (foydalanuvchi o'z xohishiga qarab to'ldirishi mumkin).

5. Barcha kerakli ma'lumotlar kiritilgach OK tugmasini sichqoncha yordamida bosing yoki ENTER tugmasini bosing.

Uzel kartasi – lokal uzel yaratishning qulay interfaol vositasi hisoblanib, unda ko'plab texnik qiyinchiliklarga ega bo'lgan topshiriqlar grafikli ko'rinishda engillik bilan amalga oshiriladi. Bu rejimda saytning kartasi tasvirlanadi. Uzel dispetcheri oynasi ikki qismga bo'lingan. Chap tomonda uzelnining kartasi kiritiladi (Kartada - sahifa belgilari HTML fayllarni ifodalaydi). Hujjatlar orasidagi gipermatnli bog'lanishlarni strelkalar ifodalaydi.



8.16-rasm. Lokal uzelnini sozlash.

Uzel kartasi

O'ng tomonda esa uzelnining fayllar va kataloglar ro'yxati beriladi. Kartaning ko'rinishi organizatsion diagrammani eslatib yuboradi. Kartada ikki darajani ko'rish mumkin. Yuqori darajani uzelnining asosiy sahifasi egallaydi. Quyi darajada esa unga tegishli va o'zaro giperbog'lanishga ega sahifalar tushuniladi. Uzelning gipermatnli hujjatlaridagi bog'lanishlar ko'k rangda tasvirlanadi. Agar bog'lanishda ko'rsatilgan fayl kengaytmasi .HTM, .HTML va boshqa shu formatdagilardan farq qilsa u holda yashil rangda tasvirlanadi. Agar bog'lanishda qo'pol xatoliklar bo'lsa, u holda bog'lanish qizil rangda ifodalanadi. Tashqi fayllar, boshqa fayllarda joylashgan fayllar va maxsus bog'lanishlar (masalan, mailto: yoki javascript:) globus ko'rinishidagi maxsus belgi oladi. Bulardan tashqari agar ko'k rangli belgi qo'yilgan bo'lsa — bu fayl foydalanuvchining o'zi tomonidan taqiqlanganini bildiradi. Qulf belgisi ostidagi belgi faqat o'qish uchun mo'ljallangan fayllarni bildiradi.



Savol va topshiriqlar:

-
1. Lokal uzal nima?
 2. Lokal uzal hosil qilish ketma-ketligini aytib bering?
 3. Uzal kartasi nima?

Hujjatlarga havolalar o‘rnatish va olib tashlash

Tegishli hujjatga havola o‘rnatish unchalik murakkab emas. Bunda foydalanuvchini chalkashtiradigan eng asosiy muammo – bu operatsiyani bajarish usul va uslublarining ko‘pligidir. Qachonki biror hujjatga yoki sahifaga havola o‘rnatmoqchi bo‘lsangiz, buning uchun quyidagi uch holatdan birini belgilash kerak. Ular bir-biridan faylning holati va yuklanish texnikasi bilan farq qiladi.

- Havola uchun ko‘rsatilgan hujjat uzalning hisobidagi ob‘yekt sanaladi. Ya‘ni faylning nomi fayllar va papkalar ro‘yxatidan iborat uzalning kartasi va panelida ifodalanadi. Bunday faylga havola o‘rnatish uchun asosiy hujjatni tanlash, sichqoncha yordamida marker ko‘rsatkichga bog‘lab olib, uni tegishli hujjatga olib o‘tish lozim. Marker-ko‘rsatkich — bu havola o‘rnatish uchun maxsus interfaol vosita hisoblanadi. Markerni uzal dispetcherining har qanday paneliga (kartaga yoki fayllar ro‘yxatiga) surib kelish mumkin.

- Butun hujjat saytning tarkibiga kiradi, ammo uning ro‘yxatidan o‘tmagan. Ya‘ni uning nomi o‘ng paneldagi fayllar ro‘yxatida mavjud, ammo u uzal kartasida yo‘q. Bunday holatlarda marker ko‘rsatkichni o‘ng panel ustiga surish va uni fayl nomi ustiga tashlash kerak.

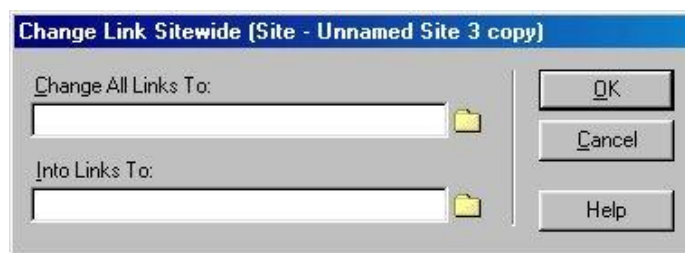
- Butun hujjat saytga kirmagan. Bunday holatda marker-ko‘rsatkichini surishni teskari yo‘nalishini qo‘llash kerak.

Havolani o‘zgartirish

Havolalarni o‘zgartirish uchun eski bog‘lanishga tegishli ko‘rsatkichlarni o‘chirib tashlab, yangi faylga tegishli ko‘rsatkichlarni kiritish mumkin. Ammo Dreamweaverda buni o‘zgartirish uchun maxsus vositalar mavjud:

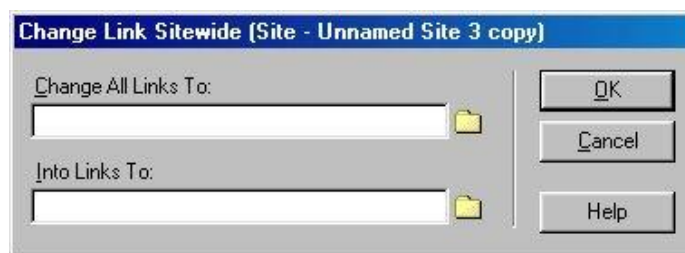
1. Uzal kartasidan hujjat belgisini tanlang;
2. Uzal dispetcheri oynasidan Site > Change Link (Сайт > Изменить ссылку) buyrug‘ini bering. Xuddi shu buyruqni - CTRL+L tugmalar kombinatsiyasi yordamida ham berish mumkin;
3. Select HTML Files (Выбор гипертекстовых файлов) muloqot oynasi ochiladi. Unda yangi faylning nomi va joylashgan katalogini ko‘rsatish kerak;
4. Fayl tanlangandan keyin, Dreamweaver dasturi Update Files (Обновление файлов) muloqot oynasi ochiladi. Barcha o‘zgarishlarni qo‘shish uchun Update (Обновить) tugmasini bosib.

Agar bir nechta hujjatlar bilan bitta hujjat o‘rtasida bog‘lanish o‘rnatmoqchi bo‘lsangiz buning uchun maxsus buyruq mavjud. Bu buyruq yordamida barcha ob‘yektlardagi havolalarni global tarzda o‘zgartirish mumkin. U barcha gipermatnli havolalardagi ko‘rsatilgan faylni nomini o‘zgartirib chiqadi.



8.17-rasm. Bog‘lanishni yangilash.

1. Uzel dispetcheri oynasini oching;
2. O‘zgartirilishi kerak bo‘lgan hujjatni oling;
3. Site > Change Link Sitewide (Сайт > Изменить связи по всему сайту) buyrug‘ini bering. Shu nomdagi muloqot oynasi ochiladi (8.18-rasm).



8.18-rasm. Havolani global o‘zgartirish.

1. Tanlangan faylning nomi Change All Links To (Изменить все ссылки на) maydonida ifodalanadi. Yangi faylning nomini Into Links To (На ссылки на) maydoniga kiritish yoki standart protseduralar yordamida tanlash mumkin.

2. OK tugmasini (кнопка) bosning yoki ENTER (клавиша) tugmasini bosning.

3. Update Files (Обновление файлов) muloqot oynasi ochiladi, bunda bog‘lanishga ega va ayni vaqtda yangilanishni talab etadigan, fayllar nomi aks etadi. Fayllarni tanlang va ularga tegishli havolani o‘rnating.

Change Link Sitewide (Изменить связи по всему сайту) buyrug‘i ham havolalarni ommaviy almashtirish uchun juda qulay. Masalan, bunday almashtirishlarni ko‘pincha saytning barcha sahifalaridagi elektron pochta adresini almashtirish uchun qo‘llash mumkin.

1. Change All Links To (Изменить все ссылки на) maydoniga o‘zgartirilishi kerak bo‘lgan havolani to‘liq kiriting.

2. Into Links To (На ссылки) maydoniga to‘liq yangi havolani kiriting.

3. OK tugmasini bosning.

Havolani olib tashlash

Havolani olib tashlashni bir nechta usullari mavjud. Masalan, havolaga tegishli, ya‘ni manba hisoblangan hujjat faylini ochish va uni oddiy yo‘l bilan o‘chirib tashlash mumkin. Bu usul balki noqulay tuyulishi mumkin, ammo maksimal aniqlikda bo‘ladi. Uzel dispetcheri orqali amalga oshiriladigan bog‘lanishni olib tashlash usulini ko‘rib chiqamiz:

1. Uzel dispetcheri oynasini oching;

2. Uzel kartasidan bog‘lanishni ko‘rsatuvchi hujjat belgisini oling;

3. Site > Remove Link (Сайт > Удалить ссылку) buyrug'ini bering. Bu buyruqning klaviaturada ikkita ekvivalenti bor: CTRL+SHIFT+L tugmalar kombinatsiyasi yordamida ham berish mumkin.

Sarlavhani o'zgartirish

Dreamweaver dasturi HTML hujjatining sarlavhasini uzal dispatcher oynasida almashtirish (o'zgartirish) imkonini beradi. Shuni esa tutish kerakki gipermatnli hujjatning sarlavhasini <title> deskriptori belgilaydi. Bu matn qatori hujjatning nomi hisoblanadi va qidiruv tizimlarida hujjatni indeksatsiyalashda muhim rol o'ynaydi. Eng mashhur Internet Explorer va Netscape Navigator brauzerlar sarlavhani dastur oynasining sarlavha qatoriga chiqarib beradi.

Sarlavhani o'zgartirish uchun quyidagilarni amalga oshirish kerak:

1. Uzel dispatcherini oching (ALT+F8).
2. View > Show Page Titles (Вид > Показать заголовки страниц) buyrug'ini bering. Bu buyruqni CTRL+SHIFT+T tugmalar kombinatsiyasi yordamida ham berish mumkin. Natijada uzal kartasida gipermatnli fayllar nomi o'rnida ularning sarlavhasi beriladi.
3. Hujjatni tanlang va File > Rename (Файл > Переименовать) buyrug'ini bering yoki klaviaturadan F2 tugmasini bosing.
4. Yangi sarlavhani kiriting va ENTER tugmasini bosing.



Savol va topshiriqlar:

1. Giperhavola (giperssылka) nima?
2. Gipermatn nima?
3. Ixtiyoriy hujjatga havola o'rnatish tartibini aytib bering?
4. Havolani o'zgartirish ketma-ketligini aytib bering?
5. Havolani olib tashlash ketma-ketligini aytib bering?
6. Sarlavhani o'zgartirish ketma-ketligini aytib bering?



3.4. Dreamweaver dasturida freymli tuzilma yaratish

Freymlar (frame) - bu gipermatnli sahifaning alohida mustaqil qismidir. Freymnlarni HTMLda hosil qilish haqida yuqorida gapirib o'tgan edik. Freymli saytlar quyidagi xususiyatlarga ega:

- Ishni to'g'ri tashkil qilinsa, yuklanadigan ma'lumotlar hajmida tejash imkonini beradi. Alohida freymlarda har xil fayllar yuklanishi mumkin.
- Bitta freymning tarkibiy qismini ko'rish vaqtida boshqa freym o'zining statistik holatini saqlab qoladi. Bu esa saytning muhim fragmentlarini ekranda har

doim ushlab turish imkonini beradi. Masalan, navigatsion tugmalar, logotiplar va boshqalar.

- Murakkab kompozitsiyali va birjinsli bo‘lmagan ko‘p bog‘li strukturali freymlar sayt yaratuvchisiga maydon o‘lchami va ob‘yektlar bog‘lanishini nazorat qilish imkonini beradi. Shuni alohida ta’kidlab o‘tish kerakki freymlar paydo bo‘lishi bilan Web-texnologiyada sanab o‘tilgan bir qator muammolar echimlarini topdi.

Ammo har qanday kuchli natija beruvchi vositalarning salbiy effektlari mavjud. Freymlar ham bu umumiy qoidadan holi emas. Ularning asosiy kamchiliklarini sanab o‘tamiz:

- Ular gipermatnli sahifalarning bir joydan boshqa joyga ko‘chirilishini keskin cheklab qo‘yadi. Ko‘rish dasturlarining birinchi versiyalari bunday loyihalar bilan tanish emas, shuning uchun freymlardan iborat sahifalarni brauzerlarning dastlabki avlodlari aks ettira olmaydi.

- Ko‘plab qidiruv tizimlari freymlarga bo‘lingan sahifalarni indeksatsiya qilmaydi. Qidiruv tizimlarida indeksatsiya qilish uchun sayt yaratuvchisidan qo‘shimcha dasturiy ta’minot talab qiladi.

- Bunday saytlarda ko‘rsatilgan gipermatnli bog‘lanish bo‘yicha o‘tish va yangi hujjatlarni har xil freymlarda ochishda ko‘rish dasturlarida adres qatori holati o‘zgarmaydi. Unda har doim asosiy o‘zak hujjat URL adresi tasvirlanadi.

Uzellar masshtabi o‘sishi bilan, gipermatnli sahifalar tarkibi murakkablashishi, yuqori chastotali yangilanish kontentida freymlarning kamchiliklari ikkinchi darajali hisoblanadi. Ko‘plab yirik korporatsiyalarning saytlari, informatsion portallar, yangiliklar bilan ishlaydigan kompaniyalarning Web-uzellari freymlar asosida qurilgan.

Freymlarni yaratish. Dreamweaver dasturida freymlarni yaratish uchun juda qulay uskunalar mavjud. Ularning yordamida har qanday murakkablikdagi freymlar strukturasi hosil qilish mumkin. Buning uchun quyidagi ketma-ketliklarni bajarish kerak:

1. Hujjatning freymli strukturasi yaratish. Bu etapda hujjatning freymlari soni, joylashuvi va o‘lchamlari belgilanadi;

2. Asosiy hujjat tarkibiga kiruvchi gipermatnli fayllarni saqlash;

3. Freymlarning parametrlarini belgilash. Bu etapda freymlarning nomlari beriladi, chegaralarini tasvirlashga oid ko‘rsatkichlar belgilanadi, prokrutkani o‘rnatish va boshqa ishlar amalga oshiriladi;

4. Freymlar strukturasi bilan hujjatning navigatsion tizimining mos kelishi, undagi mustaqil freymlar tarkibi va ularning joylashuvini belgilaydi. Bu butun freymli hujjatning fundamenti bo‘lib xizmat qiladi. Dasturda freymlarga bo‘lishning standart strukturalari to‘plami joylashgan. Uning tarkibiga mustaqil freymlarga bo‘lishning eng oddiy va taniqli variantlari kiradi. Foydalanuvchi standart to‘plamdan xohlagan strukturasi qo‘yishi mumkin. Agar berilgan variantlardan birortasi ham siz yaratmoqchi bo‘lgan strukturaga to‘g‘ri kelmasa, u holda o‘zingiz mustaqil ravishda struktura yaratishingiz mumkin. Buning uchun sodda strukturalardan bir nechtasini tegishli maydonda qo‘llash yetarli.

Standart freymli strukturalar kutubxonasi o'zida gipermatnli sahifalarni bo'lishning eng kerakli variantlarini birlashtiradi. Bundan tashqari dastur freymmlar o'lchamlarini interfaol ko'rsatishga imkon beradi va standart strukturalarni bir-birini ichiga joylaydi. Bu freymli strukturalar yaratishda eng kuchli manba hisoblanadi.

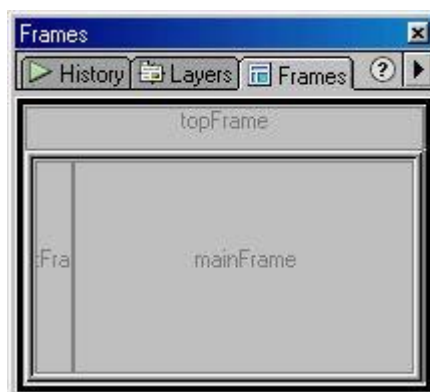
Kamdan-kam holatda yuqorida sanalgan imkoniyatlar yetarli bo'lmay qoladi. Ya'ni sayt yaratuvchisi o'zining shaxsiy xohishi va g'oyalari talabiga javob beradigan yagona freymli tizim yaratmoqchi. Buning uchun dasturda imkoniyat bor:

1. Freymmlarga bo'linishi kerak bo'lgan sahifaga matn kursorini qo'ying;
2. Freymmlarning chegaralari ko'rinishi uchun View > Visual Aids > Frame Borders (Вид > Служебная разметка > Границы фреймов) buyrug'ini bering. Shundan keyin hujjat oynasining ichki chegaralir 3 piksel qalinlikda paydo bo'ladi;
3. Ko'rsatkichni ixtiyoriy chegara ustiga olib boring va Alt tugmasini bosib turgan holatda hujjatning bo'sh qismiga suring. Bu paytda ko'rsatkich ikki strelkali ko'rinishga o'tadi;
4. Chegarani gorizontaal yoki vertikal surish yordamida o'zingizga kerakli freymli strukturani olishingiz mumkin.

Ichki chegarani olib tashlash uchun uni hujjat oynasidan tashqariga surib tashlang. Agar hujjat oynasida freymmlar chegaralari kesishgan joydagi nuqtasini ko'rsatkich yordamida surilsa, u holda hujjat oynasida gorizontaal va vertikal chiziqlar paydo bo'ladi. Bu natijani ketma-ket bajarib ham olish mumkin: avval gorizontaal chiziq qo'yiladi, keyin esa vertikal chiziq qo'yiladi (yoki teskarisi). Qaysi usul bilan bajarishdan kat'iy nazar natija bir xil bo'ladi. Demak strukturali hujjatning belgili kodlari ham bir xil bo'ladi.

Freymmlar palitrasi

Frames (Фреймы) palitrasi – bu hujjatning freymli strukturalarining miniatyurali tasvirlaridir. U o'z ichiga alohida tarkibli strukturalarni joylashuvi haqidagi ma'lumotni o'z ichiga oladi. Unda freymmlar nomlari va o'lchamlari beriladi. Palitra yordamida freymmlarga bo'lish ko'rinishi tanlanadi. Hujjatning har qanday freymli bo'lagi ustida odatdagi tahrirlash ishlarini olib borish, turli ob'ektlarni kiritish mumkin.



8.38-rasm. Frames palitrasi.

Bu palitrani ochish uchun Window > Frames (Окно > Фреймы) buyrug'ini bering yoki SHIFT+F2 tugmalar kombinatsiyasini bosish kerak.

Biror freymli ko'rinishni tanlash uchun uning Frames (Фреймы) palitrasidagi mos tasvirini sichqon chap tugmasi bilan bosish yetarli.

Hujjat oynasidan freymlarni tanlash

Hujjat oynasida freymlarni tanlashning barcha usullarini ko'rib chiqamiz:

1. Freymlar strukturasi alohida bo'limni olish uchun ALT tugmasini bosib turgan holda, sichqon chap tugmasini shu hududda bosib;

2. Strukturani butunlay belgilash uchun ixtiyoriy bo'lish chiziqlariga yoki chegara ramkasini sichqon chap tugmasi yordamida bosib;

Dastur shuningdek bir darajadagi (gorizontal bo'yicha) freymlarning hamda bir-birining ichiga solingan strukturalar va ularning elementlarining (vertikal bo'yicha) o'zaro bir-biriga o'tishini ta'minlaydi:

- o Bir darajadagi freymlar orasida o'tish uchun ALT+← va ALT+→ tugmalar kombinatsiyasini bosib;

- o Har xil darajadagi freymlar orasida o'tish uchun ALT+↑ va ALT+↓ tugmalar kombinatsiyasini bosib;

Tanlangan freymlar strukturasi bo'limi punktir chiziqlar bilan belgilanadi.

Chegaralar

Dreamweaverda strukturaning freymlari alohida ajralib turishi uchun chegarali chiziqlardan foydalaniladi. Bu chegara chiziqlari qalinligini olti pikselgacha belgilash mumkin. Property Inspector (Инспектор свойств) palitrası yordamida chegaralarning qalinligi, rangi kabi ko'rsatkichlarini o'rnatish mumkin. Chegara chiziqlarini tasvirlanishini palitraning ikkita bo'limi boshqaradi: Borders (Границы) va Border Width (Толщина границы). Birinchi bo'limi quyidagi uchta ro'yxatni o'z ichiga oladi. Bular Yes (Да), No (Нет), Default (По умолчанию). Yes(Да) bo'limi chegarani o'rnatishni ta'minlaydi. O'rnatilgan chegara uch o'lchamli ko'rinishda tasvirlanadi. No (Нет) bo'limi chegarani olib tashlaydi. Bu rejimda chegara oddiy chiziq ko'rinishida tasvirlanib, brauzer dasturlarida esa umuman ko'rinmaydi.

Keyingi bo'lim – bu chegara qalinligini pikselda belgilash uchun xizmat qiladi. Agar qalinlik 0 ga teng bo'lsa, u holda chegara ko'rish dasturlarida ko'rinmaydi. Chegara va ichki bo'luvchi chiziqlarni ranglarini o'zgartirish uchun palitraning Border Color (Цвет границы) bo'limidan foydalaniladi. Bu yerda rangni tanlash, Windowsning standart palitralarida rang tanlash kabi amalga oshiriladi.

Freymlar nomlari

Strukturaning mustaqil freymlariga alohida nom belgilash mumkin. Ammo bu nom berish shart degani emas. Ko'p hollarda o'ta zarur bo'lib qoladi, chunki u bitta tuzilma sohalari o'rtasida o'zaro kesishgan tayanishlar tashkil etish, hamda ko'rish dasturlari ichiga joylashtirilgan freymlardan tashkil topgan gipermatnli hujjatlarni aniq tahlil qilish imkonini beradi.

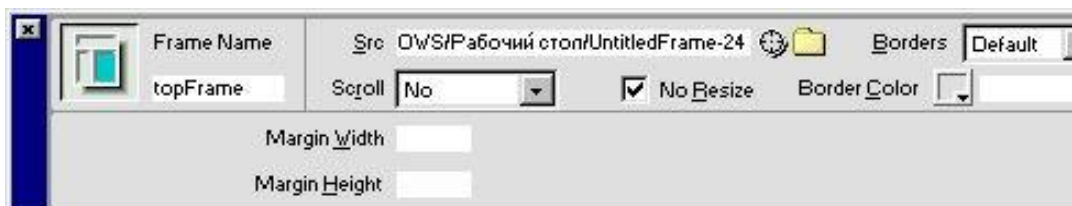
Freymga nom berish quyidagicha bo'ladi:

1. Freymni tanlang;

2. Property Inspector (Инспектор свойств) palitrasini oching. Buning uchun CTRL+F3 tugmalar kombinatsiyasini bosish kerak (agar u ekranda bo'lmasa);

3. Frame Name (Имя фрейма) maydoniga freym nomini kiriting (8.39-rasm). Buning uchun quyidagi qoidalarga amal qilish kerak:

- freym nomi bitta soʻzdan iborat boʻlishi va probel ishlatilmasligi kerak;
- barcha nomlar takrorlanmasligi kerak;
- `_blank`, `_parent`, `_self`, `_top` kabi nomlarni qoʻllash taqiqlanadi.



8.39-rasm. Frame Name - freym nomi.

Sahifalarga prokrutkalar oʻrnatish

Prokrutka tasmalari - bu har qanday foydalanuvchiga yaxshi tanish boʻlgan Windowsning boshqaruv elementidir. Uning yordamida oʻlchami katta boʻlmagan oynalarda katta hujjatlarni koʻrish mumkinligi hammamizga maʼlum. Bu boshqaruv elementi baʼzan skrolling ham deb yuritiladi.

Freymli gipermatnli hujjatlarning ajoyib xususiyatlaridan biri bu alohida freymga prokrutka qoʻyilishidir. Property Inspector (Инспектор свойств) palitrasining Scroll (Прокрутка) boʻlimi yordamida prokrutkani koʻratkichlarini belgilash mumkin:

- Default (По умолчанию). Prokrutkani aks ettirish zarurati haqidagi masalani hal qilishni brauzer dasturi ixtiyoriga qoldiradi;
- Yes (Да). Freymli hudud holatidan qatʼiy nazar prokrutkani oʻrnatish;
- No (Нет). Prokrutkani chiqishini taqiqlash;
- Auto (Автоматически). Avtomatik chiqish rejimi. Bu holatda freym oʻlchamidan kiritilayotgan maʼlumotlar ortib ketsa prokrutka ekranga avtomatik ravishda chiqariladi.



Savol va topshiriqlar:

1. Prokrutka nima?
2. Prokrutka qoʻyish ketma-ketligini aytib bering?
3. Frame nima?
4. Sahifalarni freymlarga boʻlish ketma-ketligini aytib bering?
5. Chegaralar qanday oʻrnatiladi?
6. Freymnlarni nomlash ketma-ketligini aytib bering?

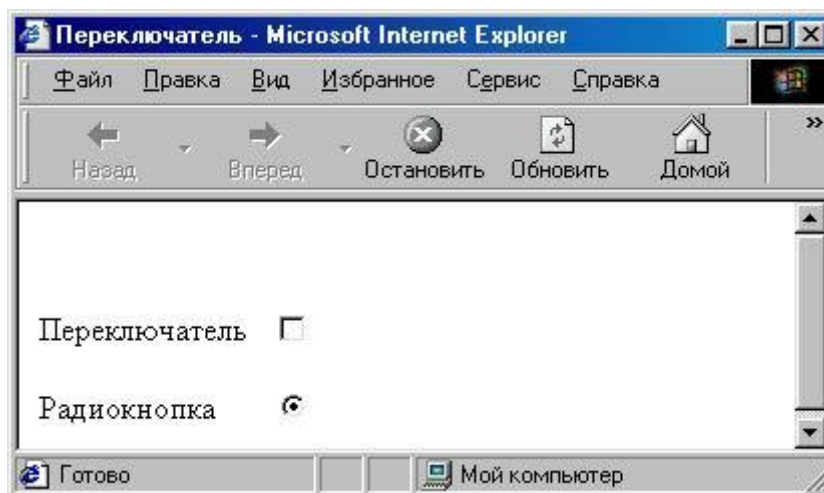


3.5. Dreamweaver dasturida elektron qoʻllanma yaratish

Foydalanuvchining Web-sahifalarda tanlash mexanizmi asosan ikki tipdagi boshqaruv elementlari yordamida amalga oshiriladi. Bular: qayta ulagichlar (radio button) va bayroqchalar (check box). Birinchisi keltirilgan bitta variantni olish

imkonini beradi. Ikkinchisi esa ko‘pchilik variantlardan alternativ tanlash imkonini beradi (8.31-rasm).

Bu elementlar nafaqat qo‘llanilishida yaqinlik, balki ularni yaratishda qo‘llaniladigan belgilash kodlari ham juda bir-biriga o‘xshash. HTML tilida bu elementlarni <input> deskriptori turli atributlari bilan qo‘llanib hosil qilinadi.



8.31-rasm. Qayta ulagich.

Masalan, bayroqcha

```
<input type="checkbox" name="bayroqcha_nomi" value="checkbox">
```

qayta ulagich esa

```
<input type="radio" name="Qayta ulagich_nomi" value="radiobutton">
```

qatorda ko‘rsatilgan belgili kodlar yordamida yaratiladi.

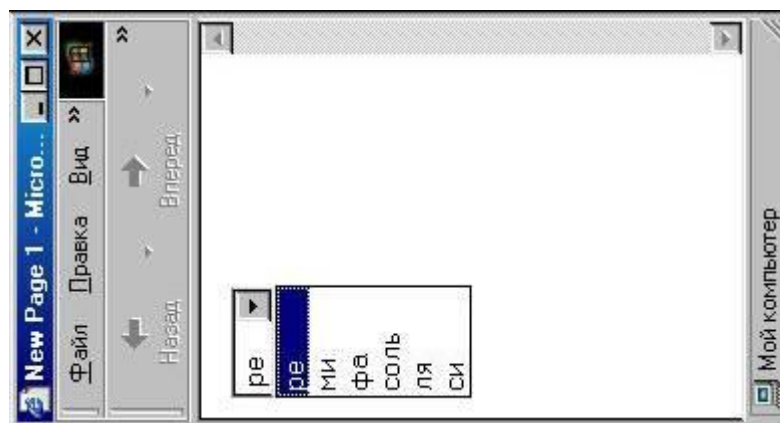
Qayta ulagichni qo‘yish uchun Insert > Form Objects > Radio Button (Вставка > Объекты форм > Переключатель) buyrug‘ini berish kerak. Bayroqcha esa Insert > Form Objects > Check Box (Флажок) buyrug‘ini berish bilan hosil qilinadi. Undan tashqari ikkala ob‘yektни ham ob‘yektlar palitrasida joylashgan Form (Форма) bo‘limining tugmalari yordamida ham qo‘yish mumkin.

Qayta ulagichlar alohida element hisoblanadi. Shuning uchun ham ularning har biri alohida nomga ega bo‘lmog‘i lozim. Buning uchun xossalar panelining Check Box (Флажок) maydoniga nom kiritiladi.

Ro‘yxatlar va menyular

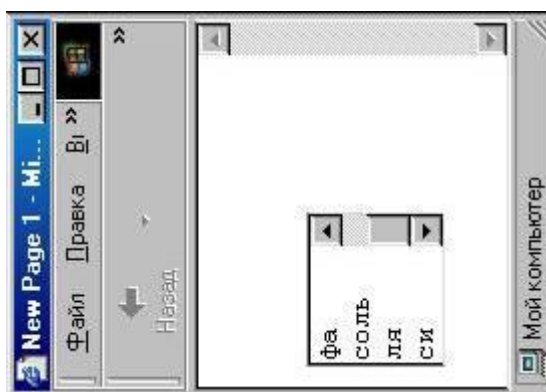
Ro‘yxatlar va menyular – bu formaning tizimli elementi hisoblanadi. Ular yordamida keltirilgan variantlardan birini tanlash imkoniyati mavjud.

Masalan, Windowsning har qanday standart menyusida uning elementlaridan faqat biri tanlanishi hammamizga ma‘lum.



8.32-rasm. Bekiladigan ro‘yxat.

Bekiladigan ro‘yxatlar bu - oddiy ro‘yxatdan ko‘ra ko‘proq tejamkor boshqaruv elementi hisoblanadi. Chunki oddiy ro‘yxatlar gipermatnli sahifada ko‘p joyni egallab qo‘yadi.



8.33-rasm. Prokrutkali ro‘yxat.

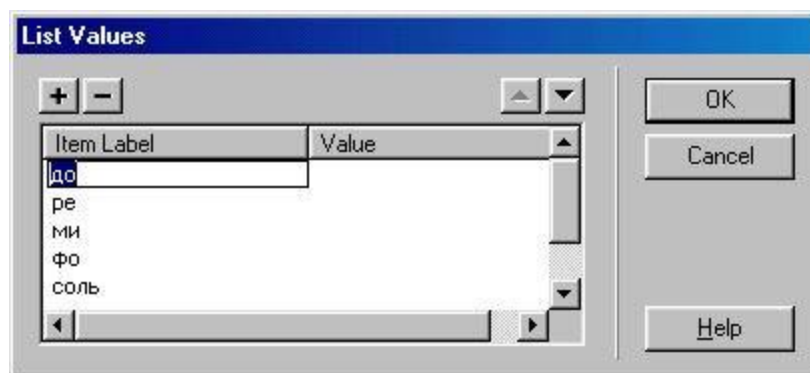
Menyu yaratish

HTML tilida menyu va ro‘yxatlar <select> deskriptor juftligi yordamida kodlashtiriladi. U o‘z ichiga menyuning punktlari yoki buyruqlarini belgilaydigan <option> tegini oladi. Masalan, navbatdagi kod fragmenti ikki punktdan iborat oddiy menyuni hosil qiladi:

```
<select name="Menyu_nomi">
<optionvalue="major">do</option>
<option value="minor">pe</option>
</select>
```

Menyuni quyidagi yo‘l bilan hosil qilish mumkin:

1. Matn kursorini hujjat ichiga qo‘ying;
2. Insert > Form Objects > List/Menu (Вставка > Объекты формы > Список/Меню) buyrug‘ini bering;
3. Property Inspector (Инспектор свойств) palitrasini oching (CTRL+F3);
4. List/Menu (Список/Меню) maydoniga ob‘yekt nomini kiriting;
5. List Values (Пункты списка) tugmasini bosib, shu nomdagi muloqot oynasini oching. Bu muloqot oynasi menyu punktlarini yaratish uchun mo‘ljallangan (8.34-rasm).



8.34-rasm. List Values muloqot oynasi.

6. List Values (Пункты списка) muloqot oynasi vositalari yordamida menyularni tizimini yarating. Menyuning har bir punkti (Label) bilan nomlanadi. «+» belgisi ostidagi tugma menyuga yangi punkt qo‘shish uchun ishlatiladi. «-» belgisi ostidagi tugma tanlangan punktni o‘chiradi. Strelkali tugmalar esa menyularni ketma-ketligini o‘zgartiradi. OK tugmasini bosish bu muloqot oynasi bilan ishlashni tugallaydi.

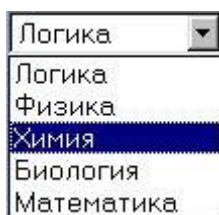
7. List/Menu (Список/Меню) maydoniga ob‘yekt nomini kiriting;

8. Initially Selected (Первоначальный выбор) ro‘yxatiga sahifa yuklanganda menyuda tasvirlanadigan punktni ko‘rsating;

Menyuning har bir bo‘limi ikkita identifikator bilan yoziladi: nomi (Label) va qiymati (Value). Bu yerda nom menyularni qismini bildiradi.

Bekiladigan ro‘yxat

Bekiladigan ro‘yxat - bu boshqaruvning eng muhim elementlaridan biri. U gipermatnli buyruqlarni o‘z ichiga olgan bekiladigan ro‘yxatdir. Bunday imkoniyat tashqi hujjatlar va fayllar bilan bog‘lanishni ta‘minlab beradi. Bekiladigan ro‘yxatga misol 8.35-rasmida ko‘rsatilgan.



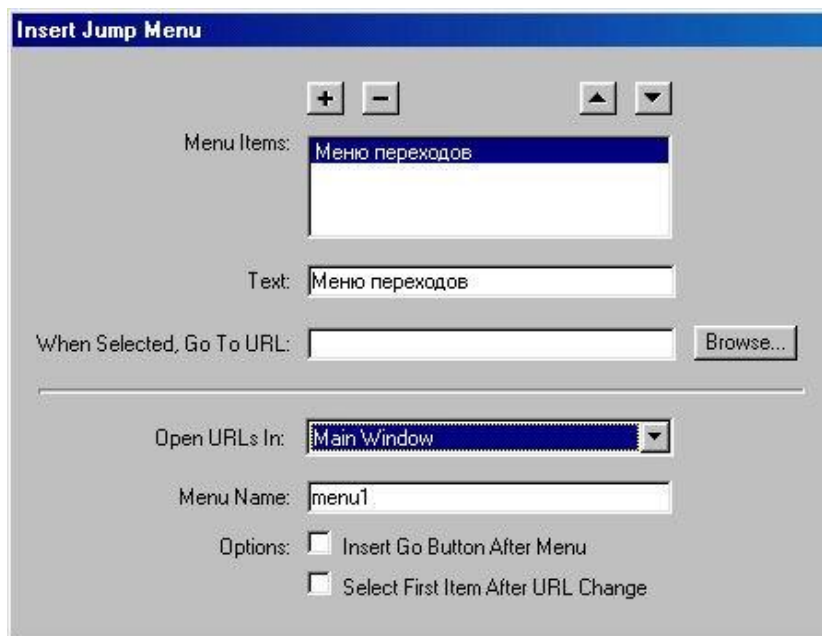
8.35-rasm. Bekiladigan ro‘yxatga misol.

Endi o‘tish menyusini qo‘yish ketma-ketli bilan tanishib chiqamiz:

1. Hujjatning menyularni qo‘yiladigan qismiga kursorni qo‘ying.
2. Insert > Form Object > Jump Menu (Вставка > Элементы форм > Меню переходов) buyrug‘ini bering yoki Objects (Объекты) palitrasining Forms (Формы) qismida joylashgan Jump Menu (Меню переходов) tugmasini bosing. Natijada Insert Jump Menu (Вставка меню переходов) muloqot oynasi ochiladi (8.36-rasm). Bekiladigan ro‘yxatni yaratish uchun mo‘ljallangan barcha amallar shu oynada bajariladi. Uning boshqaruv elementlarini ko‘rib chiqamiz:

- Menu Items (Пункты меню). Bu bo‘limda bekiladigan ro‘yxatning barcha bo‘limlari ko‘rsatiladi;
- Text (Текст). O‘tish menyusining bo‘limlariga matn kiritish.

- When Selected Go To URL (При выборе перейти по адресу). URL manzilini, bogʻlangan hujjat yoki fayl nomini toʻliq kiritiladi.
- Open URLs in (Открыть связанный документ в) bu roʻxatdan butun hujjatning yuklanish xududini (oyna yoki freym) belgilanadi;
- Menu Name (Имя меню). Matn maydoni bekiladigan roʻxatning nomini belgilaydi. Bu maydonni toʻldirmasa ham boʻladi;



8.36-rasm. Insert Jump Menu muloqot oynasi.

- Insert Go Button After Menu (Вставить кнопку перехода после меню) – agar bu bayroqcha qoʻyilsa, u holda menyu yonida Go (Перейти) tugmasi paydo boʻladi. Bu tugmani bosish bilan bekiladigan roʻxat ishga tushadi.
 - Select First Item After URL Change (Выбрать первый пункт после изменения адреса) - bu bayroqcha oʻrnatilsa, u holda bekiladigan roʻxatda yangi sahifaga oʻtgandan keyin har doim birinchi boʻlim tanlanadi.
3. Ketma-ketlikni OK tugmasini bosish bilan tugallang.

Roʻxatlar hosil qilish

Yuqorida roʻxatlar va menyular bir xil belgili kodlar asosida yaratilishi, ularni qoʻyishni umumiy buyrugʻi borligi haqida maʼlumot oldik. Quydagi buyruqni berish natijasida ham roʻxat yoki menyu qoʻyish mumkin. Buning uchun Insert > Form Objects > List/Menu (Вставка > Объекты формы > Список / Меню) buyrugʻini berib, tegishli koʻrsatkichlar tanlansa roʻxat yoki menyu qoʻyiladi. Buni belgilash uchun xossalar panelidan foydalaniladi. Roʻxat hosil qilish uchun Property Inspector (Инспектор свойств) palitrasida List (Список) bayroqchasini bosish kerak.

Boʻsh roʻxatni maʼlumotlar bilan toʻldirish uchun List Value (Пункты списка) tugmasi bosiladi. Natijada 8.37-rasmda koʻrsatilgan muloqot oynasi paydo boʻladi. Shu muloqot oynasiga boʻlimlar nomlari va qiymatlari kiritiladi.

Property Inspector (Инспектор свойств) palitrasida roʻxatlarning bir qancha qoʻshimcha koʻrsatkichlari joylashgan. Ularni koʻrib chiqamiz:

- List/Menu (Список/Меню) - bu yerda ro'yxatning nomi beriladi.
- Height (Высота) - bir vaqtning o'zida ro'yxatda tasvirlanadigan qatorlar soni.
- Selections: Allow multiple (Множественный выбор) - agar bu bayroqcha o'rnatilgan bo'lsa u holda ro'yxatdan bir vaqtning o'zida bir nechta punktni olish mumkin.
- Initially Selected (Первоначальный выбор) - sahifa yuklanganda ro'yxatda birinchi chiqadigan punktni belgilaydi.



8.37-rasm. Choose Spacer Image muloqot oynasi.

Tugmalar bilan ishlash

Ish jarayonida tugmalarga har qanday qo'shimcha funktsiyalar yoki buyruqlarni bog'lash mumkin. HTML gipermatnli belgilash tilida tugmalarni `<input>` deskriptori yordamida qo'yiladi. Navbatdagi kodli fragment tugmaga misol bo'la oladi:

```
<input type="reset" name="tugma_nomi" value="Reset">
```

Tugmani qo'yish uchun Insert > Form Objects > Button (Вставка > Объекты формы > Кнопка) buyrug'i beriladi.

Tugmalar ustida olib boriladigan asosiy tahrirlash ishlari Property Inspector (Инспектор свойств) palitrasida olib boriladi. Palitraning tegishli ko'rsatkichlarini sanab o'tamiz:

- Button Name (Имя кнопки). Tugma nomi.
- Label (Надпись). Tugmadagi yozuvni belgilash.
- Reset form (Очистить форму). Forma maydonini tozalash. Bu tugmani bosish bilan formaning matn maydoniga kiritilgan barcha ma'lumot o'chiriladi.
- Submit form (Отправить форму). Tugma bosilishi bilan forma serverga uzatiladi.

Nostandart funktsiyali tugmalarni qo'lda tahrirlashga to'g'ri keladi. Buning uchun JavaScript tilidan foydalanishga to'g'ri keladi. Quyida ana shunday nostandart funktsiyali tugmani ifodalovchi qator ko'rsatilgan:

```
<input type="button" name="Tugma_nomi" value="yes" onClick="doFunction()">
```

Grafikli tugmalar

Dasturda sahifalar bilan ishlash jarayonida dasturchi faqatgina standart tugmalardan foydalanmasdan, balki o'zi yaratgan ixtiyoriy rastrli tasvirdan tugma sifatida foydalansa bo'ladi. Unga ham biror buyruqni birlashtirish imkoniyati mavjud:

1. Grafikli tugma qo'yilishi kerak bo'lgan joyga kursorni qo'ying;
2. Insert > Form Objects > Image Field (Вставка > Объекты формы > Графическое поле) buyrug'ini bering. Select Image Source (Выбор изображения) muloqot oynasi hosil bo'ladi;

3. Bu oyna yordamida kerakli rasmni oling. Grafikli asosdagi tugma hosil qilindi. Endi unga dasturning biror funktsiyasini bog'lash kerak. Ayrim oddiy funktsiyalarni Behaviors (Режимы) (Shift+F3) palitrasida ko'rishimiz mumkin.

Grafikli tugmani ko'rsatkichlarini belgilash uchun xossalar palitrasidan foydalanishimiz mumkin.



Savol va topshiriqlar:

1. Menyu hosil qilish uchun qaysi deskriptordan foydalaniladi?
2. Ro'yxatlar hosil qilish ketma-ketligini aytib bering?
3. Bekiladigan ro'yxat qanday hosil qilinadi?
3. Grafikli tugmalar hosil qilish ketma-ketligini ayting?



3.6. Dreamweaver dasturida sayt dizayni bilan ishlash

Grafik tasvirlar ishtirokisiz tuziladigan hujjatlar juda kamchilikni tashkil qiladi. Ayniqsa Web – sahifalarini rasmlar, logotiplarsiz tasavvur ham qilib bo'lmaydi. Bu kabi grafikli ma'lumotlarning turlari juda ham ko'p. Ularni faqat formati bo'yicha ajratish mumkin. Masalan, birgina rasm degan ma'lumot ostida har xil formatdagi fayllar yotadi. Faylning formati deganda ma'lumotni yozish va saqlash uchun belgilangan standart qoidalar to'plami tushuniladi. Hozirda o'nlab grafikli formatlar mavjudki, ular turli tizimlarda har xil maqsadda qo'llaniladi. Grafikli formatlar uchta sinfga bo'linadi. Bular: rastrli, vektorli, metafaylli. Bu formatlar haqida keyingi boblarda batafsil ma'lumot berilgan.

Dasturning rasmlarni qo'yish imkoniyatlari

Rasmlar Web-sahifalarga jilo beruvchi vositalar hisoblanadi. Rasmlarni qo'yish – bu oddiy texnik operatsiya. Dasturda uni qo'yishning bir nechta yo'llari bor. Avval rasmni qo'yish buyruqlaridan birini tanlaymiz:

- Objects (Объекты) palitrasini oching, Common (Общие) bo'limiga o'ting, palitraning yuqori - chap qismida joylashgan Insert Image (Вставка изображения) tugmasini bosing.

- Insert > Image (Вставка > Изображение) buyrug'ini bering.
 - CTRL+ALT+I tugmalar kombinatsiyasidan foydalaning.
 - Tasvirni Assets (Фонды) palitrasidan hujjat oynasining kerakli qismiga ko'chiring.
 - Saytning dispatcher oynasidan rasmning belgisini olib hujjatning oynasiga tashlang.
 - Rasmni ish stolidan hujjat oynasining kerakli joyiga tashlang.
-

HTML tilida esa bu vazifalarni deskriptori bajaradi. Bu deskriptor rasmning barcha ko'rsatkichlarini belgilaydigan bir nechta atributlarni o'z ichiga oladi. Quyidagi misolda deskriptor o'ziga to'qqizta atributni bog'lagan:

```

```

Shuningdek qo'yilgan rasimga ramka ham qo'yish mumkin:

```

```

Bu misolda border atributi ramkaning qalinligini bildiradi.

Rasmni fon sifatida qo'yish

Biz yuqorida gipermatnli hujjatning parametrlarida fonga rang tanlash, fonga rasm qo'yish bilan tanishib chiqdik. Endi rasmlarni fon sifatida qo'yish bilan batafsil tanishib chiqamiz:

1. Modify > Page Properties (Изменить > Свойства страницы) buyrug'ini bering.

2. Background Image (Фоновый рисунок) maydonida rasm faylini ko'rsating.

Xuddi shu vazifani HTML tilida <body> deskriptori yordamida bajariladi. Masalan, <body background="image.jpg"> bunda rasm hujjat saqlangan papkada joylashgan.

```
<body background="www.images.com/source/back.gif">
```

 fragmenti esa internetning ko'rsatilgan manzilidagi rasmni fon sifatida qo'yib beradi.

Interfaol rasmlar

Hozirgi kunga kelib rasmlar va animatsiyalarsiz sahifalarni uchratish juda qiyin. Aksariyat Web-sahifalar animatsiyalar, animatsion tugmalar yoki harakatlanish tugmalari, rasmlar, interfaol rasmlar bilan boyitilgan.

Interfaol rasmlar so'zi rollover image iborasidan olingan bo'lib so'zma-so'z tarjimada "tasvirni ko'chirib o'tkazish" ma'nosini bildiradi. Agar Web-sahifada biror rasm ustiga sichqon ko'rsatkichini olib borganingizda boshqa rasm paydo bo'lsa bilingki u rasm interfaol rasmdir. Interfaol rasmlarni qo'yish quyidagi yo'l bilan bajariladi:

1. Matn kursorini rasm qo'yiladigan joyga qo'ying.

2. Insert > Interactive Image > Rollover Image (Вставка > Динамические изображения > Интерактивное изображение) buyrug'ini bering. 8.23-rasmda ko'rsatilgan muloqot oynasi hosil bo'ladi.

3. Bu muloqot oynasining tegishli maydonlarini to'ldiring:

o Image Name (Имя изображения) maydonda rasmning nomi ko'rsatiladi.

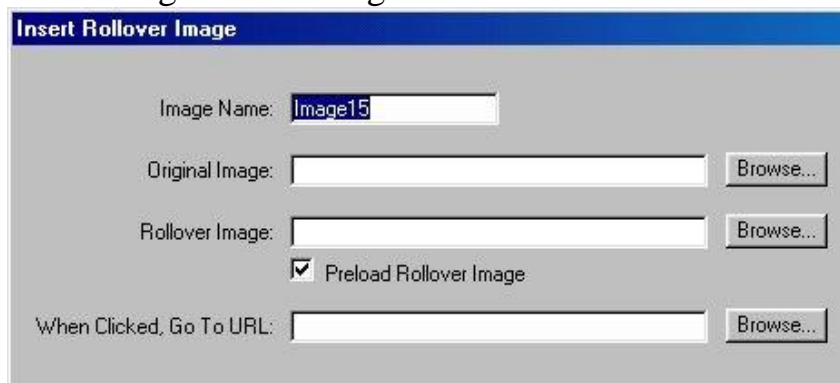
o Original Image (Исходное изображение). Joriy holatda ko'rinadigan rasmning nomi ko'rsatiladi.

o Rollover Image (Интерактивное изображение). Sichqon ko'rsatkichi kelishi bilan almashinadigan rasmni nomini ifodalaydi.

o Preload Rollover Image (Предварительная загрузка) - bu bayroqcha rasmni ko'rish dasturlarida yuklash rejimini ishga tushiradi.

o When Clicked Go To URL (переход по адресу) - bu yerda gipermatnli o'tish buyrug'i saqlanadi.

4. Tegishli ko'rsatkichlar kiritilgandan so'ng OK tugmachasini bosib yoki klaviaturadan ENTER tugmasini bosib.



8.23-rasm. Interfaol rasmni qo'yish.

Dreamweaver dasturi rasmlarning dinamik almashinuvini ko'rsatmaydi. Bu jarayonni faqat sahifa ko'rish dasturlarida ko'rish mumkin. Brauzer dasturlarini chaqirishning eng qisqa yo'li F12 tugmasini bosishdir.



Savol va topshiriqlar:

1. Rastrli formatlar nima?
2. Vektorli formatlar nima?
3. Metafayllar nima?
4. Sahifaga rasm qo'yish ketma-ketligini aytib bering?
5. Rasmni fon sifatida qo'yish uchun qanday ketma-kelikni bajarish kerak?
6. Interfaol rasmlar nima?

Jadvallar qo'yish va ular bilan ishlash

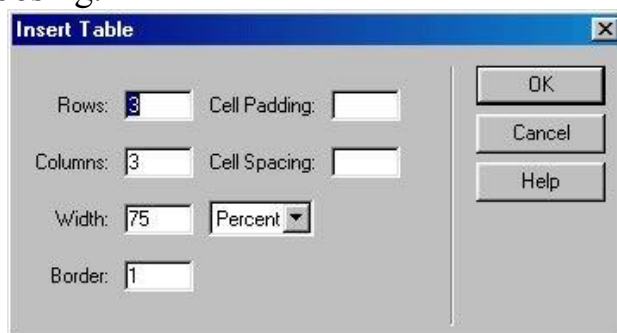
Jadvallar - matnli va grafikli ma'lumotlarni saqlash uchun tayyor forma hisoblanadi. Bundan tashqari sahifalarda rasmlarni birlashtirish uchun, matn ostiga rasm qo'yish uchun ham jadval kataklaridan foydalansa bo'ladi. Qatordagi ikkita yonma-yon joylashgan katakchalardan kolonka sifatida foydalansa bo'ladi. Bir katakli jadvaldan tortib to murakkab jadvalgacha – bularning barchasi qator, ustun, katakchalarning oddiy kombinatsiyasidan tuzilgan.

HTML tilida jadvalga tegishli barcha amallar yoki ma'lumotlar `<table>` `</table>` deskriptorlar juftligi orqali ifodalanadi. Bu deskriptorning ko'plab atributlari mavjud. Bu atributlar yordamida jadvalning eni, foni rangi, chegarasi, tekislash, maydon, kataklar orasidagi masofa va boshqa ko'rsatkichlarni belgilash mumkin. Biz yuqorida HTMLda jadvallarni qo'yishning barcha imkoniyatlari bilan tanishib chiqdik. Endi esa Dreamweaverda jadval qo'yish bilan tanishib chiqamiz.

Jadval qo'yish

Jadval qo'yish - bu uncha qiyin bo'lmagan texnik jarayondir. Dreamweaverda jadval qo'yish uchun quyidagi ketma-ketlik bajariladi:

1. Jadval quyilishi kerak bo'lgan joyga kursorni qo'ying.
2. Insert > Table (Вставка > Таблица) buyrug'ini bering. Bu buyruqni bajarishning alternativ variantlari ham bor:
 - o CTRL+ALT+T tugmalar kombinatsiyasi;
 - o Object (Объекты) palitrasidagi Common (Общие) bo'limining Insert Table (Вставить таблицу) tugmasini bosish bilan;
3. Insert Table (Вставка таблицы) muloqot oynasi maydonlarini to'ldiring (8.24-rasm). Bu oynaning boshqaruv elementlarini ko'rib chiqamiz:
 - o Rows (Строки). Jadvalning qatorlari soni.
 - o Columns (Столбцы). Jadvalning ustunlari soni.
 - o Width (Ширина). Jadvalning eni.
 - o Border (Рамка). Har bir katakchanning va butun jadvalning chegara ramkalari eni. Piksellarda beriladi.
 - o Cell Padding (Поля ячейки). Katakcha chegarasi va ichidagi ma'lumot orasidagi maydon kattaligi.
 - o Cell Spacing (Расстояние между ячейками). Kataklar orasidagi masofa.
4. **OK** tugmasini bosish.



8.24-rasm. Insert Table muloqot oynasi.

Ixtiyoriy ko'rsatkichli jadval qo'yilgandan keyin, uni to'ldirish vaqtida yoki undan keyin ham o'zgartirish mumkin.

Jadvalli ma'lumotlarni import qilish

Dreamweaver dasturida tashqi matnli fayllarni o'qish va undagi jadvallarni o'girib olish uchun maxsus buyruqlar mavjud. Jadvalli ma'lumotlarni import qilish quyidagicha amalga oshiriladi:

1. File > Import > Import Tabular Data (Файл > Импорт > Импорт табличных данных) buyrug'ini bering.
2. Import Table Data (Импорт табличных данных) muloqot oynasi hosil bo'ladi (8.28-rasm). Bu muloqot oynasining asosiy boshqaruv elementlari bilan tanishib chiqamiz:
 - o Data File (Файл данных). Bu maydonga jadvalli ma'lumot joylashgan faylni qidirish uchun to'liq yo'l ko'rsatiladi.
 - o Delimiter (Разделитель). Bu ro'uxatda matnli fayldagi jadvalni bo'luvchisi kiritiladi. Ro'uxat o'z ichiga quyidagi bandlarni oladi: Tab (Символ табуляции),

Comma (Запятая), Semicolon (Точка с запятой), Colon (Двоеточие), Other (Другой).

- Fit to Data (Подбирается по данным). Agar bu qayta ulagich oʻrnatilgan boʻlsa, jadval kengligi har bir ustunda eng uzun qator oʻlchami boʻyicha tanlab olinadi.

- Set (Равна). Agar bu qayta ulagich oʻrnatilgan boʻlsa, u holda jadvalning enini oldindan belgilash mumkin boʻladi.

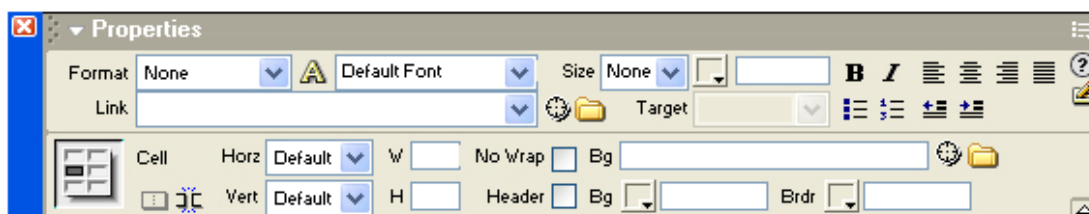
- Cell Padding (Поля ячейки). Katakcha tashkil etuvchilari va chegarasi orasidagi masofani belgilaydi.

- Cell Spacing (Расстояние между ячейками). Jadvalning yonma-yon turgan kataklari orasidagi masofani belgilaydi.

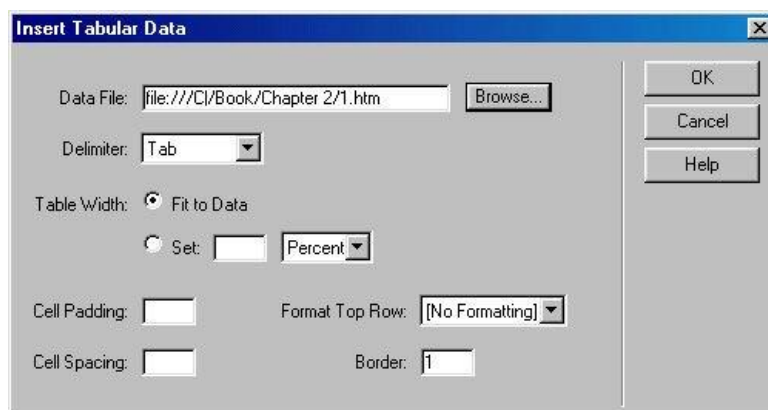
- Format Top Row (Форматирование верхней строки). Jadvalning birinchi qatorini formatlash uchun toʻrtta variantni taklif qiladi.

- Border (Рамка). Bu yerda jadvalning chegaraviy ramkasining qalinligi belgilanadi. Agar ramka zarur boʻlmasa u holda bu maydonda nol koʻrsatish kerak.

3. OK tugmasini bosib.



8.25-rasm. Property Inspector palitrasi.



8.26-rasm. Jadvalli maʼlumotlarni import qilish.

Agar muloqot oynasida koʻrsatilgan jadval tashkil qiluvchilari va matnli fayldagi jadval tashkil qiluvchilariga toʻgʻri kelmasa, u holda jadvalni import qilish notoʻgʻri bajariladi. Aniqroq aytganda jadval buzilgan holda koʻchadi.

Formatlash

Jadvalni yaratish bilan birga ularning tashqi koʻrinishlarini belgilash, formatlash operatsiyalarini bajarish mumkin. Dasturda esa bu atributlarning aksariyat qismi Property Inspector (Инспектор свойств) palitrasi yordamida oʻrnatiladi (8.25-rasm). Bu palitrani ochish usullari bilan tanishib chiqamiz:

- Window > Properties (Окно > Свойства) buyrugʻi bilan;

-
- CTRL+F3 tugmalar kombinatsiyasini bosish bilan;
 - Kontekstli menyudan Properties (Свойства) buyrug‘ini berish bilan.
- Palitraning barcha uskunolari jadvalni tahrirlash va ko‘rsatkichlarini belgilash uchun mo‘ljallangan. Endi palitraning boshqaruv elementlari bilan tanishib chiqamiz:

- Table Name (Имя таблицы) — jadval nomi;
- Rows (Строки) — qatorlar soni;
- Cols (Столбцы) — ustunlar soni;
- W (Ширина) — jadvalning eni;
- N (Высота) — jadvalning balandligi;
- CellPad (Поля) — katakcha maydoni;
- CellSpace (Интервалы) — katakchalar orasidagi masofa;
- Align (Выравнивание) — jadvalni tekislash;
- Border (Рамка) — katakcha va jadvalning ramkasi ko‘rinishi;
- Bg Color (Фоновый цвет) — fon rangi;
- Bdr Color (Цвет рамки) — katakcha va jadvalning ramkasi rangi;
- Bg Image (Фоновый рисунок) — fon sifatida qo‘yiladigan rasm;

Jadvalga qator va ustunlar qo‘shish

Dasturda qator va ustunlarni qo‘shishning bir nechta texnik usullari bor. Ularga menyu buyruqlari, klaviaturada tugmalar kombinatsiyasi va palitra bilan ishlash operatsiyalari kiradi.

Qator yoki ustun qo‘shish quyidagicha amalga oshiriladi:

1. Jadvalning katakchasiga matn kursorini quyting. Bu katakcha yangi qator yoki ustun qo‘yishda mo‘ljal vazifasini o‘taydi. Yangi qator qo‘yilsa katakcha ostidan, ustun esa katakchanning chap tomonidan qo‘yiladi.

2. Qo‘yish buyrug‘ini bering. Bular quyidagi buyruqlar:

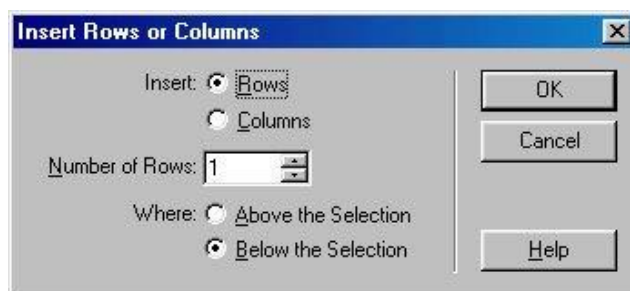
- Modify > Table > Insert Row (Изменить > Таблица > Вставить строки), Modify > Table > Insert Column (Изменить > Таблица > Вставить столбец), Modify > Table > Insert Rows or Columns (Изменить > Таблица > Вставить строки или столбцы). Birinchi ikkita buyruq bitta qator yoki ustun qo‘yib beradi. Oxirgi buyruq Insert Rows or Columns (Вставка строк или столбцов) muloqot oynasini ekranga chiqaradi (8.27-rasm). Uning yordamida qator yoki ustunni tegishli ko‘rsatkichlar asosida qo‘yish mumkin:

- Yuqorida sanalgan barcha buyruqlarni kontekstli menyu orqali ham bajarish mumkin.

- CTRL+M tugmalar kombinatsiyasi bitta qator qo‘yib beradi.

- CTRL+SHIFT+A bitta ustun qo‘yib beradi.

- Property Inspector (Инспектор свойств) palitrasini oching va jadvalni o‘lchamlarini Rows (Строки) va Cols (Столбцы) maydonlaridan foydalanib belgilang.



8.27-rasm. Jadvalga qator qo‘shish.

Jadvalning qator yoki ustunlarini o‘chirish

Jadvalning qator va ustunlarini o‘chirish quyidagicha amalga oshiriladi:

1. Matn kursorini katakchaga qo‘ying.
2. Navbatdagi buyruqlarni bersangiz, katakcha turgan qator yoki ustun o‘chiriladi:
 - Modify > Table > Delete Row (Изменить > Таблица > Удалить строку) buyrug‘i yoki CTRL+SHIFT+M tugmalar kombinatsiyasi;
 - Modify > Table > Delete Column (Изменить > Таблица > Удалить столбец) buyrug‘i yoki CTRL+SHIFT+ - tugmalar kombinatsiyasi.

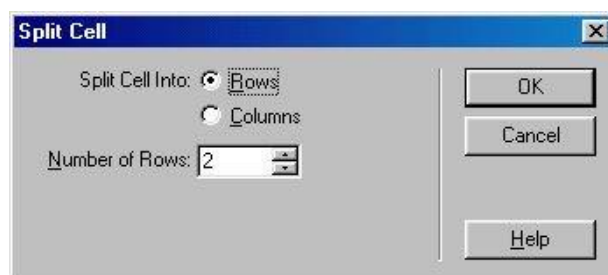
Bundan tashqari biror qator yoki ustun to‘liq belgilangan bo‘lsa, u holda Delete tugmasini bosish bilan ham o‘chirish mumkin.

Katakchalarni birlashtirish va bo‘lish

Dreamweaverda har qanday to‘g‘ri burchakli katakchalarni birlashtirish va ixtiyoriy katakchani qismlarga bo‘lish imkoniyati mavjud.

Katakchani qismlarga bo‘lish quyidagicha amalga oshiriladi:

1. Matn kursorini katakchaga qo‘ying.
2. Bo‘lish buyrug‘ini bir necha usullarda berish mumkin:
 - Modify > Table > Split Cell (Изменить > Таблица > Разделить ячейку) buyrug‘ini berish bilan;
 - Property Inspector (Инспектор свойств) palitrasini ochish va Splits cell into rows or columns (Разделить ячейку на строки или столбцы) tugmachasini bosish bilan;
 - CTRL+ALT+S tugmalar kombinatsiyasini bosish bilan.
3. Buyruq har qanday yo‘l bilan bajarilmasin natijada 8.28-rasmdagi muloqot oynasi paydo bo‘ladi. Bu oynada katakchani nechta qismga bo‘lish kerakligi ko‘rsatiladi.
4. OK tugmasini yoki klaviaturadan ENTER tugmasini bosish.



8.28-rasm. Katakni bo‘lish oynasi.

Katakchalarni birlashtirish quyidagicha amalga oshiriladi:

1. Birlashtiriladigan katakchalarni belgilab oling.
2. Katakchalarni birlashtirish buyrug'ini bering. Buni quyidagi yo'llar bilan bajarish mumkin:

- Modify > Table > Merge Cells (Изменить > Таблица > Объединить ячейки) buyrug'ini berish bilan;
- Property Inspector (Инспектор свойств) palitrasining Merges selected cells using spans (Объединить выбранные) tugmasini bosish bilan;
- CTRL+ALT+M tugmalar kombinatsiyasini bosish bilan;

Jadvalning katakchalarini birlashtirish yoki bo'lish ishlarini Modify > Table (Изменить > Таблица) menyusida joylashgan buyruqlar yordamida ham bajarish mumkin. Ularning imkoniyatlari bilan tanishib chiqamiz:

- Increase Row Span (Увеличить охват строк). Joriy katakchani uning ostida joylashgan katakcha bilan birlashtiradi.
- Decrease Row Span (Уменьшить охват строк). Avval birlashtirilgan katakchani bo'ladi.
- Increase Column Span (Увеличить охват столбцов). Joriy katakchani uning o'ng tarafidagi qo'shni katakcha bilan birlashtiradi.
- Decrease Column Span (Уменьшить охват столбцов). Birlashtirilgan katakchani o'ng tomonidan bitta oddiy katakchani ajratib oladi.

Jadval qismlarini formatlash

Dreamweaver dasturi nafaqat jadvalni formatlash, balki uning tarkibiy qismlari: qatorlarni, ustunlarni va xatto katakchalarni alohida-alohida formatlashga imkon beradi. Jadvalning tarkibiy qismlarini formatlash Property Inspector (Инспектор свойств) palitrası yordamida bajariladi (8.29-rasm). Palitrani boshqaruv elementlari jadvalning katakchalar, qatorlar va ustunlari bilan ishlashga imkon beradi.



8.29-rasm. Jadval elementlarini formatlash.

Ularning asosiy imkoniyatlari bilan tanishib chiqamiz:

- Cell (Ячейка). Bo'limi o'z ichiga ikkita tugmani birlashtiradi: Merges selected cells using spans (Объединить выбранные ячейки) va Splits cell into rows or columns (Разделить ячейку на строки или столбцы). Ular katakchalarni birlashtirish (birinchisi) va bo'lish (ikkinchisi) uchun mo'ljallangan.
- W (Ширина). Katakchani enini belgilaydi.
- N (Высота). Katakchani balandligini belgilash uchun maydon.
- No Wrap (Без переноса). Bug'in ko'chirish rejimini o'chiradi.
- Header (Заголовок). Bu bayroqcha sarlavhani belgilaydi. Ya'ni bu ko'rsatkich faol bo'lganda, katakchadagi ma'lumot o'rtaga tekislanib qalin shrift bilan yoziladi.

- Vg (Фоновый цвет) va Vg (Фоновый рисунок). Palitraning bitta nomdagi ikkita bo‘limi fon rangini va fonga qo‘yiladigan rasmni belgilaydi.
- Brdr (Цвет рамки). Tanlangan katak, qator yoki ustunning ramkasi rangini belgilaydi.
- Horz (Выравнивание по горизонтали). Katakchadagi ma’lumotni gorizontal bo‘yicha tekislab beradi.
- Vert (Выравнивание по вертикали). Katakchadagi ma’lumotni vertikal bo‘yicha tekislab beradi.



8.30-rasm. Vertikal bo‘yicha tekislash.



Savol va topshiriqlar:

1. Jadvallar bilan ishlashda qaysi deskriptorlardan foydalaniladi?
2. Jadval qo‘yish ketma-ketligini aytib bering?
3. Jadvalli ma’lumotlarni import qilish qanday amalga oshiriladi?
4. Jadvalga qator yoki ustun qo‘shish va ularni o‘chirish qanday amalga oshiriladi?
5. Jadvalni yoki uning qismlarini formatlash ketma-ketligini aytib bering?
6. Katakchalarni birlashtirish va bo‘lish imkoniyatlarini aytib bering?

Sahifalarni brauzerlarda ko‘rish

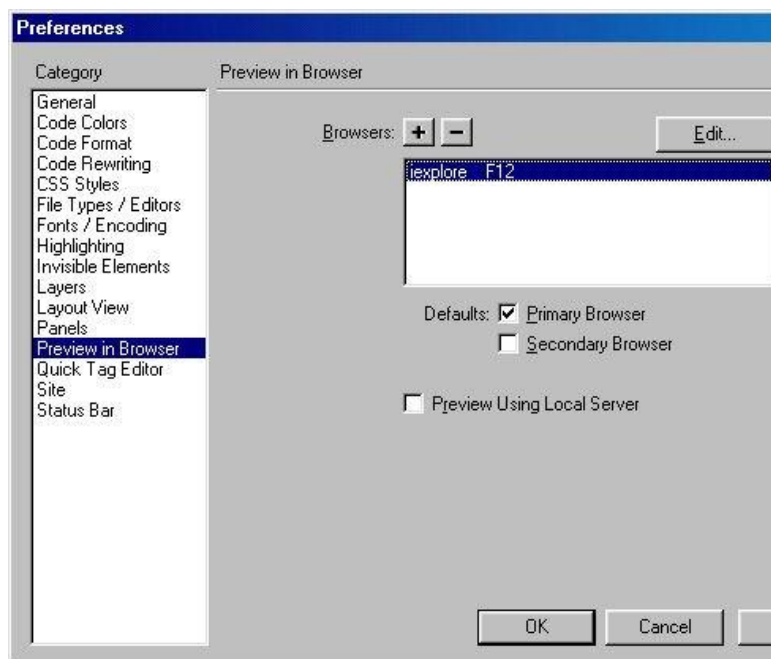
Dreamweaver HTML sahifalarni vizual tahrirlash dasturlari sinfiga kiradi. Dreamweaver dasturi HTML sahifalarni maxsus dasturlar yordamida tabiiy tasvirlarga yaqin ko‘rinishda ko‘rsata oladi. Bunday dasturlar brauzer dasturlari deb yuritiladi. Biz bu dasturlar haqida yuqorida batafsil ma’lumot oldik. Brauzerlarda ko‘rish ketma-ketligi bilan tanishib chiqamiz:

1. Sahifani brauzerda ko‘rish uchun File > Preview in Browser (Файл > Просмотр в браузере) buyrug‘ini bering;
2. Ochilgan oynadan brauzerni tanlang;

Bu buyruq hujjat oynasida ham, uzul dispatcheri oynasida ham mavjud. Asosiy brauzerni yuklash uchun F12 tugmasi xizmat qiladi. Asosiy va yordamchi brauzerlarni tanlash quyidagicha:

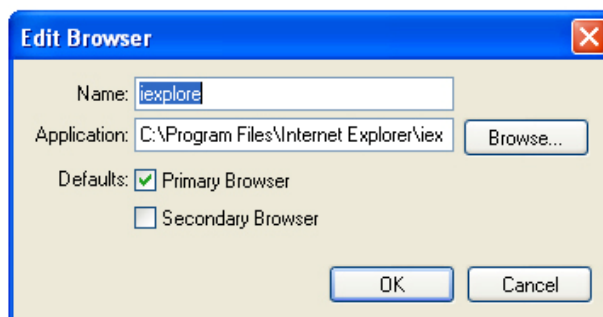
1. File > Preview in Browser > Edit Browsers List (Файл > просмотр в браузере > Изменить список браузеров) yoki Edit > Preferences (Правка > Настройки) buyrug‘ini bering. Qaysi buyruqni berishdan qat’iy nazar, ekranda Preferences (Настройки) muloqot oynasi hosil bo‘ladi;
2. Bu muloqot oynasida barcha tegishli ko‘rsatkichlarni kiriting va ularni o‘rnatilishi uchun OK tugmasini bosing.

Bu ko'rsatkichlarni 8.40- rasmda ko'ramiz:



8.40-rasm. Ko'rish dasturini tanlash.

- + belgisi ostidagi tugma, hisobga olingan ko'rish dasturlari ro'yxatiga yangi brauzerni qo'shadi;
- - belgisi ostidagi tugma tanlangan brauzerni ro'yxatdan o'chiradi;
- Edit (Изменить) tugmasi ro'yxatga olingan brauzer parametrlarini o'zgartirishga imkon beradi;
- Primary Browser (Первичный браузер) bayroqchasi ro'yxatda tanlangan dasturni birinchi darajali brauzerga aylantiradi (birinchi darajali brauzerni chaqirish uchun F12 tugmasini bosish kerak);
- Secondary Browser (Вторичный браузер) bayroqchasi ikkinchi darajali brauzerni tanlaydi. Bu vazifani CTRL+F12 tugmalar kombinatsiyasi yordamida bajariladi;
- Preview Using Local Server (Просмотр нелокальном сервере) bayroqcha lokal server yordamida sahifalarni ko'rish rejimini o'rnatish yoki olib tashlash mumkin. Barcha serverga tegishli dasturiy ta'minot lokal kompyuterga oldindan o'rnatilgan bo'lishi lozim.



8.41-rasm. Edit Browser muloqot oynasi.

Dasturning bunday imkoniyatlari web-sahifani tayyorlash jarayonining oʻzida koʻrish uchun qulaylik yaratib beradi. Bunda foydalanuvchi nafaqat bitta, balki bir nechta brauzerdan bittasini tanlash imkoniyatiga ega boʻladi. Brauzerlar sahifani bir xil koʻrsatsada, ammo ulardan foydalanish, boshqarish va interfeys har xil boʻladi. Shuning uchun ham foydalanuvchi oʻzi xohlagan brauzerni tanlash imkoniyatiga ega boʻladi.



Savol va topshiriqlar:

1. Sahifani brauzer dasturida koʻrish uchun qanday buyruq beriladi?
2. Asosiy va yordamchi brauzerlarni tanlash ketma-ketligini aytib bering?
3. Preferences (Настройки) muloqot oynasi tuzilishini tushuntirib bering?

IV BOB. WEB-SAYTLAR YARATISHNING QO‘SHIMCHA DASTURIY TA‘MINOTI

Flash – Web-sahifalaridagi animatsiyalar, bezaklar va boshqa manbalarni o‘z ichida saqlagan yuqori darajali dizaynlar yaratish uchun mo‘ljallangan dasturiy vosita hisoblanadi. Flash dasturi yuqori darajadagi imkoniyatlari mavjud bo‘lgan texnologik vositalar sirasiga kirib uni bir so‘z bilan Flash texnologiyasi deb atashimiz mumkin. Flash texnologiya yordamida animatsiyalar, Web–saytlar va ular uchun bannerlar, har-xil multimediali prezentatsiyalar yaratish mumkin. Flash texnologiyasi yordamida yaratilgan turli loyiha va mahsulotlari – har doim harakatni yorqin aks ettirib, undagi tovush va video ko‘rinishlar Internet loyihalarida harakatliligi bilan ajralib turadi. Shuning uchun bu texnologiya yordamida saytlar yaratish imkoniyati juda yuqori. Lekin Flash texnologiyasining Internet tizimida ishlashda ayrim kamchiliklari mavjud. Jumladan:

1. Flash elementlarini saytda tasvirlab berish uchun, brauzerda Flash-player o‘rnatish zarur. Internet Explorerda esa dastlabki o‘rnatishdan keyin Flash-player o‘rnatiladi. Agarda siz boshqa brauzerlarni qo‘llaydigan bo‘lsangiz, u holda Flash-playerni o‘rnatishga to‘g‘ri keladi.

2. Flash-saytlari HTML saytlariga qaraganda katta hajmni egallaydi, shuning uchun ularning yuklanishi birmuncha sekin.

3. Flash-texnologiyani ko‘pincha reklama maqsadida qo‘llaniladi, shuning uchun informatsion portallarda qo‘llashga tavsiya qilinmay. Chunki bunday portallar hajmi juda kattalashib ketadi.

4. Flash-saytni yaratish biroz mehnat talab qiladi. Shuning uchun ham bunday saytlarning narhi, HTML saytlar narhiga qaraganda ancha yuqori baholanadi.

Ayrim hollarda juda chiroyli saytlar o‘zining tashrif buyuruvchilarini vaqtning hisobiga yo‘qotadi. Ya‘ni sayt kompyuter ekranida to‘liq tasvirlanib bo‘lgunga qadar ketgan vaqtning evaziga. Ammo bunday saytlar har qanday foydalanuvchini o‘zining yuqori darajali grafikasi, ko‘pqirrali imkoniyatlari, rang-barangligi bilan foydalanuvchilarning asosiy qismini yo‘qotgani yo‘q. Natijada Flash dasturi Web – ishlab chiquvchilar orasida o‘zining keng qamrovli o‘rniga ega bo‘lib bormoqda.

Flash usulida tayyorlangan prezentatsion saytlar yuqori darajadagi grafik almashinuvlar, boy interfaol g‘oyalari va rasmlarning yuklanish tezligi yaxshiligi bilan tashrif buyuruvchilarda katta taassurot qoldiradi. Bundan tashqari harakatlanuvchi panellar, dinamik harakatlar, sinxron tovushli kliplar ham har qanday foydalanuvchini diqqatini o‘ziga tortadi. Siz xattoki informatsion sayt yaratayotgan bo‘lsangiz ham, flash-texnologiyani qisman qo‘llash yo‘li bilan ham, tashrif buyuruvchilaringiz uchun yaxshi taassurot qoldiradigan Internet resurs yaratishingiz mumkin.

Umumiy ma‘lumot va yordamchi jihozlar

Flash dasturi bilan tanishishni boshlaymiz. Bu yerda roliklar ustida ish olib boradigan oynani ko‘rish mumkin, shuningdek uskunalar paneli va bir necha yordamchi panellar mavjud. Bu yordamchi panellar ob‘yektlarning turli xossalarini

tanlashga yordam beradi (masalan, rang, qalinligi va kontur ko‘rinishi va boshqa xususiyatlar). Flash dasturi boshqa dasturlar kabi o‘zining yorlig‘iga ega.



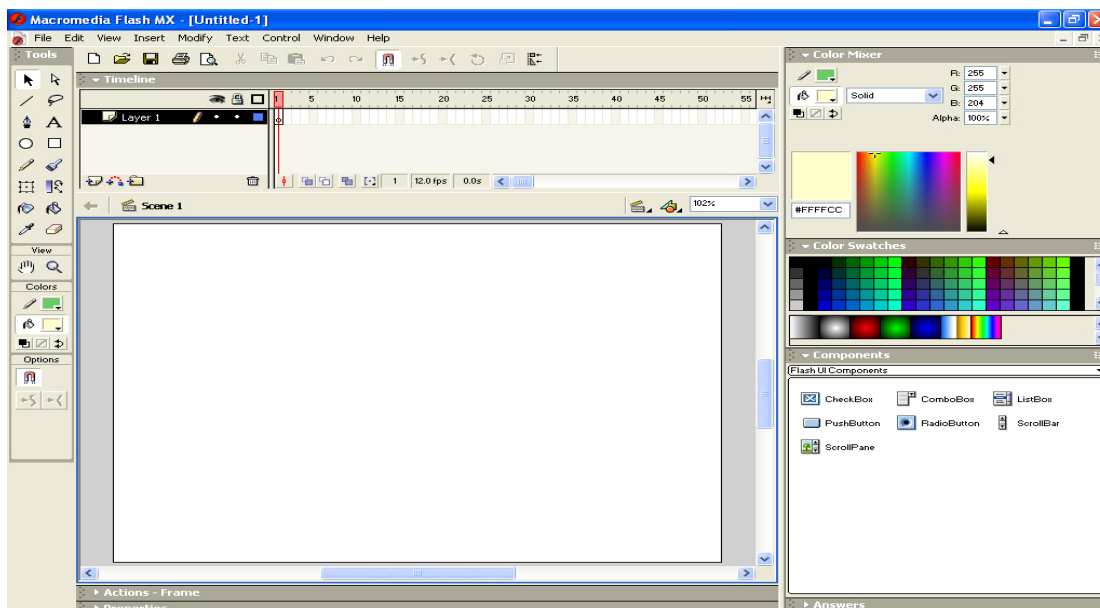
Savol va topshiriqlar:

1. Flash qanday tip dasturlarga kiradi ?
2. Flash dasturidan nima maqsadlarda foydalanish mumkin?
3. Flash texnologiyasini Internet tizimida ishlashining qanday kam-chiliklari mavjud?
4. Flash saytlari oddiy saytlardan nimasi bilan farqlanadi ?



4.1. Flash dasturida oddiy harakatli animatsiyalar va bannerlar yaratish

Flash dasturi oynasi bir nechta qismdan iborat. Ular bilan tanishib chiqamiz. Dastur oynasining asosiy qismlaridan biri dastur ish maydonidir. Ish maydonida biz ob’yektlarni chizish, tahrirlash, harakatlantirish imkoniyatini amalga oshiramiz. Undan yuqorida esa - vaqt diagrammasi yoki vaqt shkalasi (timeline) joylashgan (11.1-rasm).



11.1- rasm. Flash dasturi oynasi ko‘rinishi.

Uskunalar paneli

Uskunalar paneli rasmda ko‘rsatilgandek to‘rt qismdan iborat. Yuqoridagi qismida Tools (uskunalar) ob’yektlarni yasash (chizish) va ularni transformatsiyalaydigan uskunalar joylashgan. Keyiroq ular bilan batafsil tanishib chiqamiz. Keyin tasvirni View (Вид) boshqaruvchi uskunalar keladi: bu qo‘l (Hand tool) rasmi va lupa (Zoom tool).



11.2- rasm. Tools uskunalari paneli.

Bu uskunalari boshqa dasturlarda ham tez-tez uchraydi. Qo‘l uskunasi yordamida ishchi sohani ko‘chirish mumkin. Lupa uskunasi yordamida kattalashtirsa yoki kichiklashtirsa bo‘ladi (kichraytirish uchun ishchi sohada ALT tugmasini bosish kerak). Bu uskunalari faqat rolik tasvirini boshqaradi uning mazmuniga ta‘sir o‘tkazmaydi.

Endi dastur oynasining navbatdagi qismlari bilan tanishib chiqamiz. Ular quyidagilar:

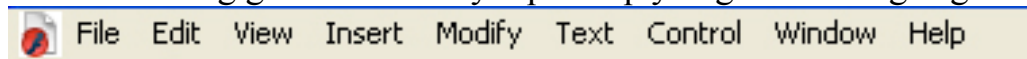
Yorliq qatori: dastur nomi va faol fayl nomini tasvirlab turadi.



11.3- rasm. Yorliq qatori.

Dasturning gorizontali menyu qatori

Flash dasturining gorizontali menyu qatori quyidagi ko‘rinishga ega:



11.4- rasm. Menyu qatori.

Menyu qatori quyidagi menyulardan iborat: **File, Edit, View, Insert, Modify, Text, Control, Window, Help**. Endi menyulardagi asosiy buyruqlar ketma-ketligi, ularning vazifalari va ularning mos funktsional tugmalar bilan tanishib chiqamiz.

11.1-jadval. **File** menyusi buyruqlari

Buyruqlar		tugmalar (Windows)
New	Yangi film	<Ctrl>+<N>
Open	Ochish	<Ctrl>+<O>
Close	Yopish	<Ctrl>+<W>
Save	Saqlash	<Ctrl>+<S>
Import	Import qilish	<Ctrl>+<R>
Export Movie	Eksport qilish	<Ctrl>+<Alt>+<Shift>+<8>
Publish Settings	Nashr o‘lehamlari	<Ctrl>+<Shift>+<F12>
Publish Preview	Nashrni ko‘rish	<F12>
Print	Chop etish	<Ctrl>+<P>
Exit	Dasturdan chiqish	<Ctrl>+<Q>

11.2-jadval. **Edit** menyusi buyruqlari

Buyruqlar		tugmalar (Windows)
Undo	Bekor qilish	<Ctrl>+<Z>
Redo	Takrorlash	<Ctrl>+<Y>
Cut	Qirqib olish	<Ctrl>+<X>
Copy	Nusxa olish	<Ctrl>+<C>
Paste	Qo‘yish	<Ctrl>+<V>
Clear	Tozalash	<Backspace>, <Delete>
Duplicate	Aynan nusxa olish	<Ctrl>+<D>
Select All	Hammasini belgilash	<Ctrl>+<A>
Deselect All	belgilanishni bekor qilish	<Ctrl>+<Shift>+<A>
Cut Frames	Kadrlarni qirqib olish	<Ctrl>+<Alt>+<X>

Copy Frames	Kadrlardan nusxa olish	<Ctrl>+<Alt>+<C>
Paste Frames	Kadrlarni qo'yish	<Ctrl>+<Alt>+<P>
Edit Symbols	Simvollarni tahrirlash	<Ctrl>+<E>

11.3-jadval. **View** menyusi buyruqlari

Buyruqlar		tugmalar (Windows)
Goto	kadrga o'tish	-
Zoom In	masshtabni kattalashtirish	<Ctrl>+<=>
Zoom Out	masshtabni kichiklashtirish	<Ctrl>+<-> .
Magnification	kattalashtirish	-
100%	100%	<Ctrl>+<1>
Show Frame	kadri ko'rsatish	<Ctrl>+<2>
Show All	hammasini ko'rsatish	<Ctrl>+<3>
Outlines	Konturlar	<Ctrl>+<Alt>+<Shift>+<0>
Fast	tez	<Ctrl>+<Alt>+<Shift>+<F>
Timeline	vaqt diagrammasi	<Ctrl>+<Alt>+<T>
Work area	ish maydoni	<Ctrl>+<Shift>+<W>
Rulers	chizg'ich	<Ctrl>+<Alt>+<Shift>+<R>
Hide Edges	qirrani bekitish	<Ctrl>+<H>
Hide Panels	panelni bekitish	<Tab>

11.4-jadval. **Insert** menyusi buyruqlari

Buyruqlar		tugmalar (Windows)
Convert to Symbol	simvolni holatini o'zgartirish	<F8>
New Symbol	yangi simvol	<Ctrl>+<F8>
Frame	kadr	<F5>
Remove Frames	kadrlarni o'chirish	<Shift>+<F5>
Keyframe	bog'langan kadr	<F6> \
Blank Keyframe	bo'sh bog'langan kadr	<F7>
Clear Keyframe	bog'langan kadri o'chirish	<Shift>+<F6>
Create Motion Tween	harakat hosil qilish	

11.5-jadval. **Modify** menyusi buyruqlari

Buyruqlar		tugmalar (Windows)
Layer . . .	qatlam parametrlari	<Ctrl>+<F>
Scane . . .	sahna parametrlari	<Ctrl>+<M>
Document . . .	hujjat parametrlari	<Ctrl>+<Alt>+<Shift>+<C>
Transform	ko'rinishni o'zgartirish	-
Scale	o'lchamni o'zgartirish uchun belgilash	<Ctrl>+<Alt>+<S>
Rotate	Burish	
Frames	kadrlar ustida amallar	
Lock	bekitish (qulflash)	<Ctrl>+<Alt>+<L>
Unlock All	bekitish (qulflash)ni bekor qilish	<Ctrl>+<Alt>+<Shift>+<L>
Group	Guruhlash	<Ctrl>+<G>
Ungroup	guruhni bo'laklash	<Ctrl>+<Shift>+<G>

11.6-jadval. **Text** menyusi buyruqlari

Buyruqlar		tugmalar (Windows)
Font	shriftni o'zgartirish	-
Size	shrift o'lchamini o'zgartirish	<Ctrl>+<Shift>+
Style	shrift stilini o'zgartirish	<Ctrl>+<Shift>+<I>
Align	matnni tekislash	-
Scrollable	markerni o'rnatish	<Ctrl>+<Shift>+<C>

11.7-jadval. **Control** menyusi buyruqlari

Buyruqlar		tugmalar (Windows)
Play	ijro etish	<Enter>
Test Movie	filmni ko'rish	<Ctrl>+<Enter>
Test Scene	sahnani namoyish qilish	<Ctrl>+<Alt>+<Enter>

11.8-jadval. **Window** menyusi buyruqlari

Buyruqlar		tugmalar (Windows)
New Window	yangi oyna	<Ctrl>+<Alt>+<N>
Toolbars	Panellar	-
Tools	ma'lumot	<Ctrl>+<Alt>+<!>
Timeline	tekislash	<Ctrl>+<K>
Properties	belgi	<Ctrl>+<T>
Actions	Protseduralar	<Ctrl>+<Alt>+<A>
Movie Explorer	filmlarni boshqaruvi	<Ctrl>+<Alt>+<M>
Library	Kutubxona	<Ctrl>+<L>

11.9-jadval. **Help** menyusi buyruqlari

buyruqlar		tugmalar (Windows)
What's new	Yangi imkoniyatlar	-
Lessons	Flash bo'yicha darslar	-
Using Flash	Yordam ma'lumotlari	F1
Samples	Flash faylga misollar	-

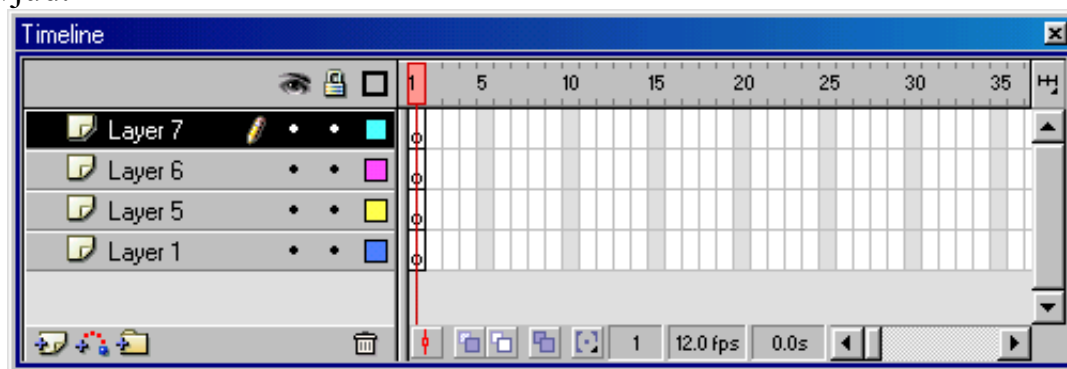
Buyruqlardan foydalanishni tegishli bo'limlarda batafsil ko'rib chiqamiz.

Vaqt diagrammasi

Ish maydoni ustida vaqt diagrammasi (Timeline) paneli joylashgan. Vaqt diagrammasi, filmning aniq sahnalashtirilishiga xizmat qiladi. U yaratilgan qatlamlarni alohida-alohida sahnaga yozib qo'yishni ta'minlaydi. Sahnada namoyish qilinadigan ob'yektlarning o'zgarishi ketma-ketligini ifodalaydi. Vaqt diagrammasi shuningdek animatsiya yaratishda, film interfaol elementlari xatti-harakatini belgilashda asosiy uskuna hisoblanadi. Vaqt diagrammasi murakkab tuzilishga ega bo'lib, ko'p sonli boshqaruv elementlaridan iborat. Vaqt diagrammasi panelida quyidagi umumiy elementlar berilgan (11.5-rasmda):

- filmning mavjud sahnasi uchun qatlamlar ro'yxati (ustma-ust ustun ko'rinishida) ko'rsatilgan. Qatlamlarning nomi va atributlari ko'rsatilgan;
- vaqt diagrammasi har bir qatlam uchun kadrlar shkalasidan iborat;
- vaqt diagrammasidagi kadrlarni formatini o'zgartirishning yig'ma menyusi;

- filmning ayrim parametrlarini ifodalaydigan vaqt diagrammasidagi holat qatori mavjud.



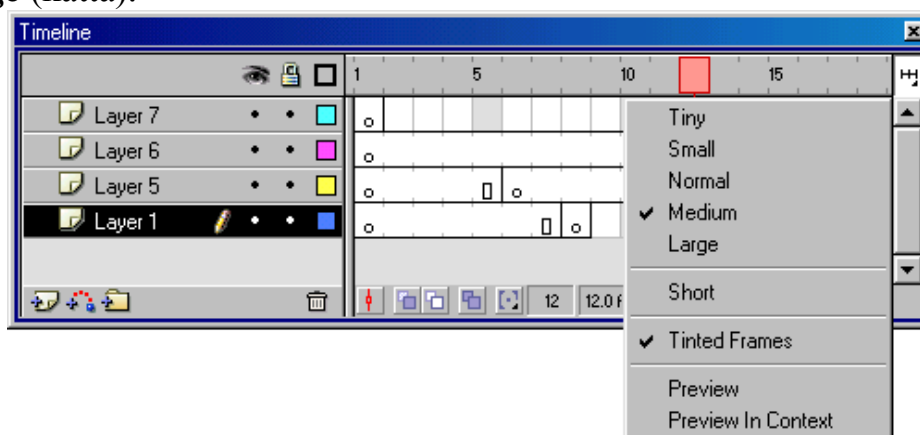
11.5- rasm. Vaqt diagrammasi.

Vaqt diagrammasining kadrlar shkalasi bilan tanishib chiqamiz. Shkala sahnaning barcha qatlamlari uchun umumiy bo‘lib, ammo har bir qatlam o‘zining kadrlar qatoriga ega. Unda kadrlar o‘shish tartibida nomerlangani tasvirlangan. Nomerlanish har 5 qadamda berilgan bo‘lib, kadrlarning har qanday formatida o‘zgartirib bo‘lmaydi.

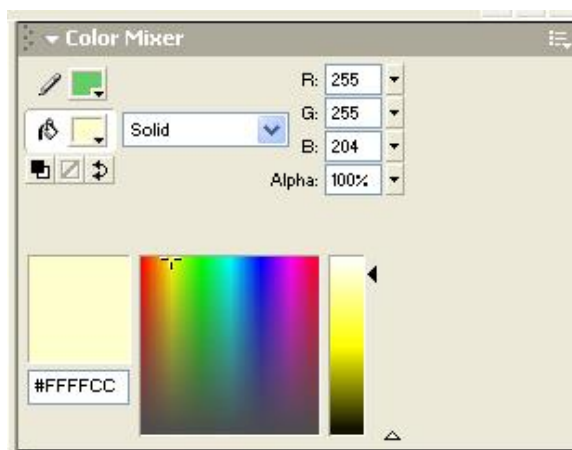
Vaqt diagrammasi ko‘rsatkichi (Play head) animatsiyaning faol kadrini ko‘rsatadi. Qaysi kadr sichqon yordamida tanlansa, o‘sha kadr ustida ko‘rsatkich paydo bo‘ladi. Ko‘rsatkich faol kadrning nomerini ko‘rsatib turadi. Bu esa animatsiyaning tasvirlanayotgan jarayonga sinxronligini ta’minlaydi (11.6-rasm).

Kadrlarni formatlash menyusidagi buyruqlar kadrning katakchasini gorizontal bo‘yicha o‘lchamini belgilaydi:

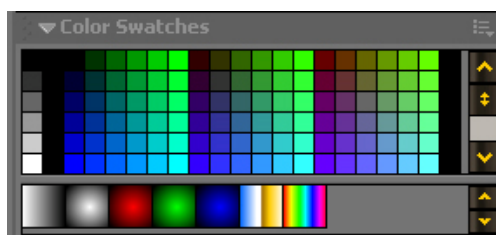
- **Tiny** (mayda);
- **Small** (kichik);
- **Normal** (odatdagi);
- **Medium** (o‘rta);
- **Large** (katta).



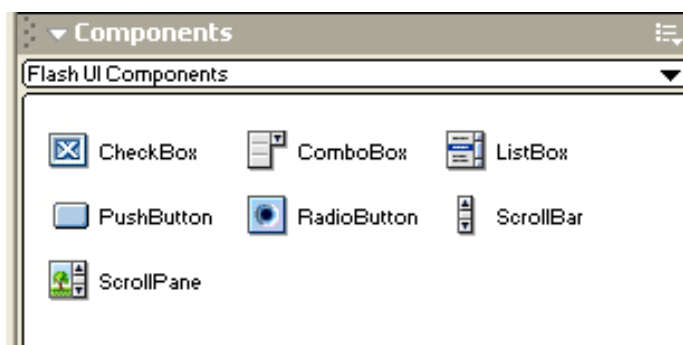
11.6- rasm. Kadrlarni formatlash menyusi.



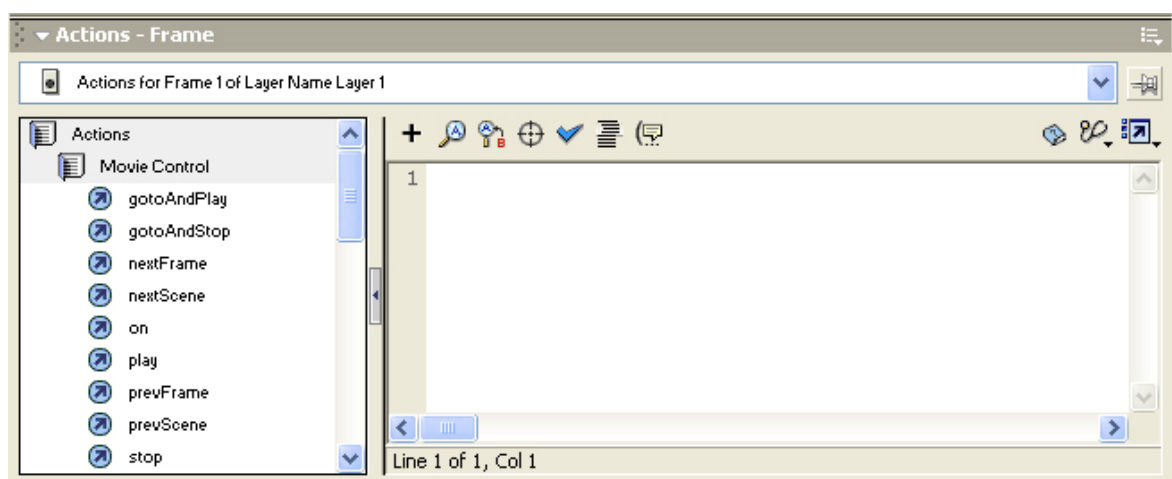
11.7-rasm. Ranglarni saralab quyish oynasi.



11.8-rasm. Quyiladigan rangni tanlash oynasi.



11.9- rasm. Komponentlar.



11.10- rasm. Actions paneli.

Boshqaruv elementlari paneli (Properties)

Yuqorida boshqaruv elementlari panelida berilgan ma'lumotlarga qo'shimcha ravishda quyidagilarni aytish mumkin:

-
- matnli maydonda **URL** ish maydoni matnli qismini belgilaymiz va shu maydonda to‘liq Web-manzilni ko‘rsatamiz (**URL - Universal Resources Locator**). Bog‘langan resurs uchun siz haqiqiy giperhavolaga ega bo‘lasiz;

Boshqa vizual tahrirlagichlardan farqli tomoni, gipermatn namoyish vaqtida ostiga chizilgan shrift bilan emas balki, shtrixli chiziq bilan beriladi. Flash mexanizmida giperhavola hosil qilishning ikkita kamchiligi bor: birinchidan, **URL**ni noavtomatik (qo‘lda) yo‘l bilan kiritiladi va ikkinchidan, Flash **URL**ning xatto lokal fayllar uchun aniqligini tekshirmaydi;

- **Target** (Целевой) ro‘yxati; u resursni qanday brauzer oynasida namoyish qilinishini (yuklanishini) ko‘rsatishda qo‘llaniladi (ko‘rsatilgan URL bo‘yicha chaqirilgan resursni). URL maydoniga hech bo‘lmaganda bitta simvol kiritilgandan keyin ro‘yxat faollashadi, ya‘ni o‘nga murojaat qilish mumkin. Ro‘yxat quyidagi variantlardan iborat (ular HTML tilining **TARGET** atributi qiymatlari bilan o‘xshash):

- **_blank** – chaqirilayotgan hujjat har doim yangi oynada yuklanadi (asl oynaning o‘zida emas);
- **_self** – chaqirilayotgan hujjat xuddi shu oynaning o‘zida yuklanadi (qaysiki bog‘lanish o‘rnatilgan oynaning o‘zida);
- **_parent** - chaqirilayotgan hujjat bosh oynada yuklanadi; agar bu oyna bosh oynaga ega bo‘lmasa, u holda yuklanish xuddi **_self** ko‘rsatkichidagidek bo‘ladi;
- **_top** - chaqirilayotgan hujjat eng yuqori qismdagi oynada yuklanadi (agar ushbu oynaning o‘zi eng yuqori qismda bo‘lsa, u holda **_self** ko‘rsatkichidagidek yuklanadi);

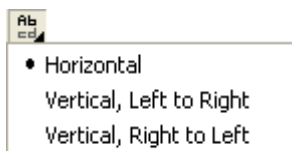
- **Selectable** (Выделяемый) tugmasi; bu tugma xuddi bayroqchadek qo‘llaniladi: tugmani bosish bilan matnga tegishli xususiyatlar panelini o‘rnatish yoki aksincha o‘rnatilgan bo‘lsa uni bekor qiladi;

- **Change Direction of Text** (matn yo‘nalishini o‘zgartirish) foydalanuvchi menyuni ochib matnning yo‘nalishini tanlab olishi mumkin: gorizontol yoki vertikal. Menu o‘z ichiga quyidagi uchta buyruqni oladi (11.11-rasm):

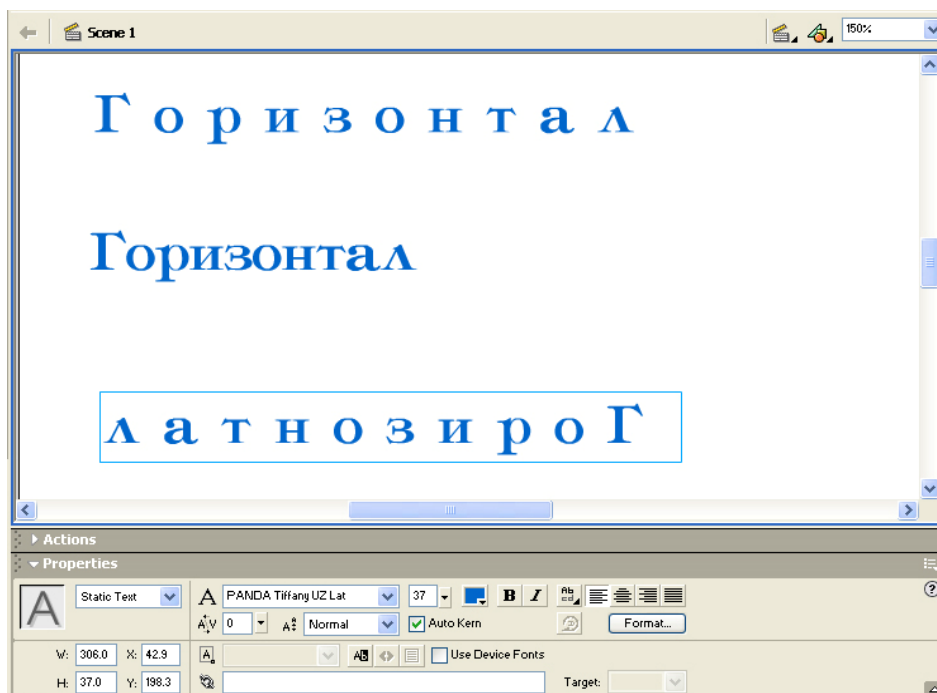
- **Horizontal** (Горизонталь);
- **Vertical, Left to Right** (Вертикаль, chapdan o‘ngga);
- **Vertical, Right to Left** (Вертикаль, o‘ngdan chapga);

Matnning vertikal yo‘nalishini o‘rnatishda formatlashning tugmalar tarkibi o‘zgaradi va undan tashqari **Rotation** faollashadi:

- **Rotation** (burish) tugmasi bayroqcha vazifasini bajaradi va matndagi simvol joylashuv holatini o‘zgartirishni ta‘minlaydi (11.12-rasm);
- **Use Device Font** belgisi – agar belgi o‘rnatilgan bo‘lsa, Flash –pleerda filmni namoyish qilish mobaynida o‘quvchining brauzeri uchun o‘rnatilgan shriftlar qo‘llaniladi. Bu film fayli o‘lchamini kichraytiradi va kichik shriftning qulay o‘qilishini ta‘minlaydi.

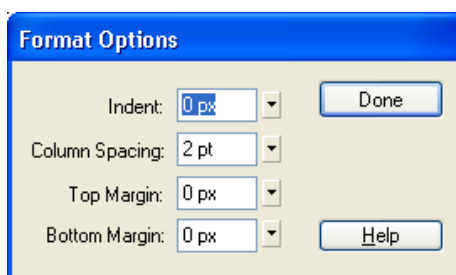


11.11- rasm. Matnning yoʻnalishini oʻzgartirish menyusi.



11.12-rasm. Vertikal matnlar uchun moʻljallangan tugmalar variantlari.

Xususiylar paneli oʻng chegarasi yaqinida **Format** (Формат) tugmasi joylashgan. Uni bosish bilan, qoʻshimcha muloqot oynasi ochiladi va undan foydalanib matn maydonini formatlash parametrlarini oʻrnatish mumkin (11.13-rasm).

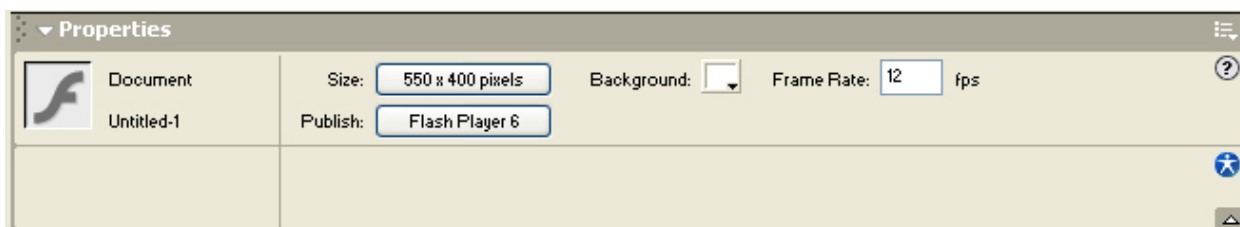


11.13-rasm. **Format Option** muloqot oynasi.

Bu oyna toʻrtta matnli maydondan tarkib topgan:

- **Indent** - yangi qator uchun abzatsning oʻlchami (pikselda);
- **Line Spacing** - yon atrofdagi qatorlar bilan joylashuvi (satrlar orasidagi interval); vertikal matnlar uchun bu parametr matnning ustunlari orasida beriladi;
- **Left Margin** (chap maydon) — matn maydonining chap chegarasi va qatorning birinchi simvoli orasidagi masofa;
- **Right Margin** (oʻng maydon) — matn maydonining oʻng tomon chegarasi va qatorning oxirgi simvoli orasidagi masofa.

Format Option muloqot oynasi matn maydonining har qanday koʻrinishi uchun oʻrinli.



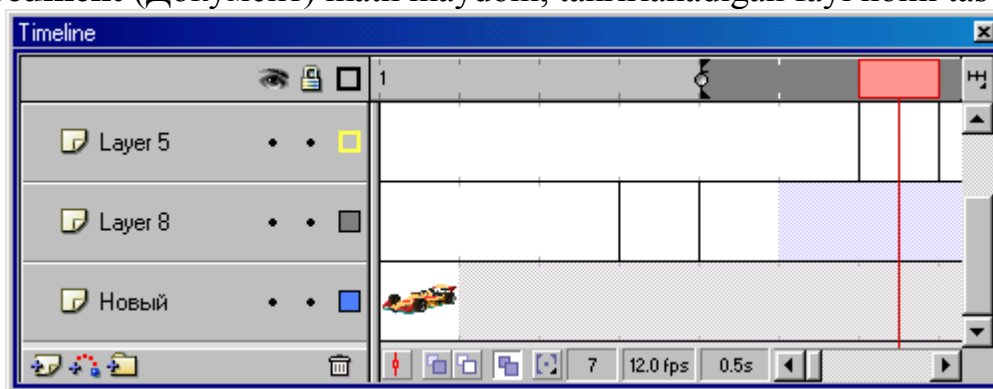
11.14- rasm. Properties paneli.

Ish maydonining quyi qismida Properties paneli joylashgan (11.16-rasm).

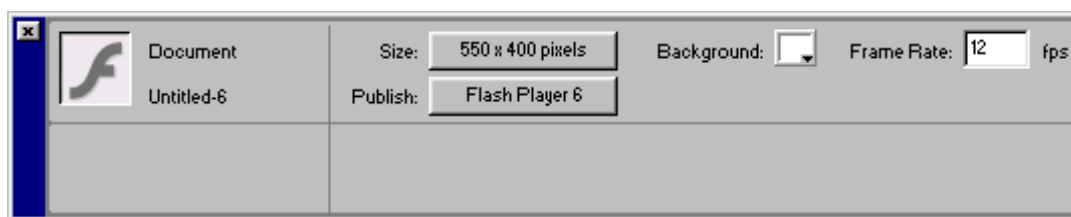
Dasturning ish maydoni va uning ko'rsatkichlarini o'zgartirish

Agar ishchi maydonida birorta ham ob'yekt tanlab olinmagan bo'lsa yoki umuman mavjud bo'lmasa, u holda boshqaruv paneli xossalari filmning umumiy parametrlarini ko'rsatadi. Qandaydir ob'yektni olsangiz uning formati avtomatik ravishda o'zgaradi. 11.16- rasmda butun film uchun xossalar holati paneli formati ko'rsatilgan. Bunda Properties paneli quyidagi elementlardan tashkil topgan bo'ladi:

- **Document** (Документ) matn maydoni, tahrirlanadigan fayl nomi tasvirlanadi;



11.15 - rasm. Preview (vverxy) va Preview In Context (vнизy) buyruqlari qo'llangandan keyingi natija.

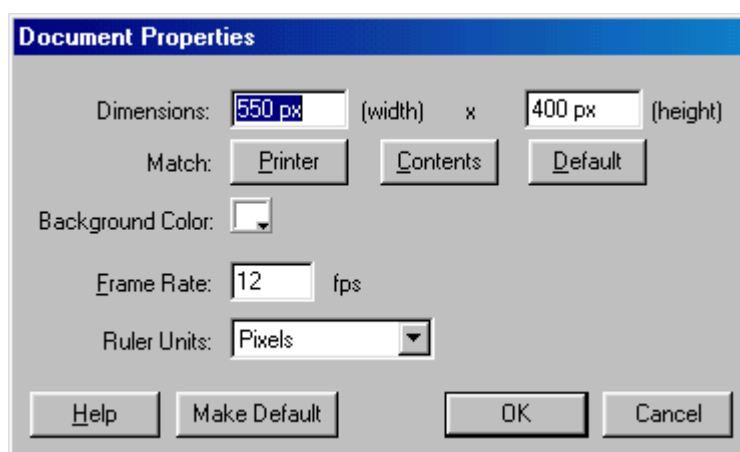


11.16 - rasm. Hujjatning xossalar formati holati.

- **Size** (Размер) tugmasi, ish stoli o'lchamini tasvirlab beruvchi tugma (11.16-rasm);
- **Publish** (Публикация) tugmasi; uni bosish bilan avval o'rnatilgan parametrlar asosida film nashrini boshlaydi;
- **Background** (Фон) tugmasi; uni bosish bilan filmning fon rangini tanlash palitra oynasi hosil bo'ladi. Shu oynadan filmning kerakli foni rangi tanlab olinadi;
- **Frame Rate** (Скорость кадров) matnli maydon, animatsiyaning kadrlar almashish tezligini belgilaydi;
- **?** (so'roq) belgisi tasvirlangan tugma HTML-formatidagi kontekstli ma'lumotnomani chaqirib beradi;

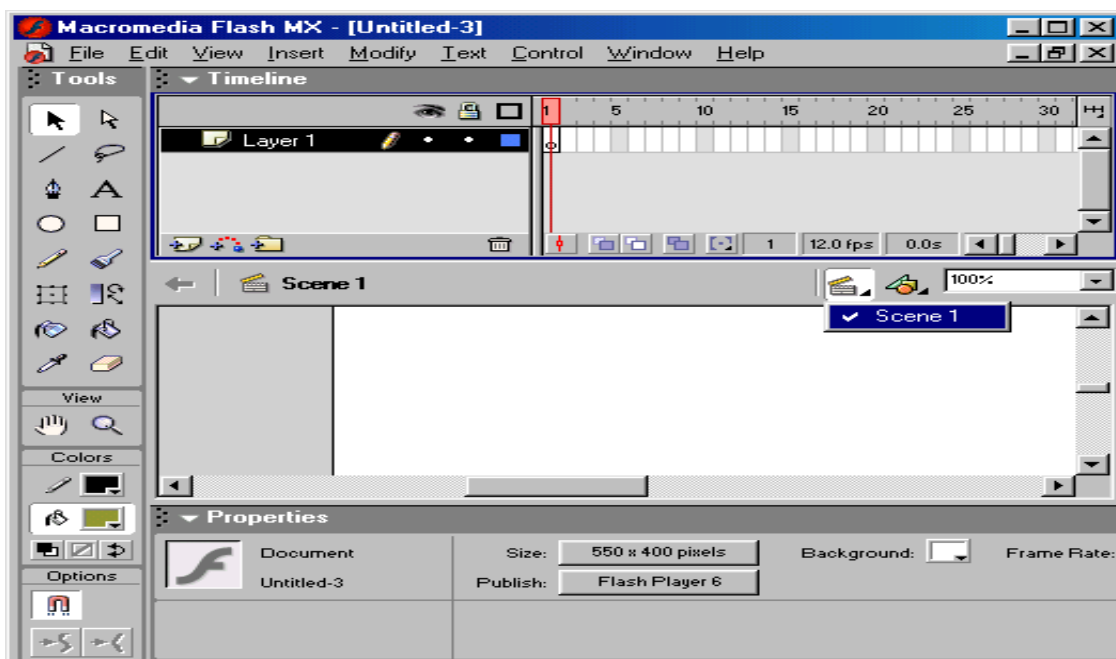
Yulduzchalar tasvirlangan tugma. Bu tugmani bosish bilan **Accessibility** (Доступность) panelini ochiladi. Bu panel filmning barcha elementlari uchun qo'shimcha xususiyatlarni o'rnatish uchun mo'ljallangan. Ish maydoni oynaning butun markaziy qismini egallaydi. Ish maydonida ob'yektlar ustida har qanday tahrirlash ishlarini amalga oshirish mumkin. Faqatgina montaj stolining tegishli kadriga ob'yektlar mos kelsa maqsadga muvofiq bo'ladi. Ish maydoni keng bo'lishidan maqsad:

- sahnadan tashqari ishlarni bajarish;
- ob'yektni sahnaga kirishidan oldin yoki chiqib ketgandan keyingi holatlarni hosil qilish uchun zarur bo'ladi;



11.17-rasm. Document Properties muloqot oynasi.

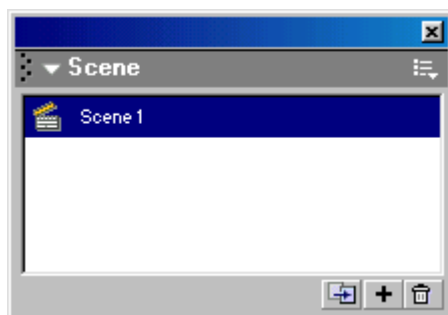
- ish maydoni ostida unga tegishli uskunalar paneli joylashgan. Unda quyidagi interfeys elementlari joylashgan:
 - strelkali tugma butun sahnani tahrirlash rejimiga qaytishni ta'minlaydi; agar siz alohida simvol ustida tahrirlash rejimida ishlayotgan bo'lsangiz, tugma faollashadi;
 - tahrirlanadigan sahna nomli matnli maydon;
 - sahnani tanlash tugmasi; bu tugmani bosish bilan film sahnalaridan tashkil topgan ro'yxatli menyu ochiladi;
 - simvolni tanlash tugmasi; bu tugmani bosish bilan film tugmalaridan tashkil topgan ro'yxatli menyu ochiladi;
 - masshtab, uning yordamida ish maydonini tasvirlash masshtabi tanlanadi.



11.18- rasm. Ish maydonining uskunalari paneli.

Ish maydonining ixtiyoriy joyida sichqon o'ng tugmasi bosilsa, ish maydonining parametrlarini o'zgartirishga mo'ljallangan buyruqlar to'plamini o'z ichiga olgan kontekstli menyu ochiladi. Bu menyu hammasi bo'lib yigirmaga yaqin buyruqlarni o'z ichiga oladi.

Filmning sahnalari bilan ishlash oynasi 11.19-rasmda keltirilgan.



11.19-rasm. Scene paneli.

Uning yordamida siz quyidagi imkoniyatlarga ega bo'lasiz:

- filmning sahnalari ro'yxatini ko'rish; sahnalarning filmda namoyish qilinish ketma-ketligi;
- filmning ro'yxatidagi har qanday sahnasiga o'tish;
- tanlangan sahnadan nusxa olish (asl sahnadan keyin nusxa hosil bo'ladi);
- yangi (bo'sh) sahna qo'shish (tanlangan sahnadan keyin bo'sh sahna qo'shiladi);
- tanlangan sahnani o'chirish.

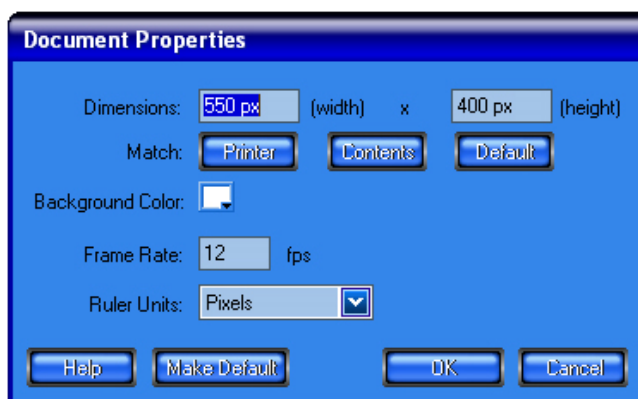
Ish maydoni ko'rsatkichlarini o'zgartirish

Flashda ish maydoni ko'rsatkichlarini o'zgartirish uchun **Modify** menyusidan **Document Properties** buyrug'i beriladi. Xuddi shu vazifani ish maydoni ustida sichqonning o'ng tugmasini bosib kontekstli menyudan **Document Properties**

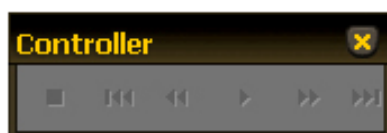
buyrug'i tanlanadi. Natijada rasmdagi **Document Properties** muloqot oynasi hosil bo'ladi.

Bu oynadan foydalanib, ish maydonining bo'yi (height), eni (width) peksillarda ko'rsatiladi. Ish maydoniga quyiladigan rang Background Colorni tanlash orqali o'rnatish, shuningdek kadrlar almashish chastotalarini (Frame Rate) belgilash mumkin.

Flash MXning oynalari bilan batafsil film yaratishni tegishli bo'limlarida tanishib chiqamiz.



11.20-rasm. **Document Properties** oynasi.



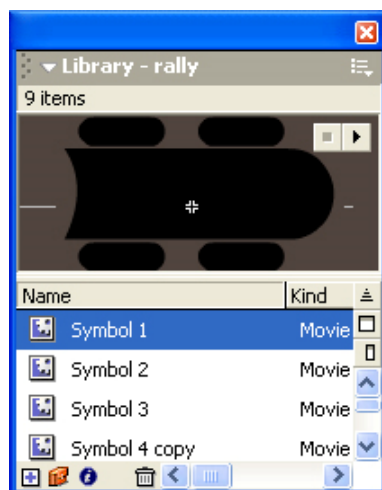
11.21- rasm. Controller paneli.

FLASHda kutubxona bilan ishlash

Flash dasturida kutubxonadan foydalanib quyidagi funktsiyalarni bajarish mumkin:

- import qilinadigan va hosil qilinadigan simvollarini avtomatik ravishda film kutubxonasiga qo'shilishi;
- fayllar tizimining butunligini nazorat qilish va avtomatik kuzatuv;
- simvollarining nomlari bir xillarini yo'qotmaslik;
- to'g'ridan-to'g'ri manipulyatsiya qilish texnologiyasi asosida yangi simvollar yaratish (drag-and-drop).

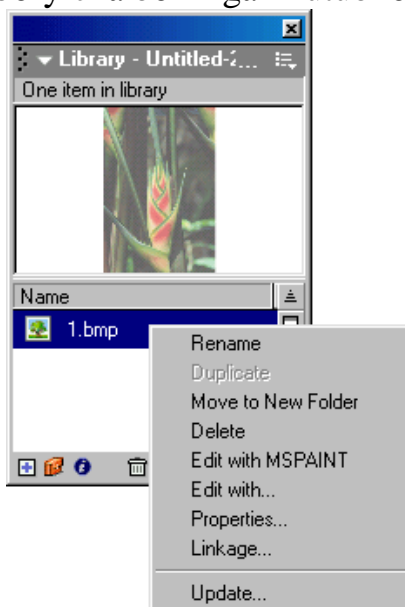
Kutubxona oynasida simvollar bilan ishlashning ko'pgina imkoniyatlari faqat kontekstli menyu orqali amalga oshiriladi. Masalan, simvolni nusxasini hosil qilish uchun, kontekstli menyudan tegishli buyruqni berish kifoya.



11.22- rasm. Flash kutubxonasi.

Kutubxonaning yangi ko‘rinishlari

Flash MXda ikki ko‘rinishga bo‘lingan kutubxonalar mavjud (Shared Library):
Author-time – kutubxonasi, simvollar qayta ishlash vaqti bo‘yicha bo‘lingan;
Rim-time – qo‘yilgan vaqti bo‘yicha bo‘lingan kutubxonona.



11.23- rasm. Filmning kutubxonasi bilan ishlash.



Savol va topshiriqlar:

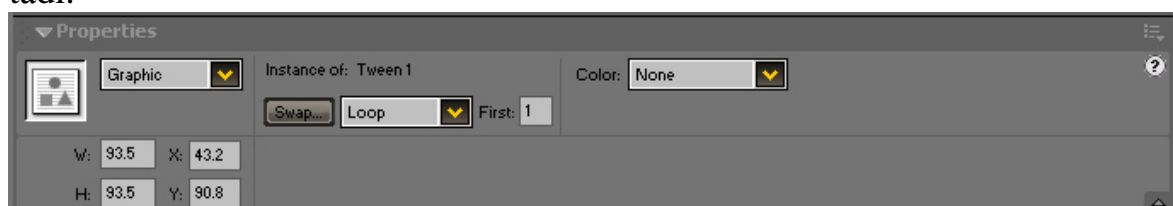
1. Dastur oynasi nechta qismdan iborat?
2. Dastur oynasining vaqt diagrammasi nima vazifani bajaradi ?
3. Uskunalar paneli nima vazifalarni bajaradi?
4. Dasturning kutubxonasi qanday imkoniyatlarga ega?
5. Dasturning ish maydon ko‘rsatkichlari qanday o‘zgartiriladi?
6. Dasturning menyu qatori qanday menyulardan iborat ?

Flashda oddiy harakatlar hosil qilish

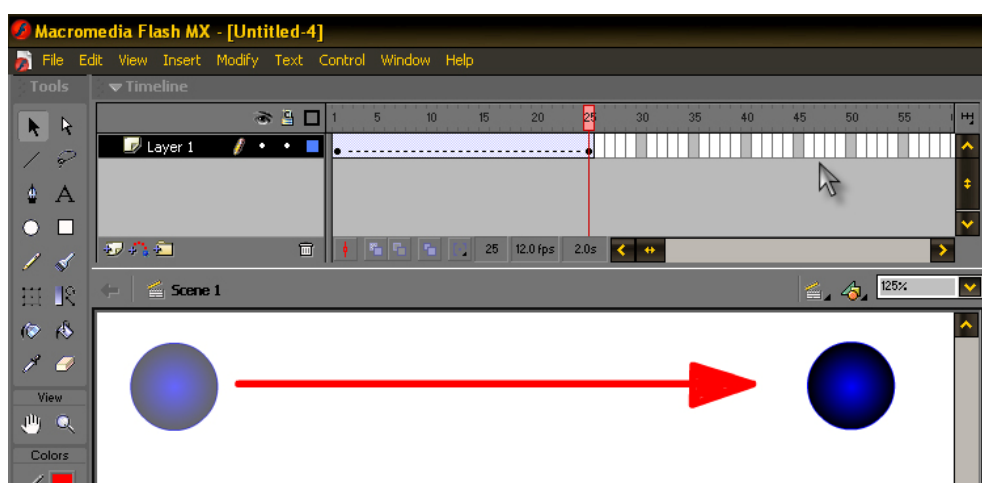
Flashda animatsiyalarni o'rganishni eng sodda harakatlarni ko'rishdan boshlaymiz.

1-misol. Dastlab biror chizilgan ob'yektni bir koordinatadan boshqa koordinataga o'tish jarayoni bilan tanishib chiqamiz:

- o **File > New** buyrug'ini berish bilan yangi ish maydoni hosil qilamiz;
- o ish maydonidagi yagona qatlamning (layer) birinchi kadrini sichqon chap tugmasi bilan bosib belgilaymiz;
- o endi sichqonni **Tools** paneliga olib kelib chizish uchun mo'ljallangan ixtiyoriy uskunani tanlaymiz;
- o sichqon yordamida ixtiyoriy rasm chizamiz;
- o qatlamning kadri ustiga sichqon ko'rsatkichini qo'yib, o'ng tugmasini bosamiz. Hosil bo'lgan menyudan **Create Motion Tween** buyrug'ini beramiz.
- o sichqon ko'rsatkichini bir nechta kadrda keyinga qo'yib **Insert Keyframe** buyrug'i beriladi. Xuddi shu **Keyframe**ni belgilagan holatda rasmni o'zini kerakli koordinataga o'tkazamiz. Buning uchun Properties menyusidan X va Y koordinatalarini o'zgartiramiz (11.38-rasm).
- o harakatni ko'rish uchun **Control > Test Movie** buyrug'i beriladi. Natijada dastlabki nuqtada turgan rasm biroz vaqt ichida boshqa koordinataga o'tadi.



11.38-rasm. Properties paneli.

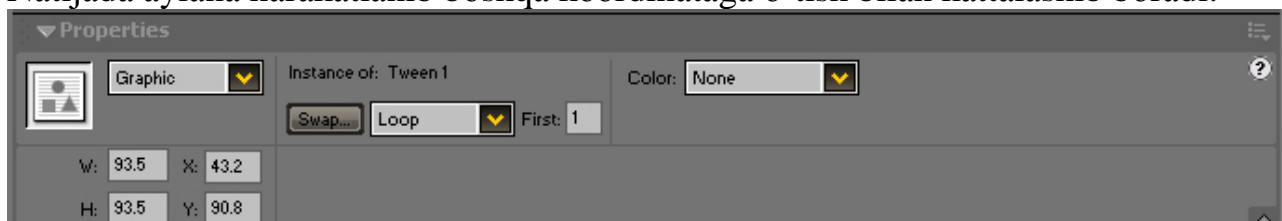


11.39-rasm. Ob'yektni bir koordinatadan boshqasiga o'tkazish.

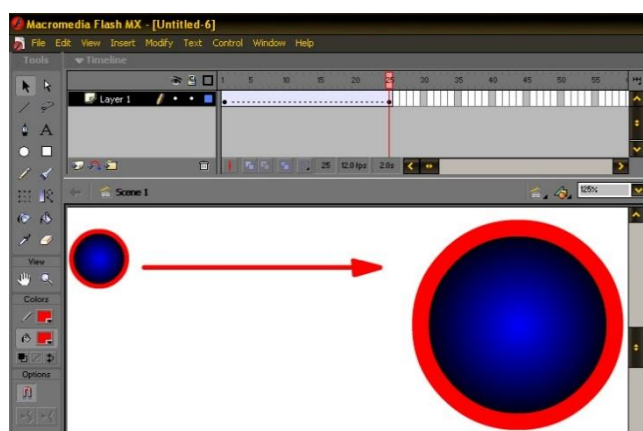
2-misol. 1-misolni biroz o'zgartirsak, shu ob'yektni kattalashganini kuzatishimiz mumkin:

- o yuqoridagi misolning 5 ta qadamini bajarib bo‘lgach, rasmning koordinatasi o‘zgargan kadr ustiga sichqon tugmasini bosish bilan belgilaymiz;
- o natijada rasm belgilanadi, rasmni kattalashtiramiz. Buning uchun **Properties** panelidan W: va H: qiymatlarini kattalashtiramiz;
- o harakatni ko‘rish uchun **Control > Test Movie** buyrug‘i beriladi.

Natijada aylana harakatlanib boshqa koordinataga o‘tish bilan kattalashib boradi.



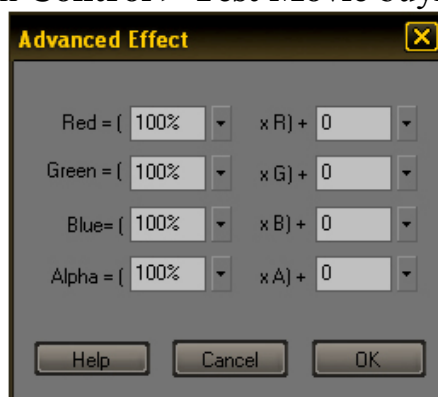
11.40-rasm. Properties paneli.



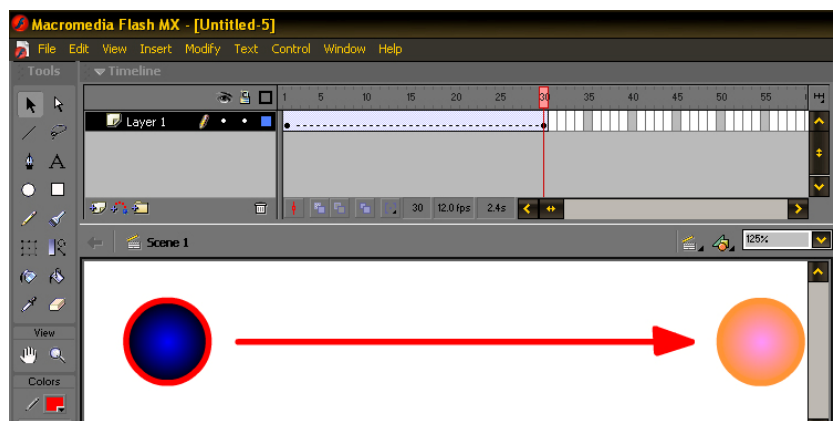
11.41-rasm. Ob‘yektning kattalashishi.

3-misol. Navbatdagi misolda esa xuddi 1-misoldagi chizilgan shar boshqa koordinataga o‘tishi bilan rangi o‘zgarib boradi. Buning uchun quyidagi ketma-ketlik bajariladi:

- o 1-misoldagi birinchi 5 ta bandini bajarib bo‘lgach, rasmning koordinatasi o‘zgargan kadr ustiga sichqon tugmasini bosish bilan belgilaymiz;
- o natijada rasm belgilanadi, keyin Properties panelidan Color ro‘yxatidan Advanced buyrug‘i beriladi;
- o ekranda hosil bo‘lgan **Advanced Effect** oynasidan o‘zimizga kerakli rang ko‘rsatkichlarini tanlab OK tugmasini bosamiz;
- o harakatni ko‘rish uchun **Control > Test Movie** buyrug‘i beriladi.



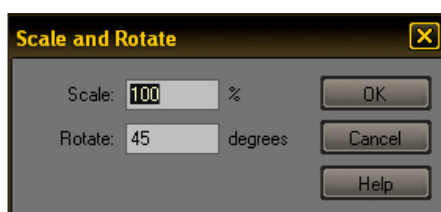
11.42-rasm. Advanced Effect muloqot oynasi.



11.43-rasm. Figurani rangini o'zgarishi.

4-misol. Endigi misolda ob'yektlarni harakatlanish bilan biror burchakka burilishi bilan tanishib chiqamiz. Buning uchun quyidagi ketma-ketlik bilan tanishib chiqamiz:

- 1-misoldagi birinchi 5 ta bandini bajarib bo'lgach, rasmning koordinatasi o'zgargan kadr ustiga sichqon chap tugmasini bosish bilan belgilaymiz;
- natijada rasm belgilanadi, **Modify > Transform > Scale and Rotate** buyrug'i beriladi;
- ekranda hosil bo'lgan **Scale and Rotate** muloqot oynasida tegishli burchak qiymatini ko'rsatib, OK tugmasini bosamiz;
- ko'rsatilgan burchakka rasm buriladi.



11.44-rasm. Scale and Rotate oynasi.



11.45-rasm. Ob'yektlarni biror burchakka burish bilan harakatlantirish.



Savol va topshiriqlar:

1. Oddiy harakat hosil qilish ketma-ketligini tushuntirib bering?
2. Flashda oddiy harakat hosil qiling?
3. Ob'yektning harakati natijasida kattalashishi animatsiyasini hosil qiling?
4. Ob'yekt harakatlanishi natijasida boshqa ranga o'tish animatsiyasini hosil qiling?
5. Ob'yekt harakatlanish yo'lida biror tomonga aylanish animatsiyasini hosil qiling?

4.2. Flash dasturida interfaol animatsiyalar yaratish

Flashda asosiy qatlam - *Guide Layer* tushunchasi mavjud. Asosiy qatlamlar ikkita xususiyatga ega:

- birinchidan, siz unga faqat tahrirlash rejimida ko‘rinadigan izohlar joylashingiz mumkin. Bu izohlar SWF formatiga eksport qilinmaydi; shu ma’noda **Guide** so‘zini «boshqarish», «ko‘rsatma» deb tushunish mumkin;
- ikkinchidan, asosiy qatlam yordamida ob’jektning bir joydan ikkinchi joyga ko‘chish harakati traektoriyasini boshqarish mumkin (tweened-animatsiya harakati yordamida).

Guide qatlamni yaratish uchun quyidagi ketma-ketlikni amalga oshirish kerak:

1. Vaqt diagrammasi panelida qatlamlar ro‘yxatidan ixtiyoriy qatlam ustida sichqon o‘ng tugmasini bosib va kontekstli menyudan **Insert Layer** buyrug‘ini bering.

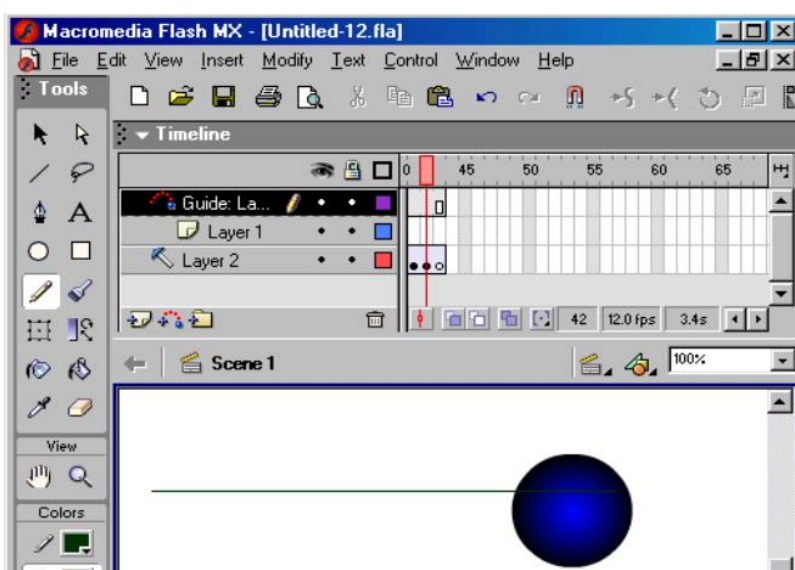
2. Yangi hosil qilingan qatlam ustida sichqon o‘ng tugmasini bosib va kontekstli menyudan **Guide** bandini tanlang. Natijada qatlamlar ro‘yxatida qatlam nomi yaqinida bolg‘acha ko‘rinishidagi belgi hosil bo‘ladi. Bu belgi faqat tahrirlash uchun mo‘ljallanganini bildirib, foydalanuvchilar buni ko‘rishdan mustasno.

Yuqorida aytilganidek bosh qatlamning ikkinchi ko‘rinishi harakatlanayotgan ob’jekt traektoriyasini (bosib o‘tishi kerak bo‘lgan yo‘lni chizmasini) belgilaydi.

Ob’jektни ko‘rsatilgan traektoriyada harakatlantirish uchun quyidagi ketma-ketlikni bajarish kerak:

1. Animatsiyalangan ob’jekt qatlamining nomi ustida sichqon o‘ng tugmasini bosib.

2. Kontekstli menyudan **Add Motion Guide** buyrug‘ini bering. Natijada qatlamlar ro‘yxatida maxsus belgi ostida asosiy qatlam hosil bo‘ladi (11.63-rasm).

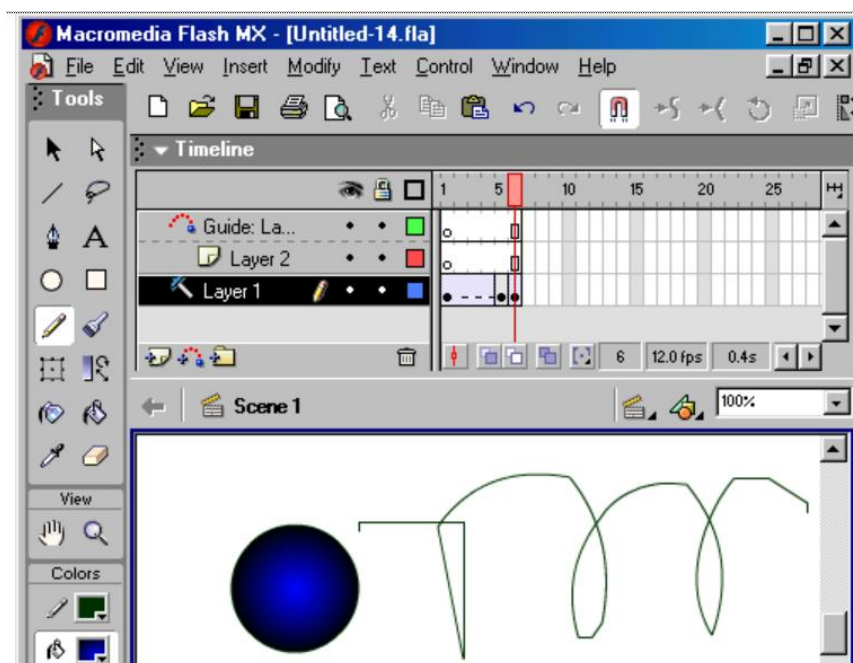


11.63-rasm. Asosiy qatlamni ob’jekt harakatini boshqarishda qo‘llanilishi.

3. Asosiy qatlamni faollashtirish uchun uni nomi ustida sichqoncha chap tugmasi bilan bosamiz.

4. Chizishning ixtiyoriy uskunasi yordamida (Pen, Pencil, Oval, Brush, Rectangle) ob'yekt harakatlanish traektoriyasini tasvirlang.
5. Animatsiyaning birinchi kadri katakchasini sichqoncha chap tugmasi bilan belgilang va Properties panelidan **Orient to Path** (marshrut bo'yicha mo'ljal olish) bayroqchasini o'rnatng.
6. Animatsiyaning oxirgi kadri katakchasini sichqon chap tugmasi yordamida belgilang va kadrning Properties panelidan xuddi shu bayroqchani o'rnatng.
7. Animatsiyalangan ob'yektdan belgilashni olib tashlab, filmni namoyish qilish uchun **Control > Test Movie** buyrug'ini bering.

Eslatma. Agar siz kesishadigan traektoriya hosil qilgan bo'lsangiz, u holda ob'yekt kesishish natijasida hosil qilgan yo'l bo'yicha harakatlanadi (11.64-rasm).



11.64-rasm. Ob'yekt harakatida marshrutning o'zgarishi.



Savol va topshiriqlar:

1. Asosiy qatlam nima va uning qanday xususiyatlari bor?
2. Ob'yektni belgilangan yo'nalishda harakatlantirish uchun qaysi buyruqlardan foydalaniladi?
3. Biror ob'yekt olib uni ixtiyoriy yo'nalish bo'ylab harakatlantiring?

Flashda niqobli animatsiyalar yaratish

Niqob o'z ichiga oddiy qatlamlarni oladi. Faqat bitta qatlam niqobli (bekitadigan) qatlam bo'lib, u faqat o'zining ostidagi qatlamga tegishli ob'yektni bekitadi (niqoblaydi). Qatlamlardan niqob hosil qilish uchun, quyidagi ketma-ketliklarni bajarish kerak:

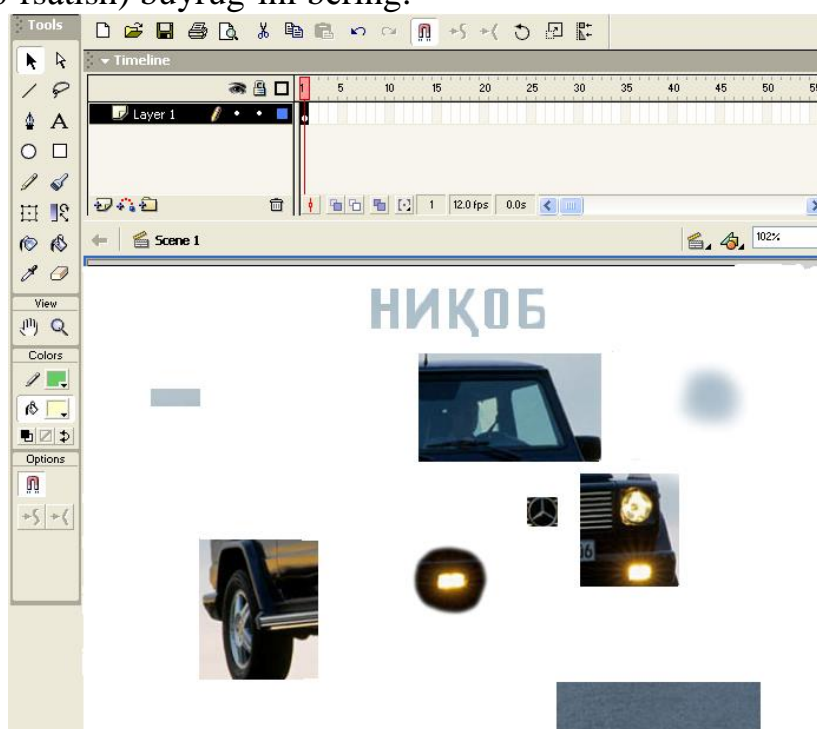
1. Niqob qo‘ymoqchi bo‘lgan qatlamning nomi ustida sichqonning o‘ng tugmasini bosilib, kontekstli menyudan **Insert Layer** buyrug‘ini bering (qo‘yilgan yangi qatlam endi niqob o‘rnida xizmat qiladi).

2. Niqob qatlamiga rang quyung, agar animatsiyali niqob yaratmoqchi bo‘lsangiz, unda faqat bir marta rang quyilgan bo‘lishi kerak.

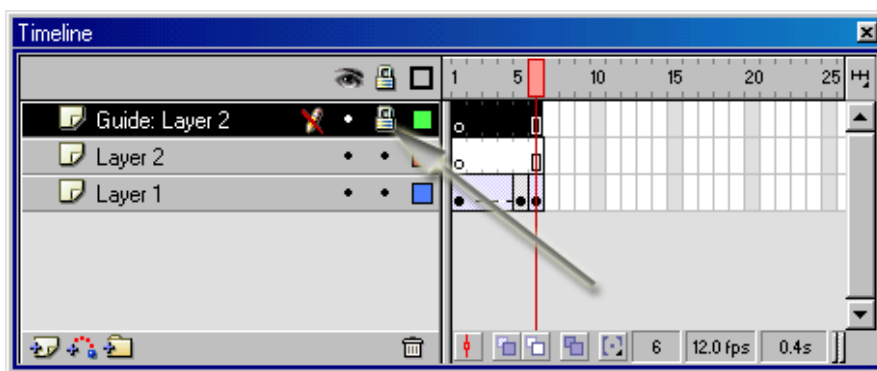
3. Niqob qatlami nomi ustida sichqon o‘ng tugmasini bosung, kontekstli menyudan **Mask** (Маска) buyrug‘ini bering. Shu daqiqadan boshlab yangi qatlam niqobga aylanadi. Shundan keyin qatlamlar to‘plamida quyidagi o‘zgarishlar bo‘ladi (11.67-rasm):

- o qatlamli niqob nomidan chapda niqobga tegishli belgi hosil bo‘ladi;
- o niqoblangan qatlamning nomi o‘ngga suriladi va uning yonida ham shuningdek yangi belgi paydo bo‘ladi;
- o ikkala qatlam ham avtomatik ravishda bekitiladi (qatlamning nomidan o‘ngda hosil bo‘lgan qulf belgisi shundan dalolat beradi).

Niqobga biriktirilgan rangning asli ko‘rinmasdan, balki uning harakati natijasida ortidagi qatlamning ochilishi kuzatiladi. Boshqacha aytganda bekitilgan ob‘yekt niqob orqali harakat bilan ochiladi. Siz niqobning holatini, ko‘rinishini har doim o‘zgartirishingiz mumkin. Uni tahrirlash holatiga keltirish uchun qulf belgisi ustida sichqonni chap tugmasini bosish kifoya. Bu bilan siz qulfni avtomatik ravishda olib tashlagan bo‘lasiz (11.66-rasm). Tahrirlash rejimini ishga tushirish uchun niqobli qatlam nomini sichqon o‘ng tugmasi bilan bosish va kontekstli menyudan **Show All** (Hammasini ko‘rsatish) buyrug‘ini bering.



11.65-rasm. Niqobli rejimi qo‘shilgan ko‘rinishi.



11.66-rasm. Niqobni tahrirlash rejimini yoqish.

Niqobni ko‘rish rejimini yoqish uchun, sichqon o‘ng tugmasini niqobli qatlam nomiga yoki niqob ostidagi qatlam nomiga bosish va kontekstli menyudan **Show Masking** buyrug‘ini berish kerak.

Animatsiyalashgan niqoblar

11.67-rasmda ko‘rsatilgan aylana shakldagi niqobni chap va o‘ng tomon oralig‘ida harakatlanishini ko‘rib chiqamiz.



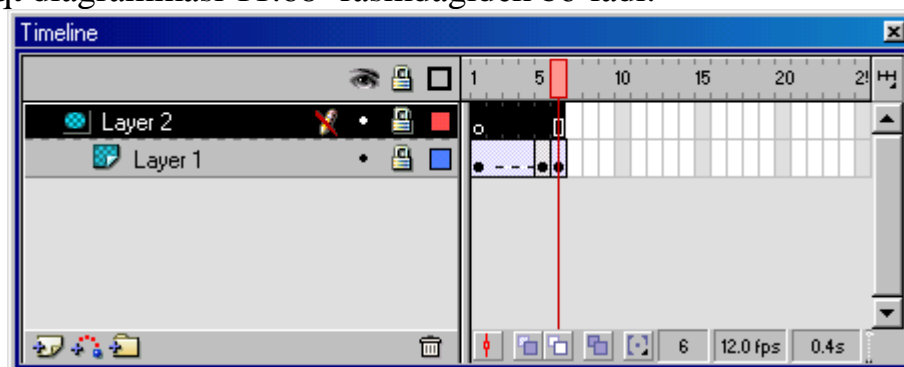
11.67-rasm. Harakatlanadigan niqobga misol.

Buning uchun quyidagi ketma-ketliklarni bajarishimiz lozim:

1. Niqoblamoqchi bo‘lgan qatlamga biror-bir rasmni joylang (masalan, import buyrug‘i yordamida).
2. Niqoblanadigan qatlam nomiga sichqon o‘ng tugmasini bosib va kontekstli menyudan **Insert Layer** buyrug‘ini bering.
3. Yangi hosil bo‘lgan qatlamda o‘zingizga kerakli shaklni chizing.
4. Create Motion Tween buyrug‘ini bering.
5. Sichqon o‘ng tugmasini niqobli qatlam ustida bosib va kontekstli menyudan **Mask** buyrug‘ini oling.

6. Niqoblangan qatlam qatorining oxirgi kadri ustida sichqonning o'ng tugmasini bosing va kontekstli menyudan **Insert Frame** buyrug'ini bering. Birinchi va oxirgi kadrlar orasi oddiy kadrlar bilan to'ladi.

Yuqorida ko'rsatilgan jarayonlar bajarilgandan so'ng animatsiyalangan niqobli filmning vaqt diagrammasi 11.68- rasmdagidek bo'ladi.

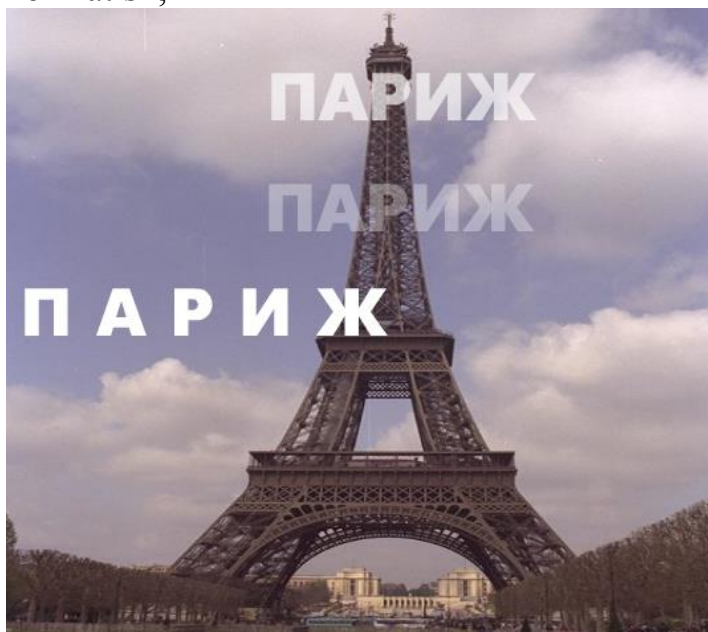


11.68-rasm. Animatsiyalashgan niqobning vaqt diagrammasi ko'rinishi.

Endi niqobni matn maydoni asosida hosil qilingan birmuncha murakkab misol yordamida ko'rib chiqamiz. Bu niqob harakati vaqtida faqatgina surilib qolmasdan balki burilish imkoniyatiga ham ega.

Bunday imkoniyatni faqat ikkita qo'shimcha mexanizmni qo'llash hisobiga ta'minlash mumkin:

- berilgan simvollar oynada aks etish ketma-ketligi;
- tweened-animatsiyaning birinchi kadriga Properties panelidan aylantirish ko'rsatkichlarini o'rnatish;



11.69-rasm. Matn maydoni asosidagi niqobga misol.

Ko'rinishi o'zgaradigan niqobni hosil qilish uchun yuqorida ko'rilgan ketma-ketliklarni bajarish kerak. Farqli tomoni faqat niqobni «jonlantirish» uchun tweened-animatsiya qo'llanilishi kerak. Bunga o'xshash oddiyroq misol 11.69-rasmda ko'rsatilgan. Agar niqobni hosil qilishda bitta fonli qatlam o'rnida ikkita fonli qatlam qo'llansa bundanda qiziqroq natija olish mumkin. Yaxshiroq tasavvur qilish uchun

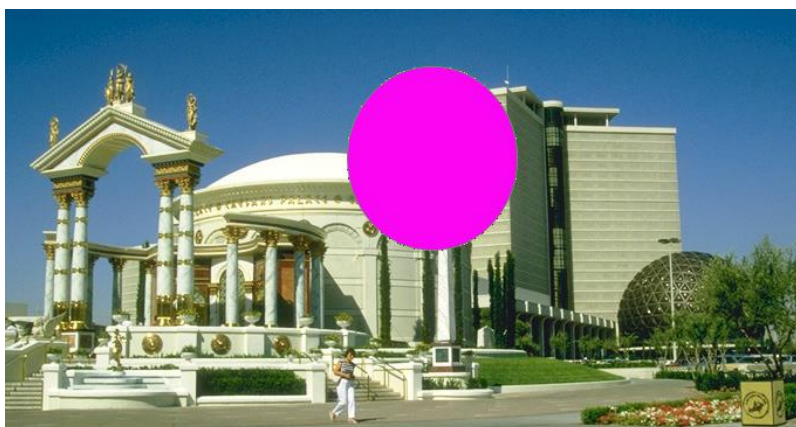
11.70-rasmga qarang. Bu yerda niqob oq maydon bo'yicha emas, balki rasm bo'yicha suriladi.



11.70-rasm. Ko'rinishi o'zgaradigan niqobga misol.



11.71-rasm. Qo'shimcha fonli qatlamga ega niqob.



11.72-rasm. Qo'shimcha fonli qatlamga ega niqob.

- Bunday dekoratsiyani ko'rish uchun, quyidagi ketma-ketliklarni bajarish kerak:
1. Niqob qatlamini va niqoblangan qatlam yaratish;
 2. Yana bitta qatlam hosil qilish uchun, niqob qatlam nomi ustida sichqon o'ng tugmasini bosib va kontekstli menyudan **Insert Layer** buyrug'ini oling.

-
3. Yangi qatlamni ro'yxatdagi niqoblangan qatlam ostiga qo'ying (ya'ni uni eng pastga qo'ying); shundan keyin yangi qatlam avtomatik ravishda niqoblangan holatga o'tadi.
 4. Qo'shilgan qatlam ustida sichqon o'ng tugmasini bosib va qatlam o'lchamlari oynasini ochish uchun kontekstli menyudan **Properties** buyrug'ini bering.
 5. Qo'shilgan qatlamni oddiy holatga qaytarish uchun **Properties** muloqot oynasida **Ture > Normal** holatiga o'tkazing.

Eslatma. Agar siz bitta niqob bilan bog'liq bir nechta niqoblangan qatlamni hosil qilishni xohlasangiz, navbatdagi niqoblangan qatlamni hosil qilish uchun, yaratilgan qatlamlardan biri ustida sichqon o'ng tugmasini bosib va kontekstli menyudan **Insert Layer** buyrug'ini bering. Natijada yangi qatlamga avtomatik ravishda **Masked** parametri o'rnatiladi.



Savol va topshiriqlar:

1. Niqob hosil qilish ketma-ketligini aytib bering?
2. Animatsiyalashgan niqob hosil qiling?
3. Qo'shimcha fonli niqob hosil qilish ketma-ketligini aytib bering?
4. Animatsiyalashgan niqob va qo'shimcha fonli niqoblarning farqlari nimada?



4.3. Flash saytlar yaratish

Tugmalarning eng oddiy vazifasi hammamizga ma'lum. Tugma bosilganda biror vazifani bajarishi lozim. Ana shu vazifalarni tugmalar ustida tegishli dasturlash amalini bajarish yo'li bilan amalga oshiriladi. Quyidagi bir nechta misollar tugmalarni dasturlashga tegishli:

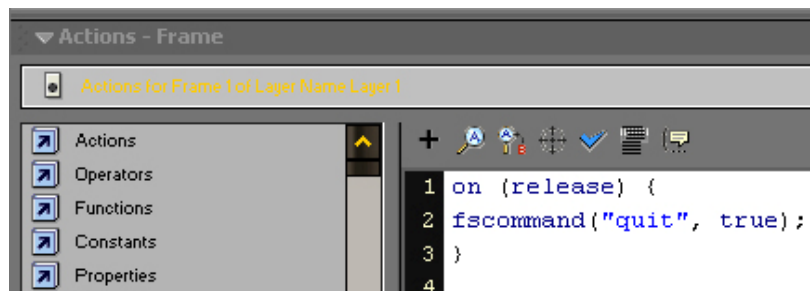
1-misol. Agar bizga biror Flash – film oynasini tegishli tugmani bosgandan keyin yopish kerak bo'lsin.

Bu vazifani bajarish uchun quyidagi ketma-ketlikni bajariladi:

- tovushli tugmalar bo'limida ko'rsatilgan yo'l bilan tugma hosil qilib oling.
 - tugmani belgilab Actions paneliga
- ```
on (release) {
 fscommand("quit", true);
}
```

ifodani yozing (bu ifoda tugma bosilganda Flash-filmni bekitish vazifasini bajaradi);

- Flash filmni ko'rish uchun uchun **Control > Test Movie** buyrug'ini bering.



11.73-rasm. Actions paneli.

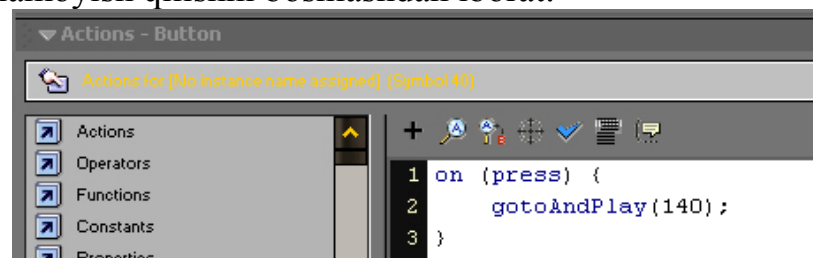


11.74-rasm. Flash – filmni bekitish tugmasi (o‘ng yuqori burchakda).

2-misol. Tugma bosilganda filmning biror boshqa qismiga o‘tish imkoniyati bilan tanishib chiqamiz. Buning uchun quyidagi ketma-ketlikni bajaramiz:

- avval tugma yaratib olamiz;
  - tugmani belgilab, Actions paneliga
- ```
on (press) {
gotoAndPlay(140);
}
```

ifodasi kiritiladi. Ifodaning mazmuni tugma bosilganda 140- kadrda o‘tib, filmning o‘sha qismidan namoyish qilishni boshlashdan iborat.



11.75–rasm. Actions paneli.

- Filmni ko‘rish uchun **Control > Test Movie** buyrug‘ini bering.

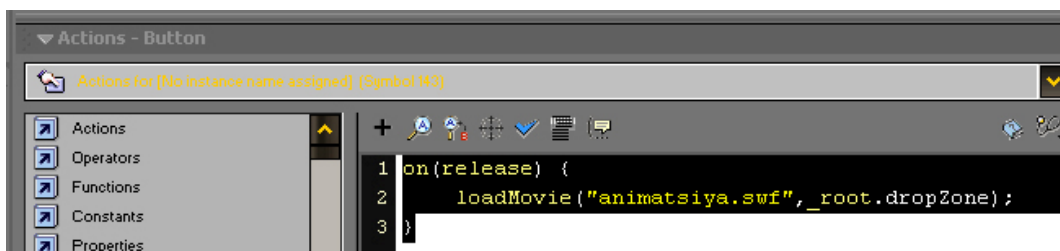
Bu imkoniyatdan faqatgina filmning biror qismida harakatni to‘xtatib qo‘yib, o‘sha qismdan foydalanuvchi xohlagan vaqtda sakrab o‘tish maqsadlarida foydalansa bo‘ladi. Foydalanuvchi o‘zi xohlagan vaqtda tugmani bosish bilan kerakli joyga sakrab filmning ko‘rsatilgan qismini namoyish qiladi.

3-misol. Tugmalarni dasturlash yordamida bir Flash filmdan boshqasiga o‘tish mumkin. Buning uchun quyidagi ketma-ketlikni bajariladi:

- dastlab tugmani yaratib oling;
 - tugmani belgilab, **Actions** paneliga
- ```
on(release) {
loadMovie("animatsiya.swf",_root.dropZone);
}
```

ifodasini kiriting (bu ifoda tugmani bosish bilan animatsiya.swf Flash faylini ochib beradi);

- o filmni ko‘rish uchun **Control > Test Movie** buyrug‘ini bering.



11.76-rasm. Actions panelida faylni yuklash ifodasi.

Dastlabki Flash – filmni namoyishining ixtiyoriy vaqtida tugmani bosganimizda, animatsiya.swf faylidagi film yuklanadi. Tugmalarning bu imkoniyatidan bir filmdan ikkinchisiga o‘tishda yoki orqadagi filmga qaytishda foydalanish mumkin.



### Savol va topshiriqlar:

1. Tugmalarni dasturlash yordamida qanday imkoniyatlarga ega bo‘linadi?
2. Ixtiyoriy animatsiya hosil qiling, unda tugmani bosgandan keyin oyna bekilsin.
3. Ixtiyoriy animatsiya hosil qiling, unda tugmani bosish bilan oxirgi kadrda o‘tsin.
4. Ixtiyoriy ikkita animatsiya hosil qiling, ulardan birinchisidagi tugma bosilganda ikkinchi animatsiyaga o‘tsin.

### Flashda prezentatsiya tayyorlash

Bizga Power-Point dasturida prezentatsiya tayyorlashdan ma’lumki prezentatsiya tayyorlash uchun avvalo taqdim qilinayotgan ma’lumotlarni to‘liq ekranda chiqishi katta ahamiyatga ega. Chunki, foydalanuvchi boshqa dasturlarga chalg‘imasligi va bor e’tiborini faqat shu ma’lumotga qaratmog‘i kerak. Power-Pointda slaydlar tayyorlangandan keyin, Показ слайдов rejimida to‘liq ekran qilib namoyish qilish mumkin. Ammo Flashda esa bu jarayonni amalga oshirish biroz boshqacha kechadi. Buning uchun quyidagi ketma-ketlikni amalga oshiriladi:

- avval namoyish qilinadigan prezentatsiya slaydlarini alohida fayllarda yaratib olinadi;
- ularning namoyishi to‘liq ekranda kechishi uchun har bir faylda alohida qatlam olib, Actions paneliga quyidagi ifodani kiritamiz:  
**fsccommand("fullscreen", true)** (bu yerda fsccommand buyrug‘i berilib, fullscreen to‘liq ekran holati tanlanganda, fayl to‘liq ekranda namoyish qilinadi).
- Slaydlarni (Flash fayllarni) bog‘lash uchun esa, tegishli slaydlarda tugma hosil qilinib belgilangach, Actions paneliga quyidagi ifodani kiritamiz:

```
on(release) {
loadMovie("slayd2.swf", _root.dropZone);
}
```

---

Bu ifoda 2 – slayd fayli slayd2.swf ga o‘tishni ta’minlaydi. Tegishli slayddan avvalgi slaydga qaytish uchun ham xuddi shu yo‘lni qo‘llash mumkin. Prezentatsiyani namoyish qilib bo‘lingach yoki namoyish qilish vaqtida chiqib ketishga to‘g‘ri keladi. Bunday paytda tegishli tugmani bosish bilan chiqib ketiladi. Aynan shu tugmani Flashning o‘zida o‘rnatish quyidagi ketma-ketlik asosida amalga oshiriladi:

- slaydlarning chetgi qismlariga (asosiy ma’lumotlardan tashqariga) chiqib ketish yoki bekitish ma’nolarini bildiruvchi tugma hosil qiling;
- tugmani belgilab, Actions paneliga quyidagi ifodani kiriting:  
on (press) {  
fscommand("quit", true);  
}

Bu yerda **fscommand** buyrug‘ining «**quit**» holati tegishli tugma bosilganda filmni bekitish vazifasini bajaradi.

- filmni namoyish qilish uchun **Control > Test Movie** buyrug‘i beriladi. Prezentatsiya tayyorlashda boshqaruv tugmalarini hosil qilish imkoniyatlari Power-Pointning imkoniyatlariga qaraganda ancha keng va ko‘pqirralidir. Ma’lumotlar namoyishidagi effektlarning boyligi va turli tumanligi har qanday tinglovchini o‘ziga tortadi.



### Savol va topshiriqlar:

1. Flashda tayyorlangan prezentatsiya Power Pointda tayyorlangan prezentatsiyadan nimasi bilan farq qiladi?
2. Flashda prezentatsiya tayyorlashda to‘liq ekran rejimiga o‘tish uchun qanday buyruq berish kerak?
3. Ixtiyoriy prezentatsiya tayyorlang, unda slayddan-slaydga o‘tish imkoniyati, slaydlarni ixtiyoriy vaqtda bekitish va orqaga qaytish imkoniyatlari mavjud bo‘lsin.

### Flashning mantiqiy imkoniyatlari

Flashning imkoniyatlari juda ham ko‘p. Shu jumladan bir qancha mantiqiy imkoniyatlarga ham ega. Faqatgina Flash – filmlar yaratish, Web-sahifalar va ular uchun gif yoki swf animatsiyalar yaratish bilan cheklanmasdan balki alohida ishlaydigan amaliy dasturlar, elektron darsliklar, elektron entsiklopediyalar yaratish va boshqa keng imkoniyatlar mavjud. Masalan, elektron darsliklar yaratishda Flashning yuqori dizaynli grafikasidan, filmni namoyishini boshqaruvi imkoniyatlaridan va Actions Script dasturlash tilidan keng foydalanish mumkin. Navbatdagi ketma-ketliklarda elektron darslik uchun test sinovi faylini yaratish bilan tanishib chiqamiz:

- avval test savolini kiritish uchun **Tools** panelidagi **Text Tool** uskunasiidan foydalaning;
- endi test javoblarini belgilash uchun **Oval Tool** uskunasiidan foydalanib aylanalar chizib oling.

- chizilgan aylanalarni tugmalarga aylantirish uchun **Insert > Convert to Symbol (F8)** buyrug'idan foydalaning;
- har bir tugma to'g'risiga javoblarni kiritib chiqamiz;
- to'g'ri javob tugmasini belgilab, Actions paneliga quyidagi ifodani kiritamiz:

```
on (release) {
 nat1 = 1;
 nextFrame();
}
```

- noto'g'ri javob tugmalariga esa:

```
on (release) {
 nextFrame();
}
```

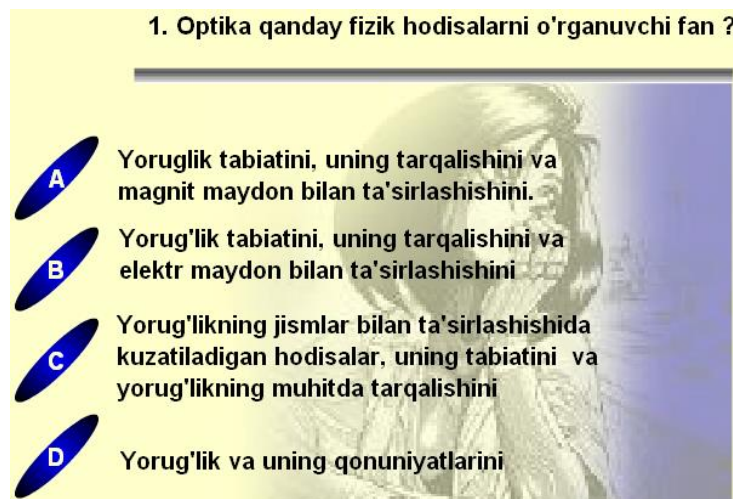
ifodani kiritamiz;

- har bir savol uchun ushbu ketma-ketlikni amalga oshiramiz;
- kadrlar oxirida yana bir kadr qo'yib, unda ikkita dinamik matn hosil qiling. Birini belgilab Properties paneli Var bandiga tnat ifodani, ikkinchisiga esa tball ifodani kiritiladi. Ular bizga test natijasini chiqarib berish uchun xizmat qiladi. O'sha oxirgi kadrning o'zini belgilab **Actions** paneliga:

```
tnat=mat1+ ... +mat10;
tball=(mat1+ ... +mat10)*5;
stop();
```

ifodani kiritamiz. Bu ifodaning birinchi satri to'g'ri bosilgan javoblar sonini qo'shib beradi. Ikkinchi satri esa o'sha to'g'ri javoblar sonini 5ga ko'paytirib beradi, ya'ni har bir savolga 5 balldan baho beradi.

- test tayyor bo'lgandan keyin, natijani ko'rish uchun, **Control > Test Movie** buyrug'i beriladi.



11.77-rasm. Test oynasining ko'rinishi.

Testning ishlash printsipi juda ham oddiy. Agar to'g'ri javobni tanlansa **tnat** ga 1 ni qo'shib keyingi kadrda o'tadi. 2-kadrda noto'g'ri javobni tanlasangiz **tnat** ga hech narsa qo'shmasdan, to'g'ridan-to'g'ri keyingi savolga o'tib ketaveradi. Shu tarzda oxirgi kadrda **tnat** ga tegishli to'g'ri javoblar sonini yig'indisi chiqariladi. **Tball** esa to'g'ri javoblar yig'indisini 5 ga ko'paytirib natija chiqaradi.

---

Bu kabi shablonni bir marta kerakli dizaynda yaratib olib, keyinchalik faqat savollar matnini o'zgartirish bilan ko'p marta qo'llash mumkin.



### Savol va topshiriqlar:

1. Dasturning qanday mantiqiy imkoniyatlari mavjud?
2. Yuqoridagi misollardan foydalanib, shunday test tuzingki, agar to'g'ri javob berilsa keyingi savolga o'tsin, aks holda shu savolda qolaversin.

### Foydalanuvchini qo'llab-quvvatlash ashyolari

Flash paketi tarkibiga kiruvchi foydalanuvchini qo'llash ashyolari, texnologiyani o'rganish uchun ketadigan vaqtni qisqartirishda, qulaylikni oshirishda va foydalanuvchining ishdagi samaradorligini ko'tarishda yordam beradi.

Dastlabki bitta maqsadi yordam tizimini rivojlantirish bo'lib, u bir necha shakllarda berilgan:

- HTML formatidagi interfaol ma'lumotlar tizimi;
- interfaol darslik, HTML-sahifa ko'rinishidagi;
- interfaol darslar, 10-seriyali Flash-filmni o'zida mujassamlagan;
- Web orqali tezkor texnik qo'llash;
- ko'p funktsiyali Flash-filmlarga misollar, ularning har biri yangi filmlar yaratishga ko'mak bo'ladi.

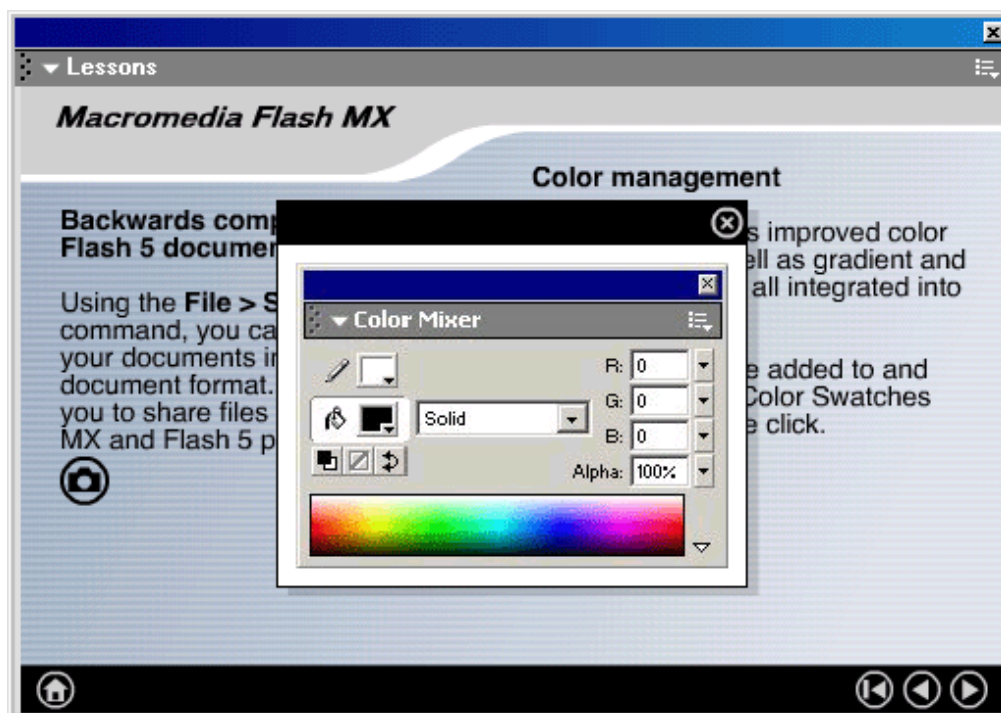
### O'qish va yordam olish ashyolari

Yuqorida ko'rsatilgan barcha kategoriyalar asosiy oynaning **Help** menyusi tarkibiy qismlari hisoblanadi. Unga quyidagilar kiradi:

- **Lessons** - asosiy operatsiyalar bajarish texnikasi haqidagi interfaol darslarni Flash-kengaytmadagi ko'rinishini chaqirish imkoniyatini beradi;
- **Tutorial** - HTML formatidagi qo'llanmani chaqirib berib, Flash-film tayyorlash haqidagi ma'lumotlar ketma-ketliklarni o'z ichiga olgan;
- **Using Flash** - HTML formatidagi ko'p sahifali ma'lumotnomani chaqirish imkoniyatini beradi;
- **ActionScript Dictionary** - ActionScript tilidagi funktsiyalar to'g'risidagi ma'lumotlarni o'zida mujassamlagan HTML formatli ma'lumotnomani chaqirish;
- **Samples** - Flash dasturida tayyorlangan turli ko'rinishdagi, har xil darajadagi misollar papkasini ochib beradi. Bu yerda hamma misollar ikkita formatda: FLA va SWF beriladi; birinchisini - Flash yordamida ochib, uning ustida tahrirlash ishlari olib borish mumkin; ikkinchisi esa - Flash-pleeri yordamida faqat namoyish uchun mo'ljallangan;
- **Flash Support Center** – Flash dasturining mazmunini o'z ichiga olgan ma'lumotlar Web-saytiga bog'lanish.

11.78-rasmda interfaol darslarga mo'ljallangan Flash formatidagi filmning rasmi ko'rsatilgan.





11.78-rasm. Interfaol darslar ko‘rinishi.

Uning yordamida siz quyidagi mavzular bo‘yicha dastlabki ma’lumotlar olishingiz mumkin:

- **Getting Started with Flash** - Flash texnologiyaning umumiy tushunchalari va asosiy imkoniyatlari;
- **Illustrating in Flash** - grafik tasvirlar hosil qilish;
- **Adding and Editing Text** - filmning matnli mundarijasini yaratish va tahrirlash;
- **Creating and Editing Symbols** - yangi simvollar yaratish va tahrirlash;
- **Understanding Layers** - qatlamlarni qo‘llash texnologiyasiga kirish;
- **Creating Tweened Animation** — avtomatik (tweened) animatsiya hosil qilish texnologiyasiga kirish;
- **Creating Buttons** — interfaol tugmalar yaratish;
- Yuqorida sanalganlardan tashqari, darslar tarkibiga yana uchta mavzu kiradi — **For Director Users, For FreeHand & Illustrator Users** va **For Fireworks & Photoshop Users** foydalanuvchilarga yordam berishga mo‘ljallangan mos mavzular haqidagi ma’lumotlar joylashgan.

Shuni alohida ta’kidlash kerakki, Flash dasturida ishlovchilar dasturni o‘rganishni asosan interfaol darslardan o‘rganishni ma’qul ko‘rishadi. Buning sababi, bunday darslar o‘qish uchun qulay, sodda tuzilishga ega va tegishli ko‘rsatmalar bilan boyitilganligi uchundir.

Bizga ma’lumki Flash dasturi 2 o‘lchamli animatsiyalar tayyorlashga mo‘ljallangan. Uning 3 o‘lchamli animatsiyalar yaratish imkoniyati ham 2 o‘lchamli grafikaga asoslanadi. 3 o‘lchamli animatsiyalar uchun maxsus dasturlar mavjud. Ularning ro‘yxati quyidagi jadvalda keltirilgan.

11.11-jadval. 3D tipidagi dasturlar ro‘yxati.

| № | dastur nomi      | Izoh                                                       |
|---|------------------|------------------------------------------------------------|
| 1 | GrowFX 1.5.0 SP1 | 3ds Max uchun loyihalash va animatsiyalar yaratish dasturi |

|    |                                                |                                                                                                                                                                  |
|----|------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2  | Chaos Group V-Ray 2.0 (Autodesk 3ds Max uchun) | Uch o'lchamli modellar tasvirlarini hosil qilishga mo'ljallangan                                                                                                 |
| 3  | Golden Software Voxler                         | Uch o'lchamli ma'lumotlarni grafik ko'rinishda tasvirlash imkonini beradi.                                                                                       |
| 4  | Chaos Group V-Ray Standalone                   | Uch o'lchamli modellar tasvirlarini hosil qilishga mo'ljallangan                                                                                                 |
| 5  | Chaos Group V-Ray (Autodesk Maya uchun)        | Uch o'lchamli modellar tasvirlarini hosil qilishga mo'ljallangan                                                                                                 |
| 6  | DS 3D                                          | Dastur «Master-Shkaf», «Редактор кривых», «Геометрические примитивы» va boshqa uskunalardan iborat bo'lib, qisqa vaqt ichida 3 o'lchamli loyihalarni tayyorlaydi |
| 7  | 3DMasterKit 4.5 Start                          | Dastur tezkorlik bilan 3D animatsiyalar tayyorlashga imkon beradi                                                                                                |
| 8  | StereoTracer 4.5 Professional                  | Dastur oddiy rasmlardan 3D rasmlar tayyorlash uchun ishlatiladi                                                                                                  |
| 9  | StereoTracer 4.5 Home                          | Dastur oddiy rasmlardan 3D rasmlar tayyorlash uchun ishlatiladi                                                                                                  |
| 10 | ArCon Eleco Professional +2009                 | Dastur professional dizayn, loyihalash va uch o'lchamli vizualizatsiyaga mo'ljallangan                                                                           |
| 11 | Autodesk 3ds Max 2011                          | Dastur 3D-loyihalash, animatsiyalar va vizualizatsiyaga mo'ljallangan                                                                                            |
| 12 | 3DMasterKit 4.5 Home                           | Turli effektlar bilan 3D animatsiyalar tayyorlashga mo'ljallangan                                                                                                |
| 13 | 3DMasterKit 4.5 Photo                          | Turli effektlar bilan 3D animatsiyalar tayyorlashga mo'ljallangan                                                                                                |
| 14 | 3DMasterKit 4.5 Professional                   | Turli effektlar bilan 3D animatsiyalar tayyorlashga mo'ljallangan                                                                                                |
| 15 | 3D 2010 dizayn studiyasi                       | 3D muhitida binolarni loyihalash imkonini beradi                                                                                                                 |
| 16 | Piranesi 2010 PRO (Windows uchun)              | 2 o'lchamli rastri muharrir 3 o'lchamli grafikani qayta ishlash uchun mo'ljallangan                                                                              |
| 17 | Piranesi 2010 LITE (Mac OS uchun)              | 2 o'lchamli rastri muharrir 3 o'lchamli grafikani qayta ishlash uchun mo'ljallangan                                                                              |
| 18 | Piranesi 2010 LITE (Windows uchun)             | 2 o'lchamli rastri muharrir 3 o'lchamli grafikani qayta ishlash uchun mo'ljallangan                                                                              |
| 19 | ArCon Eleco Small Business                     | Professional dizaynga mo'ljallangan loyihalash va uch o'lchamli vizualizatsiya                                                                                   |
| 20 | Google SketchUp 7 (Mac OS uchun)               | Oddiy va qulay 3D muharrir                                                                                                                                       |
| 21 | Google SketchUp 7 (Windows uchun)              | Oddiy va qulay 3D muharrir                                                                                                                                       |
| 22 | Autodesk MotionBuilder 2011                    | 3D animatsiyalarni real vaqtda yaratish va filmlar ko'rinishida saqlash                                                                                          |
| 23 | Autodesk Maya 2011                             | Ko'p imkoniyatli professional 3D grafika muharriri                                                                                                               |
| 24 | Insofta Cover Commander 3.0                    | Dizaynerlarga sayt, prezentatsiya va nashriyot tizimlarida PNG, GIF, JPG, BMP formatlarida tasvirlar yaratishga mo'ljallangan                                    |
| 25 | Poser Pro 2010 Windows                         | Odamning professional 3D dizayn va animatsiyasini yaratish                                                                                                       |
| 26 | Poser 8 (Windows uchun)                        | Odamning professional 3D dizayn va animatsiyasini yaratish                                                                                                       |
| 27 | Poser Pro 2010 (Macintosh uchun)               | Odamning professional 3D dizayn va animatsiyasini yaratish                                                                                                       |
| 28 | Quest3D Creative Edition 4.1                   | Real vaqtdagi 3D ilovalarni yaratish                                                                                                                             |
| 29 | Quest3D VR Edition 4.1                         | Real vaqtdagi 3D ilovalarni yaratish                                                                                                                             |
| 30 | Quest3D Power Edition 4.1                      | Real vaqtdagi 3D ilovalarni yaratish                                                                                                                             |
| 31 | Autodesk Entertainment Creation Suite 2011     | O'zida Autodesk Mudbox 2010, Autodesk MotionBuilder 2010, shuningdek Autodesk Maya 2010 yoki Autodesk 3ds Max 2010 dasturlarini birlashtiruvchi paket dastur     |
| 32 | Poser 8 (Mac OS uchun)                         | Odamning professional 3D animatsiyasini yaratish                                                                                                                 |



## 4.4. Java Scriptda dinamik saytlar hosil qilish

### Java Script haqida tushuncha

Gipermatnli ma'lumotlar tizimi ma'lumotlar bo'g'inlari to'plamlari, shu bo'g'inlarda aniqlangan gipermatnli aloqalar to'plamlari va bo'g'inlar va aloqalarni boshqarish instrumentlaridan tashkil topgan. World Wide Web texnologiyasi bu – gipermatnli taqsimlangan sistemalarni Internetga kiritish texnologiyasi va shundan kelib chiqib, u bunday tizimlarning umumiy ta'rifiga mos kelishi kerak. Bu shuni bildiradiki, gipermatnli tizimlarning yuqorida keltirilgan barcha komponentalari Web da ham bo'lishi kerak.

Webga, gipermatnli tizim sifatida, ikki xil nuqtai nazardan qarash mumkin. Birinchidan, o'zaro gipermatnli o'tishlar (ANCHOR konteyneri) vositasida bog'langan tasvirlanishi kerak bo'lgan sahifalar to'plami sifatida. Ikkinchidan, tasvirlanayotgan sahifalar (matn, grafika, kod va hokazolar)ni tashkil qiluvchi elementar ma'lumot ob'yektlarining to'plami sifatida. So'nggi holatda sahifaning gipermatnli o'tishlar to'plami – bu matnga ichki qo'yilgan rasm kabi ma'lumot bo'lagi.

Ikkinchi yondashuvda gipermatnli tizim elementar ma'lumot ob'yektlari to'plami uchun gipermatnli aloqalar rolini o'ynovchi HTML-sahifalarning o'zi tomonidan aniqlanadi. Bu yondashuv tayyor komponentalardan tasvirlanayotgan sahifalarni qurish nuqtai- nazaridan ancha serhosilroqdir.

**Webda** sahifalarni ajratishda “klient-server” arxitekturasi bilan bog'liq muammo yuzaga chiqadi. Sahifalarni ham klient tomonida, ham server tomonida yaratish mumkin. 1995 yilda Netscape kompaniyasi muttaxasislari *JavaScript* dasturlash tilini ishlab chiqib, sahifalarni klient tomonida boshqarish mexanizmini yaratishdi.

Shunday qilib, *Javascript* – bu Webni gipermatnli sahifalarini klient tomonida ko'rish senariyalarini boshqarish tili. Yanada aniqroq aytadigan bo'lsa, *Javascript* – bu nafaqat klient tomonidagi dasturlash tili. Liveware *Javascript* tilining avlodi bo'lib, Netscape serveri tomonida ishlovchi vosita bo'ladi. Ammo *Javascript* tilini mashhur qilgan narsa bu klient tomonida dasturlashdir.

*JavaScript*ning asosiy vazifasi – *HTML-konteynerlar* atributlarining qiymatlarini va ko'rsatuvchi muhitining *hossalirini* HTML-sarlavhalarni ko'rish jarayonida foydalanuvchi tomonidan o'zgartirish imkoniyatlarida, boshqacha aytganda ularni dinamik sarlavhalar qilish (DHTML). Yana shuni aytish joizki, sarlavhalar qayta yuklanmaydi

Amalda buni, masalan, quydagicha ifodalash mumkin, sarlavhaning fonini rangini yoki hujjatdagi rasmni o'zgartirish, yangi oyna ochish yoki ogohlantirish oynasini chiqarish.

“*JavaScript*” nomi Netscape kompaniyasining hususiy maxsuloti. Microsoft tomonidan amalga oshirilgan til rasman *Jscript* deb nomlanadi. *Jscript* versiyalari

---

*JavaScript*ning mos versiyalari bilan mos keladi (aniqroq qilib aytganda oxirigacha emas).

*Javascript* – ECMA (European Computer Manufacturers Association – Evropa Kompyuter Ishlab Chiqaruvchilar Assotsiyatsiyasi) tomonidan standartlashtirilgan. Mos standartlar quyidagicha nomlanadi ECMA-262 va ISO-16262. Ushbu standartlar bilan *Javascript* 1.1ga taqriban ekvivalent ECMAScript tili aniqlanadi. Eslatish joizki, bugungi kunda *Javascript*ning hamma versiyalari ham ECMA standartlariga mos kelavermaydi. Mazkur kurs yoki qo‘llanmada barcha hollarda biz *JavaScript* nomidan foydalanamiz.

*JavaScriptning asosiy xususiyatlari.* *Javascript* – bu Internet uchun katta bo‘lmagan klient va server ilovalarni yaratishga mo‘ljallangan nisbatan oddiy ob‘yektga yo‘naltirilgan til. *JavaScript* tilida tuzilgan dasturlar HTML-hujjatning ichiga joylashtirilib ular bilan birga uzatiladi. Ko‘rish dasturlari (*brauzerlar* – *browser* ingliz so‘zidan) Netscape Navigator va Microsoft Internet Explorer hujjat matniga joylashtirilgan dasturlarni (*Scriptkod*) uzatishadi va bajarishadi.

Shunday qilib, *Javascript* – interpretatorli dasturlash tili hisoblanadi. *JavaScript*da tuzilgan dasturlarga foydalanuvchi tomonidan kiritilayotgan ma‘lumotlarni tekshirayotgan yoki hujjatni ochganda yoki yopganda biror bir amallarni bajaruvchi dasturlar misol bo‘lishi mumkin.

*JavaScript*da yaratilgan dasturlarga misol sifatida foydalanuvchi tomonidan kiritilgan ma‘lumotlarni tekshiruvchi, hujjatni ochish yoki yopish vaqtida qandaydir amallarni bajaruvchi dasturlarni keltirish mumkin. Bunday dasturlar foydalanuvchi tomonidan berilgan ko‘rsatmalarga – sichqoncha tugmachasini bosilishiga, ma‘lumotlarni ekran orqali kiritishiga yoki sichqonchani sahifa bo‘ylab siljutilishiga ko‘ra ish bajaradi. Bundan tashqari *JavaScript*dagidasturlar brauzerning o‘zini va hujjatning atributlarini ham boshqarishi mumkin.

*JavaScript* dasturlash tili sintaktik jihatdan Java dasturlash tiliga, ob‘yektli modellashtirishni istisno qilgan holda, o‘hshab ketsada, lekin ma‘lumotlarni statik tiplari va qat’iy tiplashtirish kabi xususiyatlarga ega bo‘lmaydi. *JavaScript*da Java dasturlash tilidan farq qilib, sinf (klass) tushunchasi bu tilning asosiy sintaktik qurilmasi hisoblanmaydi. Bunday asos sifatida foydalanilayotgan tizim tomonidan qullab-quvvatlanayotgan, oldindan aniqlangan ma‘lumot tiplari: sonli, mantiqiy va satrli; mustaqil ham bo‘lishi, ob‘yektning metodi (*JavaScript*da metod tushunchasi funktsiya/qism-dasturning o‘zi) sifatida ham ishlatilishi mumkin bo‘lgan funktsiyalar; katta sondagi o‘z hossalarga va metodlariga ega bo‘lgan oldindan aniqlangan ob‘yektlardan iborat ob‘yektli model va yana dastur ichida foydalanuvchi tomonidan yangi ob‘yektlarni berish qoidalari hisoblanadi.

*JavaScript*da dasturlar yaratish uchun hech qanday qo‘shimcha vositalar kerak bo‘lmaydi – faqatgina tegishli versiyadagi *JavaScript* qo‘llanishi mumkin bo‘lgan brauzer va DHTML-hujjatlarni yaratishga imkon beruvchi matn muharriri kerak bo‘ladi. *JavaScript*dagidastur bevosita HTML – hujjatlarni ichiga joylashtirilganligi uchun dastur natijasini hujjatni brauzer yordamida ko‘rish orqali tekshirish mumkin va kerakli holda o‘zgartirishlar kiritilishi mumkin.

*JavaScript* dasturlash tilining imkoniyatlari. Uning yordamida HTML – hujjatlarning ko‘rinishi va tuzilishini dinamik ravishda boshqarish mumkin. Ekranda

---

tasvirlanayotgan hujjatga brauzer tomonidan yuklangan hujjatning sintaktik tahlil qilish jarayonida istalgan HTML-kodlarni joylashtirish mumkin. Hujjat ob'yekti yordamida foydalanuvchining oldingi bajargan amallari yoki boshqa bir faktorlarga ko'ra yangi hujjatlarni avtomatik hosil qilish mumkin.

JavaScript yordamida brauzer ishini boshqarish mumkin. Masalan, Window ob'yekti suzib yuruvchi oynalarni ekranga chiqarish, brauzerning yangi oynalarini yaratish, ochish va yopish, oynalarning yugurdagi va o'lchamlarining rejimlarini o'rnatish va hokazolarni imkoniyatini beruvchi metodlarga ega.

JavaScript hujjatdagi ma'lumotlar bilan bog'lanish imkoniyatini beradi. Hujjat ob'yekti va undagi mavjud ob'yektlar dasturlarga HTML-hujjatlarning qismlarini o'qish va ba'zida ular bilan bog'lanish imkoniyatini beradi. Matnning o'zini o'qish mumkin emas, lekin masalan berilgan hujjatdagi gipermatnli o'tishlar ro'yxatini olish mumkin. Hozirgi vaqtda Form ob'yekti va undagi mavjud bo'lishi mumkin bo'lgan ob'yektlar: Button, Checkbox, Hidden, Password, Radio, Reset, Select, Submit, Text va Textarealar hujjatdagi ma'lumotlar bilan bo'g'lanish uchun keng imkoniyatlar beradi.

JavaScript foydalanuvchi bilan aloqa qilishga imkon beradi. Bu tilning eng muhim hususiyati unda amalga oshirilgan hodisalarni qayta ishlashni aniqlash imkoniyati – ma'lum bir hodisaning (odatda foydalanuvchi tomonidan bajarilgan amal) ro'y berish vaqtida bajariladigan dastur kodining ixtiyoriy qismi hisoblanadi. JavaScript hodisalarni qayta ishlovchi sifatida ixtiyoriy yangi oldindan berilgan funktsiyalardan foydalanish imkoniyatini beradi. Masalan, foydalanuvchi sichqoncha ko'rsatkichini gipermatnli o'tishlar ustiga keltirsa, holatlar satrida mahsus habarni chiqaruvchi yoki ma'lum bir amalni bajarishni tasdiqlashni so'rovchi dialogli oynani ekranga chiqaruvchi yoki foydalanuvchi tomonidan kiritilgan qiymatlarni tekshiruvchi va hatolik yuz bergan holda kerakli ko'rsatmalarni berib, to'g'ri qiymatni kiritishni so'rovchi dasturlarni yaratish mumkin.

JavaScript ixtiyoriy matematik hisoblashlarni bajarish imkoniyatini beradi. Bundan tashqari bu tilda vaqt va sanalarning qiymatlari bilan ishlovchi yuqori darajada rivojlangan vositalar mavjud. JavaScript CGI-dasturlarga va Perl dasturlash tiliga va to'ldiruvchi sifatida ayrim hollarda Java tiliga muqobil til sifatida yaratilgan.

Har bir boshlovchi dasturchining asosiy savoli: "Dasturlar qanday tuziladi va bajariladi?". Bu savolga iloji boricha soddaroq, lekin *JavaScript*-kodlarini qo'llanilishining barcha usullarini unutmagan holda javob berishga harakat qilamiz.

Birinchidan, *JavaScript*-kodlari brauzer tomonidan bajariladi. Unda mahsus *JavaScript* interpretatori mavjud. Unga ko'ra dasturning bajarilishi interpretator tomonidan boshqaruvni qachon va qay tarzda olishiga bog'liq bo'ladi. Bu esa, o'z navbatida kodning funktsiyaviy qo'llanilishiga bo'g'liq bo'ladi. Umuman olganda *JavaScript* ning funktsional qo'llanilishining 4 hil usulini ajratib ko'rsatish mumkin:

1. *gipermatnli o'tish (URL sxemasi);*
2. *hodisalarni qayta ishlash (handler);*
3. *o'rniga qo'yish(entity)*
4. *qo'yish (SCRIPT konteyneri).*

*JavaScript* bo'yicha o'quv qo'llanmalarida *JavaScript*ni qo'llashning bayoni odatda SCRIPT konteyneridan boshlanadi. Lekin dasturlash nuqtai nazaridan bu

---

unchalik ham to'g'ri emas, chunki bunday tartib asosiy savol: "*JavaScript*-kodi boshqaruvni qanday oladi?"ga javob bermaydi. Ya'ni *JavaScript*da yozilgan va HTML-hujjatning ichiga joylashtirilgan dastur qanday tarzda chaqiriladi va bajariladi.

HTML-sahifa muallifining kasbi va uning dasturlash asoslaridan habardorligining darajasiga qarab *JavaScript*ni o'zlashtirishga kirishishni bir necha hil variantlari mavjud. Agar siz klassik tillar (C, Fortran, Pascal va h.) bo'yicha dasturlovchi bo'lsangiz, u holda hujjat matni ichida dasturlashdan boshlagan ma'qul. Agar siz Windows sistemasida dasturlashga o'rgangan bo'lsangiz, u holda hodisalarni qayta ishlashni dasturlashdan boshlaganingiz ma'qul. Bordiyu siz faqat HTML bo'yichagina tajribaga ega bo'lsangiz yoki anchadan beri dasturlash bilan shug'ullanmayotgan bo'lsangiz, u holda gipermatnli o'tishlarni dasturlashdan boshlaganingiz ma'qul.

*URL-sxemali JavaScript*. URL sxemasi (Uniform Resource Locator)- bu Web-texnologiyalarning asosiy elementlaridan biri. Webdagi har bir informatsion resurs o'zining o'ziga hos URLiga ega bo'ladi. URL A konteynerining HREF atributida, IMG konteynerining SRC atributida, FORM konteynerining ACTION atributida va hokazolarda ko'rsatiladi. Barcha URLlar resursga ruhsatning protokoliga bog'liq bo'lgan ruhsat sxemalariga bo'linadi, masalan, FTP-arxiviga kirish uchun **ftp** sxemasi, Gopher-arxiviga kirish uchun **gopher** sxemasi, elektron maktublarni jo'natish uchun **smtp** sxemasi qo'llaniladi. Sxemaning tipi URLning birinchi komponentasiga ko'ra aniqlanadi: <http://directory/page.html>.

Bu holatda URL <http> bilan boshlanayapti - mana shu kirish sxemasini aniqlashdir (<http> sxema).

Gipermatnli sistemalar uchun dasturlash tillarining asosiy vazifasi gipermatnli o'tishlarni dasturlashdir. Bu shuni bildiradiki, u yoki bu gipermatnli o'tishlarni tanlashda gipermatnli o'tishni amalga oshiruvchi dastur chaqiriladi. Web-texnologiyalarida standart dastur sifatida sahifani yuklash dasturi hisoblanadi. HTTP protokoli bo'yicha standart o'tishni *JavaScript*da dasturlangan o'tishdan farq qilish uchun yaratuvchilar yangi URL sxemasi – *JavaScript*ni kiritishdi:

```
...
```

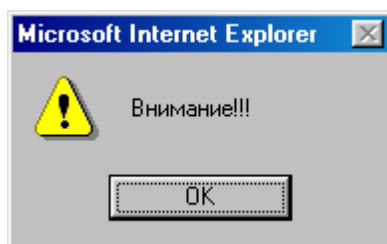
```

```

Bu holatda "JavaScript\_код" matni birinchi holatda gipermatnli o'tishni tanlanganda chaqiriladigan va ikkinchi holatda rasmni yuklashda chaqiriladigan *JavaScript*da yaratilgan dasturiy-qayta ishlovchilarni bildiradi.

Masalan, **Внимание!!!** gipermatnli o'tishga keltirilgan holda ogohlantiruvchi oynani chiqarish mumkin:

```
 Внимание!!!
```



Formadagi **submit** tipidagi tugmachani bosish orqali shu formadagi matnli maydonni to'ldirish mumkin:

```
<FORM NAME=f METHOD=post
ACTION="JavaScript: window.document.f.i.VALUE='Tugmani bosdingiz
Click';void(0);">
<TABLE BORDER=0>
<TR>
<TD><INPUT NAME=i></TD>
<TD><INPUT TYPE=submit VALUE=Click></TD>
<TD><INPUT TYPE=reset VALUE=Reset></TD>
</TABLE>
</FORM>
```

URLda murakkab dasturlar va funktsiya chaqirilishlarini joylashtirish mumkin. Faqatgina shuni yodda tutish kerakki, *JavaScript* sxemasi hamma brauzerlarda ham ishlamaydi, faqatgina Netscape Navigator va Internet Explorerlarning to'rtinchi versiyalaridan boshlab ishlaydi.

Shunday qilib gipermatnli o'tishlarni dasturlashda interpretator boshqaruvni foydalanuvchi sichqoncha tugmasini gipermatnli o'tishga bosgandan keyingina oladi. *Hodisalarni qayta ishlovchilar*. Hodisalarni qayta ishlovchi dasturlar (handler) shu hodisa bog'liq bo'lgan konteynerlarning atributlarida ko'rsatiladi. Masalan, tugmachani bosishda `onClick` hodisasi ro'y beradi:

```
<FORM><INPUT TYPE=button VALUE="Кнопка"
onClick="window.alert('Внимание!!!');"></FORM>
```

*O'rniga qo'yish*. O'rniga qo'yish (entity) Web-sahifalarda anchagina kam uchraydi. Shunga qaramay u brauzer tomonidan HTML-sahifani hosil qilish uchun ishlatiladigan kerakli darajadi kuchli instrument hisoblanadi. O'rniga qo'yishlar HTML-konteynerlarning atributlari qiymatlari sifatida ishlatiladi. Masalan, foydalanuvchining shahsiy sahifasini aniqlovchi forma maydonining boshlang'ich qiymati sifatida joriy sahifaning URLi ko'rsatiladi:

```
<SCRIPT>
function l()
{
 str = window.location.href;
 return(str.length);
}
</SCRIPT>
<FORM><INPUT VALUE="&{ window.location.href};" SIZE="&{l()};">
</FORM>
```

---

```
<SCRIPT>
<!--Bu ...JavaScript-kodga...sharx // -->
</SCRIPT>
<BODY>
Hujjat tanasi ...
</BODY>
</HTML>
```

Bu yerda sahifaning berilgan bo‘lagini eski brauzerlardagi HTML tomonidan interpretatsiya qilinishidan saqlanish uchun qo‘yilgan (yuqori boshqaruvdagilarda hali ham uchraydi). O‘z navbatida HTML-sahifaning ohiri JavaScript interpretatori tomonidan interpretatsiya qilinishidan himoya qilingan (satr boshidagi // belgisi). Bundan tashqari konteyner boshlanishidagi tegning LANGUAGE atributining qiymati sifatida “*JavaScript*” ko‘rsatilgan. *JavaScript*ga muqobil sifatida ko‘riluvchi VBScript keng qo‘llaniluvchi tildan ko‘ra ko‘proq ekzotika hisoblanadi, shuning uchun bu atributni tushirib qoldirish mumkin – “*JavaScript*” qiymati o‘z-o‘zidan qabul qilinadi.

O‘z –o‘zidan ayonki, hujjat sarlavhasida matnni hosil qilishni joylashtirish mantiqsizdir – u brauzer tomonidan namoyish qilinmaydi. Shuning uchun sarlavhada keyinchalik hujjat ichida foydalaniladigan umumiy o‘zgaruvchilar va funktsiyalar e‘lon qilinadi. Bunda Netscape Navigatorda Internet Explorerga nisbatan talab qat‘iyroqdir. Agar funktsiya sarlavhada e‘lon qilinmagan bo‘lsa, hujjat ichida bu funktsiya chaqirilsa, bu funktsiya aniqlanmaganligi to‘g‘risidaga habarni olish mumkin.

Funktsiyani joylashtirish va ishlatishga oid misolni ko‘ramiz:

```
<HTML>
<HEAD>
<SCRIPT>
function time_scroll()
{
 d = new Date();
 window.status = d.getHours()+":"+d.getMinutes()+":"+d.getSeconds();
 setTimeout('time_scroll();',500);
}
</SCRIPT>
</HEAD>
<BODY onLoad=time_scroll()>
<CENTER>
<H1>Matn</H1>
</CENTER>
</BODY>
</HTML>
```

Shuni ta‘kidlab o‘tish kerakki, o‘rniga qo‘yishlar Internet Explorerning 4.0 versiyasida ishlaymaydi, shuning uchun ulardan foydalanishda ehtiyotkorroq bo‘lish kerak. Brauzerga almashtirish mumkin bo‘lgan sahifani berishdan avval shu brauzer tipini tekshirib olish kerak.



---

Oʻrniga qoʻyishlar boʻlgan holda brauzer (parser komponent) tomonidan HTML-hujjatni tahlil qilish vaqtida interpretator boshqaruvni oʻz qoʻliga oladi. Parser konteyner atributida &{..} konstruktsiyasini uchratishi bilan boshqaruvni *JavaScript* interpretatoriga beradi, u esa oʻz navbatida bu kodni bajargandan soʻng boshqaruvni yana parserga qaytaradi. Shunday qilib bu operatsiya HTML-sahifaga grafikani yuklashga oʻhshab ketadi.

*Qoʻyish (SCRIPT konteyneri- interpretatorni majburiy chaqirish).* SCRIPT konteyneri bu oʻrniga qoʻyishlarni *JavaScript*-kod tomonidan hujjat matnini hosil qilish imkoniyati darajasigacha rivojlantirilishidir. Bu maʼnoda SCRIPTni qoʻllanilishi Server Side Includesga oʻhshab ketadi, yaʼni server tomonidan hujjatlarning sahifalarini hosil qilishga. Lekin bu yerda biz sal ilgari ketdik. Hujjatni tahlil qilish vaqtida HTML –parser SCRIPT konteyneri boshlanishi tegini uchratgandan soʻng boshqaruvni interpretatorga beradi. Interpretator SCRIPT konteynerining ichidagi barcha kod boʻlagini bajaradi va SCRIPT konteyneri ohirini koʻrsatuvchi tegdan soʻng sahifa matnini qayta ishlash uchun boshqaruvni HTML-parserga qaytaradi.

SCRIPT konteyneri quyidagi 2 ta asosiy funktsiyani bajaradi:

1. HTML-hujjat ichiga *kodni joylashtirish*;
2. brauzer tomonidan HTML-belgilarni *shartli hosil qilish*.

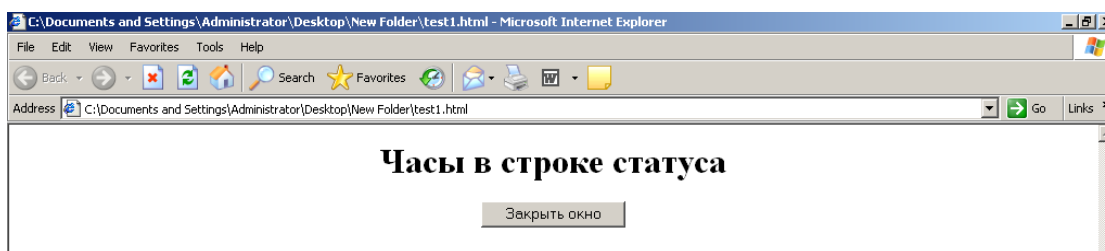
Birinchi funktsiya keyinchalik oʻtishlar dasturi, hodisalarni qayta ishlovchilar va almashtirishlar sifatida ishlatish uchun oʻzgaruvchilar va funktsiyalarni eʼlon qilinishiga oʻhshab ketadi. Ikkinchisi bu hujjatni yuklash yoki qayta yuklash paytida *JavaScript*-kod bajarilishi natijalarini oʻrniga qoʻyishdir.

*HTML-hujjat ichiga kodni joylashtirish.* Hususan olganda bu yerda hech qanday ajralib turadigan har hillik yoʻq. Kodni hujjat sarlavhasida, HEAD konteynerining ichida yoki BODYning ichiga joylashtirish mumkin. Soʻnggi usul va uning hususiyatlari “Brauzer tomonidan HTML-boʻlimlarini shartli hosil qilish” boʻlimida koʻrib chiqiladi. Shuning uchun diqqatimizni hujjat sarlavhasiga qaratamiz.

Sarlavhada kod SCRIPT konteynerining ichiga joylashtiriladi:

```
<HTML>
<HEAD>
<SCRIPT>
function time_scroll()
{
 d = new Date();
 window.status = d.getHours()+":"+d.getMinutes()+":"+d.getSeconds();
 setTimeout('time_scroll();',500);
}
</SCRIPT>
</HEAD>
<BODY onLoad=time_scroll()>
<CENTER>
<H1>Matn</H1>
<FORM>
<INPUT TYPE=button VALUE="Oynani yopish" onClick=window.close()>
```

</FORM>  
</CENTER>  
</BODY>  
</HTML>



Bu misolda biz `time_scroll()` funktsiyani hujjat sarlavhasida e'lon qildik, keyinchalik esa uni `BODY` konteyneri boshining tegida hodisalarni qayta ishlovchi **load** sifatida chaqirdik (`onLoad=time_scroll()`).

O'zgaruvchilarni e'lon qilishga misol sifatida avvalgi-oyna tomonidan keyingi-oynaning statusini o'zgartirishni ko'ramiz.

Quyidagi funktsiya yordamida hosilaviy oynani e'lon qilish va keyin chaqirish orqali yaratamiz:

```
function sel()
{
 id = window.open("", "example", "width=500,height=200,status,menu");
 id.focus();
 id.document.open();
 id.document.write("<HTML><HEAD>");
 id.document.write("<BODY>");
 id.document.write("<CENTER>");
 id.document.write("<H1>Change text into child window.</H1>");
 id.document.write("<FORM NAME=f>");
 id.document.write("<INPUT TYPE=text NAME=t SIZE=20 MAXLENGTH=20
VALUE='Bu test'>");
 id.document.write("<INPUT TYPE=button VALUE='Oynani yopish'
onClick=window.close()></FORM>");
 id.document.write("</CENTER>");
 id.document.write("</BODY></HTML>");
 id.document.close();
}
<INPUT TYPE=button VALUE="Status maydonini misol oynasida o'zgartirish "
onClick="id.defaultStatus='Salom'; id.focus();">
```

Keyingi oynani ochishda **id** o'zgaruvchiga oyna ob'yektiga bo'lgan ko'rsatkich `id=window.open()` ni joylashtirdik. Endi biz uni `Window` sinfining o'bekti identifikatori sifatida ishlatishimiz mumkin. Bizning holatda `id.focus()` dan foydalanish shart. "Misoldagi oynadagi status maydonini o'zgartirish" tugmasini bosish orqali e'tiborni asosiy oynaga o'tkaziladi. U ekran o'lchamida bo'lishi mumkin. Bunda asosiy oyna tomonidan berkitib turilgan ikkinchi oynada o'zgarishlar

---

ro‘y beradi. O‘zgarishlarni ko‘rish uchun diqqatni yana ikkinchi oynaga berish kerak. **id** o‘zgaruvchi qandaydir funktsiya tashqarisida aniqlangan bo‘lishi kerak. Bu holatda u oynaning hususiyatiga aylanadi. Agar biz uni keyingi oynani ochish funktsiyasi ichiga joylashtirsak, u holda hodisalarni qayta ishlovchi **click** orqali unga murojaat qila olmaymiz.

*Brauzer tomonidan HTML-bo‘limlarini shartli hosil qilish.* Serverdan o‘zimizning brauzer imkoniyatlariga yoki foydalanuvchiga moslashtirilgan sahifalarni olish doimo qiziqarli. Bunday sahifalarni hosil qilishning faqatgina 2 imkoniyati bor: server tomonidan yoki bevosita klient tomonidan. *JavaScript* –kod klient tomonida bajariladi (aslida Netscape kompaniyasi serverlari *JavaScript*-kodni server tomonida ham bajarishi mumkin, lekin bu holda u LiveWire-kod nomini oladi; LiveConnect bilan adashtirilmasin), shuning uchun faqatgina klient tomonida hosil qilinishini ko‘rib chiqamiz.

HTML-bo‘limlarini hosil qilish uchun SCRIPT konteyneri hujjat asosiy qismining ichiga joylashtiriladi. Oddiy misol- mahalliy vaqt sahifasiga moslashtirish:

```
<BODY>
```

```
...
```

```
<SCRIPT>
```

```
d = new Date();
```

```
document.write("
");
```

```
document.write("Sahifani yuklanish vaqti:
```

```
"+d.getHours()+" ":"+d.getMinutes()+" ":"+d.getSeconds());
```

```
document.write("
");
```

```
</SCRIPT>
```

```
...
```

```
</BODY>
```



### Savol va topshiriqlar:

1. Java Script nima?
2. Java Script qachon va qayerda ishlab chiqilgan?
3. Java Script imkoniyatlarini aytib bering?
4. Hodisalarni qayta ishlovchilar nima?

## JavaScriptda operatorlar, ifodalar, funktsiyalar

**Operatorlar:** arifmetik amallar, o‘zlashtirish, orttirish, kamaytirish. Shartli ifodalar:

Qo‘shish “+”, ayirish “-”, ko‘paytirish “\*”, bo‘lish “/”, qoldiqli bo‘lish “%”.

Bu ifodalar har qanday sonli ifodalarda uchrashi mumkin.

Shuni ko‘rish mumkinki, JavaScriptdagi o‘zlashtirish operatorlari C va Java dagilar bilan bir hil: “=”, “+=”, “-=”, “\*=”, “/=”, “%=”.

---

$x=y$  ifoda boshqa ko'pgina dasturlash tillaridagi kabi "x" o'zgaruvchiga "y" qiymatni berishni bildiradi.

Quyidagi operatorlar C dagi mos operatorlar bilan bir hil sintaktik qoidalarga ega:

$y+=x$  bilan  $y=y+x$  ekvivalent

$y-=x$  bilan  $y=y-x$  ekvivalent

$y*=x$  bilan  $y=y*x$  ekvivalent

$y/=x$  bilan  $y=y/x$  ekvivalent

$y\%=x$  bilan  $y=y\%x$  –  $y$  ni  $x$  ga butun bo'lgandagi qoldiq- ekvivalent.

Shartli ifodalar quyidagi ko'rinishga ega

$(shart)?qiymat1:qiymat2$

Agar *shartning* qiymati **true** bo'lsa, shartli ifodaning qiymati *qiymat1* ga teng bo'ladi, aks holda *qiymat2* ga teng bo'ladi. Shartli ifodalarni oddiy ifodalarni qo'llash mumkin bo'lgan hamma joyda qo'llash mumkin.

Misol:

$a=(b<1)?0:(x-1)+c$

Orttirish va kamaytirish operatorlari ham C dagi kabi sintaksisga ega: "x++", "++x", "x--", "--x".

**Ifodalar:**

$y=x++$  ikki o'zlashtirishga ekvivalent:  $y=x$ ;  $y=y+1$ ,

$y=++x$  ikki o'zlashtirishga ekvivalent:  $x=x+1$ ;  $y=x$ ,

$y=x--$  ikki o'zlashtirishga ekvivalent:  $y=x$ ;  $x=x-1$ ,

$y=--x$  ikki o'zlashtirishga ekvivalent:  $x=x-1$ ;  $y=x$ .

*Satrlari operatorlar:*

Satrlar bilan ishlash uchun bir nechta operatorlar mavjud:

"+" – satrlarni qo'shish  $s1+s2$ ,  $s1$  satrning simvollar ketma-ketligidan keyin  $s2$  satrning simvollar ketma-ketligi kelgan satrni beradi.

*eval(s)* – JavaScriptda qurilgan ichki funktsiya. U berilgan argument- bir yoki bir nechta JavaScript operatorlarini (nuqtali vergul bilan ajratilgan holda) o'z ichiga olgan  $s$  satr bilan berilgan kodni bajaradi. Berilgan funktsiyani nafaqat operatorni bajarish uchun, balki ifodalarni hisoblash uchun ham ishlatish mumkin. U berilgan koddagi ohirgi hisoblangan ifodaning qiymatini qaytaradi. *eval(s)* funktsiyasi foydalanuvchi tomonidan kiritish punktida kiritilgan qiymatni hisoblash va yana JavaScript dasturda bajarilayotgan kodni dinamik ravishda moslashtirish imkoniyatini beradi. U *parseInt* va *parseFloat* funktsiyalariga nisbatan umumiyroqdir.

*parseFloat(s)* – JavaScriptda ichki qurilgan funktsiyadir. U  $s$  satrdan birinchi belgidan son bo'lmagan dastlabki belgigacha oraliqdan sonni qidiradi (Float tipidagi). Agar son topilmasa NaN ("Not a Number") qiymatini qaytaradi.

*parseInt(s)* – butun sonlar uchun huddi o'sha amalni bajaradi (Integer). Bunda avtomatik ravishda asos topiladi.

*parseInt(s,n)* –  $n$  asos bo'yicha huddi o'sha amal bajariladi ( $n$  2 dan 36 gacha).  $n=0$  bo'lgan hol – bu *parseInt(s)* ni beradi. Bunda avtomatik ravishda asos topiladi.

*Bitli o'zlashtirish operatorlar:*

Bitli o'zlashtirishning bir nechta operatorlari mavjud:

---

$x \ll n$   $x = (x \ll n)$  ga ekvivalent — ikkilik ko‘rinishdagi  $x$  sonda chapga  $n$  bit siljish;

$x \gg n$   $x = (x \gg n)$  ga ekvivalent — ikkilik ko‘rinishdagi  $x$  sonda ishora bitini saqlagan holda o‘ngga  $n$  bit siljish (manfiy sonlarning qo‘shimcha kodida birinchi bit 1 ga teng bo‘ladi. Siljishdan keyin birinchi bitning o‘rniga 1 yoziladi va son manfiyligicha qoladi);

$x \ggg n$   $x = x \ggg n$  ga ekvivalent — ikkilik ko‘rinishdagi  $x$  sonda birinchi bitiga 0 ni qo‘ygan holda o‘ngga  $n$  bit siljish (siljishdan keyin birinchi bitning o‘rniga 0 yoziladi va son musbat songa aylanadi);

Chap tomondagi barcha ifodalar (bizda  $u$  — " $x$ ") 32-bitli butun sonlarga aylantirib olinadi, keyin siljish bajariladi, keyin hosil bo‘lgan son ifoda- natijaning (bizda  $u$  yana " $x$ ") tipiga aylantiriladi va o‘zlashtirish bajariladi.

Misollar:

1)  $9 \ll 2$  ifoda 36 ni beradi, chunki 1001 (9 sonining 2 lik ko‘rinishdagi tasviri) da chapga 2 bit siljish 100100 ni, ya’ni 36 ni beradi. Yetishmayotgan bitlar 0 lar bilan to‘ldiriladi.

$9 \gg 2$  ifoda 2 ni beradi, chunki 1001 (9 sonining 2 lik ko‘rinishdagi tasviri) da o‘ngga 2 bit siljish 10 ni, ya’ni 2 ni beradi. Yetishmayotgan bitlar 0 lar bilan to‘ldiriladi.

"&" — bitli AND — "VA";

"|" — bitli OR — "YOKI";

"^" — bitli XOR — "INKOR QILUVCHI YOKI".

Barcha operatsiyalar sonlarning 2 lik ko‘rinishida bajariladi, lekin natija oddiy o‘nli ko‘rinishda qaytariladi.

Misollar:

$15 \& 9$  ifoda 9 ni beradi, ya’ni (1111) AND (1001) ifoda 1001 ga teng;

$15 | 9$  ifoda 15 ni beradi, ya’ni (1111) OR (1001) ifoda 1111 ga teng;

$15 \wedge 9$  ifoda 6 ni beradi, ya’ni (1111) XOR (1001) ifoda 0110 ga teng.

**Mantiqiy ifodalar.**

"&&" — mantiqiy AND — "VA";

"||" — mantiqiy OR — "YOKI";

"!" — mantiqiy NOT — "YO‘Q".

Misol:

$(a > b) \&\& (b \leq 10) || (a > 10)$

JavaScriptda mantiqiy ifodalarni qisqartirilgan tekshirilishi deb ataluvchi amal doimo qo‘llaniladi:  $B1 \&\& B2$  operandda  $B1 = \text{false}$  bo‘lgan holda  $B2$  ni baholash bajarilmaydi va **false** qiymati qaytariladi. Shunga o‘hshab  $B1 || B2$   $B1 = \text{true}$  holatda **true** deb baholanadi. Bunda mantiqiy ifodalarni tahlili chapdan o‘ngga qarab olib boriladi va faqatgina to‘la ifoda baholanib bo‘lishi bilan natija qaytariladi. Shuning uchun agar  $B2$  sifatida funktsiya bo‘lsa, u chaqirilmaydi va agar u aks ta’sirga ega bo‘lsa, bu hatolikka olib kelishi mumkin.

**Taqqoslash operatorlari:**

"=" - "teng";

">" - "katta";

"<" - "kichik";

---

">=" -"katta yoki teng";  
"<=" -"kichik yoki teng";  
"!=" -"teng emas".

Taqqoslash operatorlari nafaqat sonli ifodalarga, balki satrli ifodalarga ham qo'llanilishi mumkin. Bunda satrlar teng hisoblanadi qachonki ulardagi barcha simvollar ustma-ust tushsa va bir hil tartibda kelsa (bo'sh belgi ham simvol deb qaraladi). Agar satrlar turli hil uzunliklarga ega bo'lsa, u holda uzunroq satr katta hisoblanadi. Agar ularning uzunliklari teng bo'lsa, u holda chapdan o'ngga borgan sari kattaroq nomerdagi simvolga ega bo'lgan satr katta hisoblanadi.

( $a < b < c < \dots < z < A < \dots < Z$ ).

Satrlarni qo'shish mumkin, agar  $S1="bu "$ ,  $S2="mening satrim"$  bo'lsa, u holda  $S1+S2$  "bu mening satrim"ni beradi.

Operatorlarning bajarilish tartibi (kichigidan boshlab; bir satrdagilarning tartibi bir hil):

"+=", "-=", "\*=", "/=", "%=", "<<=", ">>=", ">>>=", "&=", "^=", "|=".

**Operatorlarning tartibi.** Quyida operatorlarning bajarilish tartibi o'sish tartibida berilgan:

shartli operator: "?:";

mantiqiy "YOKI": "||";

mantiqiy "VA": "&&";

bitli "YOKI": "|";

bitli "XOR": "^";

bitli "VA": "&";

tenglikka solishtirish: "=", "!=";

solishtirish: "<", "<=", ">", ">=";

bitli siljish: "<<", ">>", ">>>";

qo'shish, ayirish: "+", "-";

ko'paytirish, bo'lish: "\*", "/", "%";

inkor qilish, orttirish, kamaytirish: "!", "~", "++", "--";

qavslar: "()", "[ ]".

*Funktsiyalar:*

JavaScriptda huddi C yoki Javadagi kabi protseduralar va protsedura-funktsiyalar funktsiya deb ataladi. Funktsiyalarni e'lon qilish quyidagilarni o'z ichiga oladi:

- ✓ Qabul qilingan **function** so'zi;
- ✓ Funktsiyaning nomi;
- ✓ Yumaloq qavs ichiga olingan va vergul bilan ajratilgan funktsiya argumentlari;
- ✓ Figuraviy qavs ichiga olingan funktsiyaning asosiy qismi.

Ya'ni:

```
function myFunction(arg1, arg2, ...)
{
 ...
 Operatorlar ketma-ketligi
 ...
}
```

---

Bu yerda `myFunction` — funktsiyaning nomi, `arg1`, `arg2` — formal parametrlar roʻyxati

**Misol:**

```
function Factorial(n)
{
 if((n<0)||(round(n)!=n))
 {
 alert("Factorial funktsiyasi "+n argumentida aniqlanmadi);
 return NaN;
 }
 else
 {
 result=(n*Factorial(n-1));
 return result;
 }
}
```

Funktsiya qabul qilingan **return** soʻzi yordamida qiymatni qaytarmasligi mumkin.

**Misol:**

```
function Greeting(s)
{ document.write("Hello,"+s+"!");
}
```

Funktsiyani chaqirish aniq bir qiymatlar orqali bajariladi.

**Misol:**

`Factorial(2+1)`; — 6 ni qaytaradi,  
`Greeting("world")`; — ekranga "Hello, world!" satrini chiqaradi.

Har bir funktsiya, masalan, **myFunction** funktsiyasi argumentlari **arguments** deb nomlangan massiv koʻrinishida saqlanuvchi **myFunction** deb nomlanuvchi obʻjekt hisoblanadi, bunda uning argumentlariga quyidagicha murojaat qilinadi:

`myFunction.arguments[i]`,

bu yerda `i` — argument nomeri (nomerlash 0 dan boshlanadi).

Funktsiyaning sonli argument qiymatlari soni funktsiya eʼlon qilingandagi formal parametrlari soniga teng yoki undan ortiq boʻlishi kerak. Bunda funktsiyani chaqirishdagi argumentlar soni `myFunction.arguments.length` maydonida saqlanadi va bu maydonning qiymatini qaytadan berish orqali dinamik ravishda oʻzgartirish mumkin.

**Misol:** Hujjatda HTML formatida roʻyxatni chiqarish.

Bu yerda birinchi argument (`ListType`) tartiblangan roʻyxat uchun "o" yoki "O" qiymatni olish va tartiblanmagan roʻyxat uchun "u" yoki "U" qiymatni olishi mumkin. Keyin roʻyxatning elementlari keladi.

```
function myList(ListType)
{
 document.write("<" + ListType + "L");
 for(var i=1; i < myList.arguments.length; i=i+1)
 { document.write("" + myList.arguments[i]);
```

---

```
}
document.write("</"+ListType+"L>");
}
```

HTML hujjatda bu funktsiyani chaqirish:

```
<script>
myList("0", "bir", 2, "3")
</script>
```

Quyidagi matnni chiqaradi:

```
bir
2
3
```

*Shartli (nisbiy) o‘tishli funktsiyalar*

if shartli o‘tish operatori

if operatori sintaksisining birinchi varianti:

*if(shart)*

```
{
 tasdiq
}
```

Bu yerda tasdiq – operator yoki operatorlar ketma-ketligi.

Bu holatda shartli operator quyidagicha ishlaydi: avval shart tekshiriladi va agar uning qiymati **true** ko‘rinishida bo‘lsa, tasdiq bajariladi. Aks holda **if** dan keyin keluvchi operator bajariladi.

if operatori sintaksisining ikkinchi varianti:

```
if(shart)
{
 tasdiq1
}
else
{
 tasdiq2
}
```

Bu holda avval shart tekshiriladi va agar uning qiymati **true** ko‘rinishda bo‘lsa, tasdiq1 bajariladi, aks holda, ya’ni **false** bo‘lsa, tasdiq2 bajariladi.

Shartli o‘tish operatorining ishlatilishiga misol:

```
function checkData()
{
 if (document.form1.threeChar.value.length==3)
 {return true;
 }
 else
 {alert('3 ta simvol kiriting');
 return false;
 }
}
```

*switch* tanlash operatori:



---

```

switch (ifoda)
{
 qiymat1: operator1 break;
 qiymat2: operator2 break;
 //...
 default: operatorN break;
}

```

Tanlash operatori quyidagi tartibda ishlaydi: avval ifodaning qiymati hisoblanadi, keyin uning qiymat1 bilan tengligi tekshiriladi va agar u teng bo'lsa, operator1 bajariladi, keyin ifoda qiymatining qiymat2 bilan tengligi tekshiriladi va agar u teng bo'lsa operator2 bajariladi va hokazo. Agar ifoda qiymati hech bir qiymat: qiymat1, qiymat2, v.h.larga teng bo'lmasa, u holda o'z-o'zidan operatorN bajariladi.

### **Sikli funktsiyalar**

```

for tsikli
for (boshlang'ich qiymat sektsiyasi;shart sektsiyasi; hisoblagich o'zgarishi
sektsiyasi)
{
 tasdiq
}

```

Bu sektsiyalardan har biri ham bo'sh bo'lishi mumkin. Boshlang'ich qiymat berish va hisoblagich o'zgarishi sektsiyalarida ifodalar ketma-ketligini vergul bilan ajratgan holda yozish mumkin. Siklni bajarilishi quyidagi tartibda bo'ladi. Birinchi boshlang'ich qiymat sektsiyasi bajariladi. Keyin shart tekshiriladi. Agar shartning qiymati **true** bo'lsa, u holda siklning asosiy qismi (*tasdiq*) bajariladi, keyin hisoblagich o'zgartirgich sektsiyasi bajariladi. Agar shartning qiymati **false** bo'lsa, sikldan chiqiladi.

### **Misol:**

```

function HowMany(SelectObject)
{
 var numberSelected=0
 for (i=0; i< SelectObject.options.length; i++)
 {
 if (SelectObject.options[i].selected==true) numberSelected++;
 }
 return numberSelected;
}

```

for operatori ob'yektdagi barcha maydonlarni ko'rib chiqish uchun ishlatilishi mumkin (keyingi ob'yektli model haqidagi bo'limni qarang).

### Sintaksis:

```

for (o'zgaruvchi in ob'yekt)
{
 ifoda
}

```

---

Bunda ob'yektdagi ko'rsatilgan o'zgaruvchining barcha mumkin bo'lgan qiymatlari hosil qilinadi va ularning har biri uchun tasdiq bajariladi.

**Misol:** student sinfini va shu sinfnig ob'yekti (nusxa) Helenni yaratamiz.

```
function student(name, age, group)
{
 this.name=name;
 this.age=age;
 this.group=group;
}
function for_test(myObject)
{
 for(var i in myObject)
 {
 document.write("i="+i+" => "+myObject[i]+"/n");
 }
};
Helen=new student("Helen K.", 21, 409);
for_test(Helen);
```

Ekkranga chiqarish:

i=0 => Helen K.

i=1 => 21

i=2 => 409

while sikli

*while(shart)*

{

*ifoda*

}

while siklinig bajarilishi shartni tekshirishdan boshlanadi. Agar uning qiymati **true**ga teng bo'lsa, sikl bajariladi, aks holda boshqaruv sikldan keyingi operatorga beriladi.

while operatorining ishlatilishiga misol:

```
n1=10
```

```
n=0
```

```
x=0
```

```
while(n<n1)
```

```
{
```

```
n=n+1;
```

```
x=x+n;
```

```
}
```

Sikllarni bajarilishini to'htatib qo'yuvchi operatorlar **for** va **while**, sikllarning joriy bajarilishlarini to'htatish uchun **break** operatori ishlatiladi.

break operatorining ishlatilishiga misol:

```
function test(x)
```

```
{
```

```
var j=0;
```

---

```
var sum=0;
while(n<n1)
{
if(n==x)
{ sum=x;
break;
}
sum=sum+n;
n=n+1;
}
return sum;
}
```

*continue* operatori **for** va **while**larning ichida joriy iteratsiyani bajarilishini to'xtatadi va keyingi iteratsiyaga o'tishni ta'minlaydi.

*continue* operatorining ishlatilishiga misol:

```
function test1(x)
{
var j=0;
while(n<n1)
{ if(n==x)
{ sum=x;
continue;
}
sum=sum+n;
n=n+1;
}
return sum;
}
```

Bu yerda biz JavaScript tilining asoslari, ma'lumotlarning tiplari, ma'lumotlar ustida bajariladigan asosiy operatorlar va dasturning bajarilishini boshqarish kabi tushunchalarni hamda JavaScript-kodlarni HTML-hujjat ichiga joylashtirish usullarini ko'rib chiqdik.



### Savol va topshiriqlar:

1. PHPda qanday operatorlar bor?
2. PHPda ifoda nima?
3. PHPda qanday funktsiyalar bor?
4. Operatorlarning tartibi nima?

---

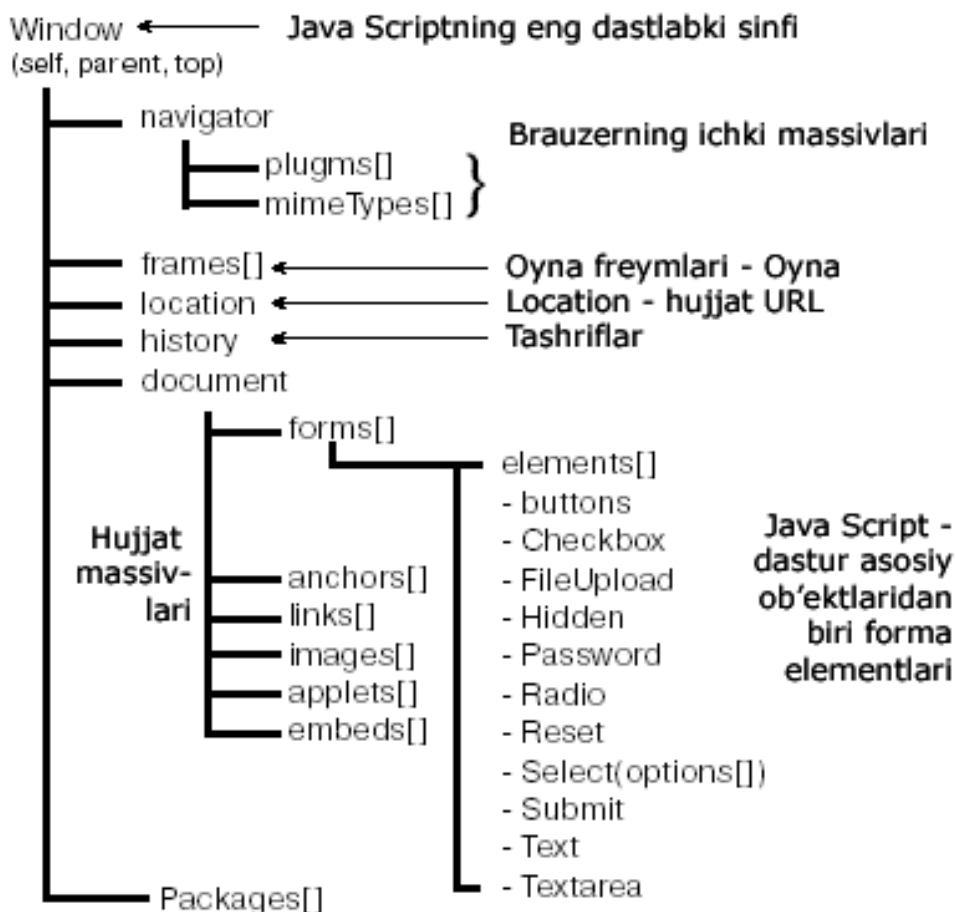
## JavaScriptda ob'yektlarga asoslangan dasturlash.

Bu yerda biz JavaScriptda ob'yektlarga asoslangan dasturlash va hujjat (Web-sahifa)ning ob'yektli modelini ko'rib chiqamiz.

Ob'yektlarga asoslangan dasturlash tillarida sinflar ob'yektlarining ierarhiyasi mavjudligi faraz qilinadi. JavaScriptda bunday ierarhiya Windows sinfining ob'yektlaridan boshlanadi, ya'ni har bir ob'yekt u yoki bu oynaga biriktirilgan bo'ladi. Har bir ob'yektga yoki uning hossasiga murojaat qilish uchun bu ob'yektning yoki uning hossasining shu ob'yekt kirgan ierarhiyadagi eng yuqori turgan ob'yektdan boshlab to'liq yoki qisman nomi ko'rsatiladi.

Hoziroq shuni aytishimiz mumkinki, ob'yektli modelning keltirilgan sxemasi Netscape Navigatorning 4 va undan yuqori versiyalari hamda Microsoft Internet Explorerning 4 va undan yuqori versiyalari uchun to'g'ri. Yana bir marta ta'kidlab o'tamizki, Internet Explorer va Netscape Navigatorlarda ob'yektli modellar turlichadir, keltirilgan sxema esa ularning umumiy qismlari asosida tuzilgan.

Umuman olganda *JavaScript* klassik ob'yektlarga asoslangan til hisoblanmaydi (uni yana yengillashtirilgan ob'yektli til deb ham atashadi). Unda meros qilib olish va polimorfizm yo'q. Dasturchi o'zining ob'yektlari sinfini **function** operatori yordamida aniqlab olishi mumkin, lekin ko'pincha standart ob'yektlar, ularning konstruktorlaridan foydalanadi va sinflarning destruktorlaridan umuman foydalanmaydi. Bu shu bilan tushintiriladiki, *JavaScript*-dasturlarning bajarilish doirasi joriy oynadan tashqariga chiqmaydi.



Ba'zida JavaScriptdagi turli ob'ektlarda bir hil nomdagi hossalari aniqlangan bo'ladi. Bu holatda dasturchi qaysi ob'ektning hossasidan foydalanmoqchi ekanligini aniq ko'rsatishi kerak. Masalan, Window va Documentlar location hossasiga ega bo'ladi. Faqat Window uchun bu Location sinfining ob'ekti, Document uchun esa qiymat sifatida yuklanayotgan hujjatning URLini oluvchi satrli imkoniyatdir.

Yana shuni hisobga olish kerakki, ko'pgina ob'ektlar uchun ob'ektlar hossalari qiyamatlarini oddiy o'zgaruvchilarga aylantirish uchun standart metodlar mavjud. Masalan, barcha ob'ektlar uchun simvollar satriga aylantirish uchun metod aniqlangan: **toString().location** bilan misolda agar satrli kontekstda **window.location** ga murojaat qilinsa, aylantirish o'z-o'zidan bajariladi va dasturchi buni payqamaydi:

```

<SCRIPT>
document.write(window.location);
document.write("
");
document.write(document.location);
</SCRIPT>

```

Lekin baribir farq bor va anchagina. Huddi shu misolda satrli konstantaning uzunligini olamiz:

```

<SCRIPT>
w=toString(window.location);
d=toString(document.location);

```

---

```
h=window.location.href;
document.write(w.length);
document.write(d.length);
document.write(h.length);
</SCRIPT>
```

Shunga osongina ishonch hosil qilish mumkinki, URL tipidagi ob'yektning hossasiga murojaat qilinsa, **location** hossasi aynan shu tipdagi ob'yekt hisoblanadi, almashtirishdan keyin simvollar satri uzunligi boshqacha bo'ladi.

*Barcha ob'yektlar uchun standartlar: hossalar va hodisalar metodlari*

Klient tomonida sahifalar ustidan boshqaruv mexanizmini yaratish uchun hujjatning ob'yektli modelidan foydalanish taklif qilingan. Modelning asosiy ma'nosi shundaki, har bir HTML-konteyner – bu quyidagi uchlik bilan karakterlanuvchi ob'yekt:

1. *hossalar*,
2. *metodlar*,
3. *hodisalar*.

Ob'yektli modelni sahifalar va brauzer orasidagi bog'lanish usuli sifatida tasavvur qilish mumkin. Ob'yektli model – bu brauzerning dasturiy ta'minotida ishtirok etuvchi va ro'y beruvchi, ular yordamida HTML kodi va sahifadagi senariy matnlari bilan ishlash qulay bo'lgan ko'rinishdagi ob'yektlar, metodlar va hodisalarning tasvirlanishidir. Biz uning yordamida brauzerga nima xohlashimizni bildirishimiz va unga mos ravishda ekrandagi sahifani o'zgartirishimiz mumkin.

Bir hil hossalar, metodlar va hodisalar to'plamlariga ega bo'lgan ob'yektlar bir hil tipli ob'yektlar sinflariga birlashtiriladi. Sinflar – bu bo'lishi mumkin bo'lgan ob'yektlarning tarflanishidir. Ob'yektlarning o'zlari esa faqatgina brauzer tomonidan hujjatni yuklagandan so'ng yoki dastur ishi natijasi sifatida paydo bo'ladi. Buni yo'q ob'yektga murojaat qilmaslik uchun doimo esda tutish kerak.

*Hossalar.*

Ko'pgina HTML-konteynerlar atributlarga ega bo'ladi. Masalan, yakor konteyneri `<A ...>...</A>` uni gipermatnli o'tishga aylantiruvchi HREF atributiga ega bo'ladi:

```
tuit
```

Agar yakor konteyneri `<A ...>...</A>` ni ob'yekt sifatida ko'radigan bo'lsak, u holda HREF atributi “yakor” ob'yektining *hossasini* beradi. Dasturchi atribut qiymatini, shu bilan ob'yektning hossasini o'zgartirishi mumkin:

```
document.links[0].href="tuit.html";
```

Barcha atributlarda ham qiymatlarni o'zgartirish mumkin emas. Masalan, grafik rasmning balandligi va eni sahifada rasmni tasvirlashda dastlabki yuklanganiga ko'ra aniqlanadi. Barcha keyingi rasmlar avvalgisiga ko'ra masshtablashtiriladi. Shuni ta'kidlab o'tish kerakki, Microsoft Internet Explorerda rasmning o'lchamlari o'zgartirilishi mumkin.

Tasvirning umumiyliigi uchun JavaScriptda hossalar bilan HTML-bo'laklarda muqobili bo'lmagan ob'yektlar beriladi. Masalan, Navigator ob'yekti yoki brauzer oynasi deb ataluvchi, JavaScriptdagi umuman eng yuqori ob'yekt hisoblanuvchi bajarish muhiti.

---

Metodlar.

*JavaScript* terminologiyasida *ob'yekt metodlari* deganda uning *hossalarini* o'zgartiruvchi funktsiyalar tushiniladi. Masalan, "dokument" ob'yekti bilan `open()`, `write()`, `close()` metodlari bog'langan. Bu metodlar hujjatni hosil qilish yoki uning mazmunini o'zgartirish imkoniyatini beradi. Quyidagi oddiy misolni ko'ramiz:

```
function hello()
{ id=window.open("", "example", "width=400, height=150");
 id.focus(); id.document.open();
 id.document.write("<H1>Salom!</H1>");
 id.document.write("<HR><FORM>");
 id.document.write("<INPUT TYPE=button VALUE='Oynani yopish' ");
 id.document.write("<onClick='window.opener.focus();window.close();>");
 id.document.close();
}
```

Bu misolda `open()` metodi hujjatga yozish oqimini ochadi, `write()` metodi bu yozishni amalga oshiradi, `close()` metodi hujjatga yozish oqimini yopadi. Bularning barchasi oddiy faylga yozishdagi kabi ro'y beradi. Agar oynada status maydonchasi bo'lsa (odatda unda hujjatni yuklash darajasi tasvirlanadi), yopilmagan hujjatga yozish oqimi bo'lgan holda unda huddi hujjatni yuklashdagi kabi yozishni davom etayotganligini ko'rsatuvchi to'rtburchak tasvirlanadi.

### Hodisalar

Metodlar va hossalardan tashqari ob'yektlar hodisalar bilan ham xarakterlanadi. Hususan, *JavaScript*da dasturlashning mazmuni ana shu hodisalarni qayta ishlovchilarni yozishdan iboratdir. Masalan, `button` tipidagi ob'yekt (`button` tipidagi `INPUT` konteyneri – "Tugmacha") bilan `click` hodisasi bo'lishi mumkin, ya'ni foydalanuvchi tugmachani bosishi mumkin. Buning uchun `INPUT` konteynerining atributlari `click` hodisasini qayta ishlovchi atribut – `onClick` bilan kengaytirilgan. Bu atributning qiymati sifatida `HTML`-hujjatning muallifi *JavaScript*da yozadigan hodisani qayta ishlash dasturi ko'rsatiladi:

```
<INPUT TYPE=button VALUE="Bosing"
onClick="window.alert('Iltimos, yana bir marta bosing');">
```

Hodisalarni qayta ishlovchilar shu hodisalar bog'langan konteynerlarda ko'rsatiladi. Masalan, `BODY` konteyneri butun hujjatni hossalarni aniqlaydi, shuning uchun butun hujjatni yuklanib bo'lish hodisasini qayta ishlovchi bu konteynerda **onLoad** atributining qiymati sifatida ko'rsatiladi.

Qat'iy qilib aytganda har qanday brauzer, xoh u *Internet Explorer* yoki *Netscape Navigator* yoki *Opera* bo'lsin, o'zining ob'yektli modeliga ega bo'ladi. Turli hil brauzerlar (va hatto bitta brauzerning turli hil versiyalari)ning ob'yektli modellari bir-birlaridan farq qiladi, lekin printsiplial jihatdan bir hil strukturaga ega bo'ladi. Shuning uchun ularning har biriga alohida to'htalib o'tishdan ma'no yuq. Biz barcha brauzerlarga qo'llash mumkin bo'lgan umumiy yondashuvlarni va ayrim hollarda ular orasidagi farqlarni ko'rib chiqamiz.

*Window (oyna) ob'yekti*

`Window` ob'yektlar sinfi – bu *JavaScript*da ob'yektlar ierarhiyasidagi eng yuqori sinf. Unga `Window` ob'yekti va `Frame` ob'yekti kiradi. `Window` ob'yekti

---

brauzer-dastur oynasi bilan, Frame ob'yekti esa HTML-sahifa muallifi tomonidan FRAMESET va FRAME konteynerlaridan foydalanganda brauzer oynasi tomonidan hosil qilinuvchi va uning ichida joylashgan oynalar bilan bog'lanib ketadi.

JavaScriptda dasturlashda ko'pincha Window tipidagi ob'yektlarning quyidagi hossalari va metodlari ishlatiladi:

Hossalar	Metodlar	Hodisalar
status	open()	Hodisalar yo'q
location	close()	
history	focus()	
navigator		

Window ob'yekti faqatgina oynani ochish vaqtidagina yaratiladi. Sahifani oynaga yuklashda paydo bo'ladigan barcha qolgan ob'yektlar Window ob'yektining hossalari. Shunday qilib turli hil sahifalarni yuklashda Windowning hossalari turlicha bo'lishi mumkin.

*Status maydoni (holatlar paneli).* Status maydoni — bu HTML-sahifa mualliflari JavaScriptdan foydalanishni boshlagan birinchi narsadir. Kalkulyatorlar, o'yinlar, matematik hisoblashlar va boshqa elementlar judayam sun'iy ko'rindi. Buning natijasida status maydonidagi yugurib yuruvchi satr Webdan foydalanuvchilarning diqqatini haqiqatdan torta oladigan durdona bo'lgan edi. Asta sekin uning ommaviyligi yo'qola bordi. Yuguruvchi satrlar juda kam qo'llaniladigan bo'ldi, lekin status maydonchasini dasturlash ko'pgina Web-bo'g'imlarida uchraydi.

Status maydoni (status bar) deb HTML-sahifani tasvirlovchi sohaning shundoq ostidagi brauzer oynasining quyi qismidagi o'rta maydonga aytiladi. Status maydonida brauzer holati (hujjatni yuklash, grafikani yuklash, yuklashni tugatish, appletni bajarish va h.) haqidagi ma'lumot tasvirlanadi. JavaScriptdagi dastur bu maydon bilan oynaning o'zgaruvchi hossasi kabi ishlash imkoniyatiga ega. Bunda amalda u bilan 2 ta turli hildagi hossalari bog'langan bo'ladi:

1. *window.status;*
2. *window.defaultStatus.*

Ular o'rtasidagi farq shundaki, amalda brauzer ba'zi bir hodisalar bilan bog'liq bo'lgan bir nechta holatlarda bo'ladi. Brauzer holati status maydonidagi habarda aks etadi. Umuman olganda faqatgina 2 ta holatlar mavjud: hech qanday hodisalar yo'q (defaultStatus) va qandaydir hodisalar ro'y berayapti (status).

*statusni dasturlaymiz.* status hossasi sahifani oddiy yuklashdan farq qiluvchi hodisalar to'g'risidagi habarlarni tasvirlash bilan bog'liq. Masalan, sichqoncha kursori gipermatnli o'tish ustidan o'tayotganda, HREF atributida ko'rsatilgan URL status maydonchasida aks ettiriladi. Sichqoncha kursori gipermatnli o'tishdan holi bo'lgan maydonga o'tishi bilan status maydonchasida boshlang'ich habar tiklanadi (Document: Done). Bu texnika berilgan sahifada status va defaultStatuslarni bayon qilishda amalga oshirilgan.

```
<A HREF=#status onMouseover="window.status='Jump to status description'; return true;"
```



---

```
onMouseout="window.status='Status bar programming'; return true;">
window.status
```

JavaScriptning ma'lumotnomalarida ko'rsatilganki, **mouseover** va **mouseout** hodisalarni qayta ishlovchilar **true** qiymatini qaytarishi kerak. Bu brauzer boshlang'ich harakatlarni bajarmasligi uchun kerak bo'ladi. Tajriba shuni ko'rsatadiki, Netscape Navigator 4.0 true qiymati holatida ham juda yahshi ishlaydi.

*defaultStatusni dasturlaymiz.* defaultStatus hossasi hech qanday hodisa ro'y bermayotgan vaqtda status maydonida aks ettirilayotgan matnni aniqlaydi. Bizning hujjatda biz uni hujjatni yuklash vaqtida aniqladik:

```
<BODY onLoad="window.defaultStatus='Status bar programming';">
```

Bu habar sahifaning barcha komponentalari (matn, grafika, appletlar va h.) yuklab bo'lingandan keyin paydo bo'ladi. U hujjatni ko'rishda ro'y berishi mumkin bo'lgan xohlagan hodisadan qaytgandan keyin status maydonchasida qayta tiklanadi. Qizig'i shundaki, sichqonchani gipermatnli o'tishlardan holi bo'lgan sohalar bo'ylab harakatlantirilishi **defaultStatusni** doimo aks ettirilishiga olib keladi.

*location maydoni (adres satri).* Adres maydonida yuklangan hujjatning URLi aks ettiriladi. Agar foydalanuvchi o'zicha qaysidir sahifaga o'tmoqchi bo'lsa (uning URLini terib), u buni location maydonida amalga oshiradi. Maydon brauzer oynasining yuqori qismida, instrumentlar panelidan quyida, lekin shahsiy tanlovlar panelidan yuqorida joylashgan.

Umuman olganda Location – bu ob'yekt. JavaScript versiyalaridagi o'zgarishlar tufayli **Location** quyi sinf sifatida Window sinfiga ham, Document sinfiga ham kiradi. Biz **Locationni** faqatgina **window.location** sifatida ko'rib chiqamiz. Bundan tashqari Location – bu Area va Link sinfining ob'yektlari kiruvchi URL sinfining quyi sinfi hamdir. Location URLning barcha hossalarini meros qilib oladi va bu unga URL sxemasining istalgan qismiga kirishga imkon beradi.

Location ob'yektining xarakteristikalarini va ishlatish usullarini ko'rib chiqamiz:

- hossalar;
- metodlar;
- Locationni xarakterlovchi hodisalar yuq.

Ko'rib turibmizki, **Location** ob'yektining xarakteristikalari ro'yxati to'liq emas.

*Hossalar.* Aytaylik, brauzer quyida berilgan adresdagi sahifani aks ettirayotgan bo'lsin: <http://tuit.uz:80/r/dir/page?search#mark>

U holda Location ob'yektining hossalari quyidagi qiymatlarni qabul qilishi mumkin:

```
window.location.href = http://tuit.uz:80/r/dir/page?search#mark
window.location.protocol = http;
window.location.hostname = tuit.uz;
window.location.host = tuit.uz:80;
window.location.port = 80
window.location.pathname = /r/dir/;
window.location.search = search;
window.location.hash = mark;
```

---

Metodlar. Location metodlari sahifani yuklashni va qayta yuklashni boshqarish uchun mo'ljallangan. Bu boshqaruv shuni bildiradiki, hujjatni qayta yuklash (reload) yoki yuklash (replace) mumkin. Bunda sahifalarni ko'rish yo'liga (history) ma'lumot kiritilmaydi:

```
window.location.reload(true);
window.location.replace('#top');
```

reload() metodi instrumentlar panelidagi Reload tugmachasini bosgandagi brauzer harakatini to'raligicha modellaydi. Agar bu metodni argumentsiz yoki unga true qiymat bergan holda chaqirilsa, brauzer hujjatning ohirgi o'zgartirilgan vaqtini tekshiradi va uni yoki keshdan (agar hujjat o'zgartirilmagan bo'lsa) yoki serverdan yuklaydi. Bunday harakat Reload tugmasining oddiygina bosish bilan mos keladi. Agar argument sifatida false ko'rsatilsa, u holda brauzer hujjatni har qanday holatda ham serverdan yuklaydi. Bunday harakat Reload va Shift tugmachalarini birgalikda bosish bilan mos keladi (Reload+Shift).

replace() metodi bir sahifani ikkinchisi bilan shunday almashtirishga imkon beradiki, bu almashtirish HTML-sahifalarni ko'rish yo'lida aks ettirilmaydi va instrumentlar panelidagi Back tugmachasini bosish orqali foydalanuvchi doimo dastlabki oddiy usulda (gipermatnli o'tish bo'yicha) yuklangan sahifaga qaytadi. Eslatib o'tamizki, Location hossasini o'zgartirishda ham sahifani qayta yuklash ro'y beradi, lekin bu holda bu o'tish haqidagi ma'lumot **history**ga kiritiladi.

*Tashriflar ro'yxati (History)*. World Wide Web sahifalariga tashriflar ro'yxati faoydalanuvchiga u bir necha minut (soat, kun) oldin ko'rgan sahifaga qaytish imkoniyatini beradi. Tashriflar ro'yxati JavaScriptda history sinfining ob'yektiga aylantiriladi. Bu ob'yekt foydalanuvchi ko'rgan va brauzer menyusidagi GO rejimini tanlagan holda olishi mumkin bo'lgan URL-sahifalar massivini ko'rsatadi. History ob'yekti metodlari shu massivdagi URLdan foydalangan holda sahifalarni yuklashga imkon beradi.

Brauzer havfsizligi bilan muammolar bo'lmasligi uchun History bo'yicha faqatgina URLning indeksi bo'yicha sayr qilish mumkin. Bunda URL matnli qator sifatida dasturchiga berilmaydi. Ko'pincha bu ob'yekt bir nechta turli hil sahifalarga o'tishlar bo'lgan misollar yoki sahifalarda misol yuklanadigan sahifaga qaytish mumkin deb faraz qilgan holda foydalaniladi:

```
<FORM><INPUT TYPE=button VALUE="Orqaga"
onClick=history.back()></FORM>
```

Berilgan kod bosish orqali oldingi sahifaga qaytishimiz mumkin bo'lgan "Orqaga" tugmasini aks ettiradi.

*Brauzer tipi ( Navigator ob'yekti )*. Brauzerlar o'rtasidagi raqobat tufayli (uni allaqachon Microsoft Internet Explorer foydasiga hal bo'ldi deb hisoblash mumkin) sahifani aniq bir ko'rish dasturiga moslashtirish muammosi vujudga keldi. Bunda ikki hil variant bo'lishi mumkin: server tomonidagi brauzer tipini aniqlash va klient tomonidagi brauzer tipini aniqlash. Ohirgi variant uchun JavaScriptda Navigator ob'yekti mavjud.

Ko'rish dasturining tipini aniqlashga oddiy misolni ko'ramiz:  
<FORM><INPUT TYPE=button VALUE="Navigator tipi "  
onClick="window.alert(window.navigator.userAgent);"></FORM>

---

Tugmachani bosish bilan ogohlantirish oynasi aks ettiriladi. Unda tegishli brauzer HTML-sarlavhaga joylashtiradigan userAgent satri bo‘ladi.

Bu satrni komponentalar bo‘yicha bo‘laklash mumkin, masalan:

```
navigator.appName = Microsoft Internet Explorer
```

```
navigator.appCodeName = Mozilla
```

```
navigator.appVersion = 4.0 (compatible; MSIE 5.5; Windows 98)
```

```
navigator.userAgent = Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)
```

Navigator ob‘yektini dasturlash nuqtai nazaridan qiziqarli bo‘lgan bir necha hil qo‘llash usullari mavjud. Masalan, Javani qo‘llash mumkinligini tekshirish.

Bu imkoniyatni misolda ko‘rsatamiz:

```
<SCRIPT>
```

```
document.write("<P ID=red>");
```

```
if(navigator.javaEnabled()==true)
```

```
 document.write("Sizning dasturingiz Java-appletlar ijrosini qo‘llab-quvvatlaydi");
```

```
if(navigator.javaEnabled()==false)
```

```
 document.write(" Sizning dasturingiz Java-appletlar ijrosini
qo‘llab-quvvatlamaydi ");
```

```
</SCRIPT>
```

```
</example>
```

Shunga o‘xshab sizning brauzeringizda ishlaydigan grafik fayllar formatlarini ham tekshirish mumkin:

```
<SCRIPT>
```

```
if(navigator.mimeTypes['image/gif']!=null)
```

```
 document.write("Sizning brauzeringiz GIFni qo‘llab-quvvatlaydi
");
```

```
if(navigator.mimeTypes['image/tif']==null)
```

```
 document.write("Sizning brauzeringiz TIFFni qo‘llab-quvvatlamaydi ");
```

```
</SCRIPT>
```

Afsuski, bunday tekshiruv grafikani avtomatik yuklashni mavjudligini aniqlashga imkon bermaydi.

**Oynalarni boshqarish.** Oynalar bilan nimalar qilish mumkin? Ochish (yaratish), yopish (yo‘qotish), uni boshqa barcha ochiq oynalar ustiga joylashtirish. Bundan tashqari oynaning va unga bo‘ysungan ob‘yektlar hossalarni boshqarish mumkin. Asosiy hossalarning bayoni “Brauzer oynasining hossalarni dasturlaymiz” bo‘limida berilgan, shuning uchun asosiy e‘tiborni oddiy va ko‘proq ishlatiladigan oynalarni boshqarish metodlariga qaratamiz:

- *alert()*;
- *confirm()*;
- *prompt()*;
- *open()*;
- *close()*;
- *focus()*;
- *setTimeout()*;
- *clearTimeout()*.

Bu yerda faqat ikki metod: *scroll()* va *blur()* lar ko‘rsatilmagan.

---

Birinchisi oynani berilgan pozitsiyaga joylashtirishga imkon beradi. Lekin uni oynaning koordinatalarini bilmagan holda ishlatish juda qiyin. Keyingisi esa oddiy ish hisoblanadi agar qatlamlarni dasturlash texnologiyalari yoki CSS (Cascading Style Sheets)lar ishlatilmayotgan bo'lsa.

Ikkinchi metod diqqatni oynadan oladi. Bunda diqqatni qaerga qaratish umuman ma'lum bo'lmaydi. Yahshisi diqqatni yo'qotgandan ko'ra uni aniq maqsadli yo'naltirgan ma'qul.

*window.alert()*

alert() metodi ogohlantirish oynasini chiqarishga imkon beradi:

```

```

```
So'rovni qaytaring!
```

Hammasi juda oddiy, lekin shuni nazarda tutish kerakki, habarlar sistema shriftlarida chiqariladi, shuning uchun rus tilidagi habarlarni olish uchun operatsion tizimning mahalliyashtirilgan varianti bo'lishi kerak.

*window.confirm()*

confirm() metodi foydalanuvchiga u ma'qullashi yoki rad qilishi mumkin bo'lgan savollarni berishga imkon beradi:

```
<FORM>
```

```
<INPUT TYPE=button VALUE="Siz JavaScriptni bilasizmi?"
```

```
onClick="if(window.confirm('Hammasini bilaman')==true)
```

```
{ document.forms[0].elements[1].value='Ha'; }
```

```
else {
```

```
 document.forms[0].elements[1].value='Yoq';
```

```
};">

```

```
</FORM>
```

Rus tilidagi habarlar uchun alert() metodi uchun bayon qilingan barcha cheklashlar confirm() metodi uchun ham o'rinli.

*window.prompt()*

prompt() metodi foydalanuvchidan informatsion oynaning kiritish maydoniga teriladigan qisqa matn satrini qabul qilishga imkon beradi:

```
<FORM>
```

```
<INPUT TYPE=button VALUE="Kiritish oynasini ochish"
```

```
onClick="document.forms[1].elements[1].value=window.prompt('Xabarni kiriting');">
```

```
<INPUT SIZE=30>
```

```
</FORM>
```

Foydalanuvchi tomonidan kiritilgan satr istalgan o'zgaruvchi tomonidan o'zlashtirilishi mumkin va keyin JavaScript-dasturda ishlatish mumkin.

*window.open()*

Oynaning bu metodida atributlar boshqa ob'yektlarnikiga qaraganda ko'proq. Open() metodi yangi oynalarni yaratishga mo'ljallangan. Umumiy holda uning sintaksisi quyidagi ko'rinishda bo'ladi:

```
open("URL", "window_name", "param,param,...", replace);
```

bu yerda: URL — yangi oynaga yuklanadigan sahifa, window\_name — A va FORM konteynerlaridagi TARGET atributida ishlatish mumkin bo'lgan oyna nomi.

Parametrlar	Ishlatilishi
replace	Oynani ochish vaqtida History massiviga yozishni boshqarishga imkon beradi
param	Parametrlar ro'yxati
width	Oynaning piksellardagi kengligi
height	Oynaning piksellardagi balandligi
toolbar	Brauzerning sistema tugmalari bo'lgan oynani yaratadi
location	location maydonli oynani yaratadi
directories	Foydalanuvchining tanlovlari menyusi oynasini yaratadi
status	Status maydonli oynani yaratadi
menubar	Menyuli oynani yaratadi
scrollbar	Yugurdak polosali oynani yaratadi
resizable	O'lchamini o'zgartirsa bo'ladigan oynani yaratadi

Quyidagi misolni keltiramiz:

<FORM>

```
<INPUT TYPE=button VALUE="Oddiy oyna"
onClick="window.open('about:blank','test1','directories=no,height=200, location=no,
menubar=no, resizable=no, scrollbars=no, status=no, toolbar=no, width=200');">
```

```
<INPUT TYPE=button VALUE="Murakkab oyna"
onClick="window.open('about:blank','test2','directories=yes,
height=200,location=yes,menubar=yes,resizable=yes,
scrollbars=yes,status=yes,toolbar=yes,width=200');">
```

</FORM>

“oddiy oyna” tugmasiga bosib, quyidagi parametrlil oynani hosil qilamiz:

- directories=no – menyusiz oyna
- height=200 – balandligi 200 px
- location=no – location maydoni yo‘q
- menubar=no - menyusiz
- resizable=no – o‘lchamini o‘zgartirish mumkin emas
- scrollbars=no – yugurdak polosasi yo‘q
- status=no – status satri yo‘q
- toolbar=no – brauzerning sistema tugmachalari yo‘q
- width=200 – kengligi 200

“murakkab oyna” tugmasini bosib, quyidagi *oynani* olamiz:

- directories=yes – menyuli oyna
- height=200 - balandlik 200 px
- location=yes - location maydoni mavjud
- menubar=yes – menyu mavjud
- resizable=yes – o‘lchamini o‘zgartirish mumkin
- scrollbars=yes – yugurdak polosasi mavjud
- status=yes – status satri mavjud
- toolbar=yes – brauzerning sistema tugmachasi mavjud
- width=200 - kenglik 200

---

### *window.close()*

close() metodi – bu open() metodining aks tomonidir. U *oynani* yopish imkoniyatini beradi. Ko‘pincha aynan qaysi *oynani* yopish kerak degan savol tug‘iladi.

Agar joriy oynani yopish kerak bo‘lsa, u holda:

```
window.close();
self.close();
```

Agar bosh oynani, ya’ni joriy oyna ocilgan oynani yopish kerak bo‘lsa, u holda:

```
window.opener.close();
```

Agar ixtiyoriy oynani yopish kerak bo‘lsa, u holda avval uning identifikatorini olish kerak bo‘ladi:

```
id=window.open();
```

...

```
id.close();
```

Ohirgi misoldan ko‘rinib turibdiki, oynani nomi bo‘yicha emas (TARGET atributining qiymati bu yerda ahamiyatga ega emas), balki ob’jektning ko‘rsatkichi bo‘yicha yopiladi.

### *window.focus()*

focus() metodi diqqatni u ishlatilgan *oynaga* qaratish uchun ishlatiladi. Diqqatni berish oynani qaysi vaqtda tanlash holatlarini eslamasdan, oynani ochishda ham, yopishda ham juda foydali.

**Misol.** Oynani ochamiz va uni yopmasdan turib yana shu nomdagi, lekin boshqacha matnli oynani ochamiz. Yangi oyna asosiy oynaning ustida paydo bo‘lmaydi, chunki diqqat unga berilmadi. Endi oyna ochilishini takrorlaymiz, faqat bu safar diqqatni berish orqali:

```
function myfocus(a)
```

```
{
```

```
id = window.open("", "example", "scrollbars,width=300,height=200");
```

```
//oynani ochamiz va unga ko‘rsatkich bo‘lgan o‘zgaruvchini kiritamiz
```

```
//agar shu nomli oyna mavjud bo‘lsa, yangi oyna yaratilmaydi,
```

```
//faqatgina shu oynaga yozish uchun oyna ochiladi
```

```
if(a==1)
```

```
{
```

```
id.document.open();
```

```
//yaratilgan oynaga yozish uchun oyna ochamiz
```

```
id.document.write("<CENTER>>Oynani birinchi marta ochdingiz");
```

```
//Bu oynaga yozamiz
```

```
}
```

```
if(a==2)
```

```
{
```

```
id.document.open();
```

```
id.document.write("<CENTER>Oynani ikkinchi marta ochdingiz");
```

```
}
```

---

```

if(a==3)
{
id.focus();
//diqqatni beramiz, keyin oldingi holdagi amallarni bajaramiz
id.document.open();
id.document.write("<CENTER>Oynani uchinchi marta ochdingiz");
}
id.document.write("<FORM><INPUT TYPE=button
onClick='window.close();' VALUE='Oynani yopish'></CENTER>");
id.document.close();
}

```

Yangi oynani eski (bosh) oynadan turib yozayotganligimiz uchun yangi ob'yektga ko'rsatkich sifatida **id** o'zgaruvchining qiymatidan foydalanamiz.

*window.setTimeout()*

*setTimeout()* metodi bajarilishi ikkinchi argumentda ko'rsatilgan millisekundlarcha keyingi qoldiriluvchi yangi hisoblash oqimini ochish uchun foydalaniladi:

```
idt = setTimeout("JavaScript_код",Time);
```

Bu funktsiyaning tipik qo'llanilishi – ob'yektlar hossalarini avtomatik o'zgarishini tashkil qilish. Masalan, forma maydonida soatni qo'yish mumkin:

```

var flag=0;
var idp=null;
function myclock()
{
if(flag==1)
{
d = new Date();
window.document.c.f.value = d.getHours()+":"+d.getMinutes()+":"+d.getSeconds();
}
idp=setTimeout("myclock();",500);
}
function flagss()
{
if(flag==0) flag=1; else flag=0;
}
...
<FORM NAME=c>
Joriy vaqt:<INPUT NAME=f size=8><INPUT TYPE=button VALUE="Start/Stop"
onClick="flagss();myclock();">
</FORM>

```

Shuni nazarda tutish kerakki, oqim har doim, hattoki soat to'htagan holatda ham paydo bo'laveradi. Agar u flag o'zgaruvchisining 1 ga teng qiymatidagina yaratilganida edi, u holda **flag**ning 0 ga teng qiymatida yo'qolgan bo'lar edi va u tugmachani bosishda soatlar to'htab turishda davom etgan bo'lardi.

*window.clearTimeout*

---

clearTimeout() metodi setTimeout() metodi tufayli hosil qilingan oqimni yo‘q qilishga imkon beradi. Aniqki, uni ishlatilishi hisoblash qurilmalari resurslarini samaraliroq taqsimlash imkoniyatini beradi. Bu metodni soatlar bilan bo‘lgan misolda foydalanish uchun, biz funktsiyani va formani moslashtirishimiz kerak:

```
var idp1 = null;
function start()
{
d = new Date();
window.document.c1.f1.value =
d.getHours()+":"+d.getMinutes()+":"+d.getSeconds();
idp1=setTimeout("start();",500);
}
function stop()
{
clearTimeout(idp1);idp1=null;
}
...
<FORM NAME=c1>
Joriy vaqt:<INPUT NAME=f1 size=8>
<INPUT TYPE=button VALUE="Start" onClick="if(idp1==null)start();">
<INPUT TYPE=button VALUE="Stop" onClick="if(idp1!=null)stop();">
</FORM>
```

Berilgan misolda soatlarni to‘xtatish uchun clearTimeout() metodi ishlatiladi. Bunda ko‘plab oqimlarni paydo bo‘lmasligi uchun oqim ob‘yektiga bo‘lgan ko‘rsatkichning qiymati tekshiriladi.

*Hujjat ob‘yekti (window.document). Oynalarni yaratish.*

Brauzerda yangi oynalarni yaratish – JavaScriptning juda katta imkoniyati. Siz yangi oynaga yangi hujjatlarni yuklashingiz (masalan, huddi o‘sha HTML hujjatlarini) yoki (dinamik ravishda) yangi materiallarni *yaratishingiz* mumkin. Avval yangi oynani qanday ochish mumkinligini, keyin esa bu oynaga qanday qilib HTML-sahifani yuklash mumkinligini va nihoyat, uni qanday qilib yopish mumkinligini ko‘rib chiqamiz.

Quyida keltirilgan script brauzerning yangi oynasini ochadi va unga qandaydir web-sahifani yuklaydi:

```
<html>
<head>
<script language="JavaScript">
<!-- hide
function openWin() {
 myWin= open("abc.html");
}
// -->
</script>
</head>
<body>
```



```

<form>
<input type="button" value="Yangi oynani ochish" onClick="openWin()">
</form>
</body>
</html>

```

Keltirilgan misolda yangi oynaga open() metodi yordamida *abc.html* sahifasi yoziladi.

Chunki siz shuni ko‘rishingiz mumkinki, oyna yaratish jarayonining o‘zi ustidan nazorat o‘rnata olasiz. Masalan, siz yangi oyna status satri, instrumentlar paneli yoki menyuga ega bo‘lishi kerakligini ko‘rsata olasiz. Bundan tashqari siz oynaning o‘chamini ham bera olasiz. Masalan, keyingi skriptda 400x300 piksel o‘lchamli oyna ochiladi. U status satriga ham, instrumentlar paneliga ham, menyuga ham ega bo‘lmaydi.

```

<html>
<head>
<script language="JavaScript">
<!-- hide
function openWin2() {
 myWin= open("abc.html", "displayWindow",
 "width=400,height=300,status=no,toolbar=no,menubar=no");
}
// -->
</script>
</head>
<body>
<form>
<input type="button" value="Yangi oynani ochish " onClick="openWin2()">
</form>
</body>
</html>

```

Ko‘rib turganingizdek, oynaning hossalari quyidagi satrda hosil qilamiz *"width=400, height=300, status=no, toolbar=no, menubar=no"*.

Yana shu narsaga e‘tibor beringki, siz bu satrda bo‘sh o‘rin simvollarini joylashtirishingiz kerak emas.

Siz boshqara olishingiz mumkin bo‘lgan oyna hossalari ro‘yxati:

directories	yes no
height	Piksellar soni
location	yes no
menubar	yes no
resizable	yes no
scrollbars	yes no
status	yes no
toolbar	yes no
width	Piksellar soni

---

JavaScript tilining 1.2 versiyasida bir nechta yangi hossalari qo'shildi (ya'ni Netscape Navigator 4.0 da). Siz bu hossalardan Netscape 2.x, 3.x yoki Microsoft Internet Explorer 3.x versiyalar uchun materiallar tayyorlashda foydalanmasligingiz kerak, chunki bu brauzerlar 1.2 JavaScript tilini tushinmaydi. Oynalarning yangi hossalari:

alwaysLowered	yes no
alwaysRaised	yes no
dependent	yes no
hotkeys	yes no
innerWidth	<i>Piksellar soni (width ning o'rniga)</i>
innerHeight	<i>Piksellar soni (height ning o'rniga)</i>
outerWidth	<i>Piksellar soni</i>
outerHeight	<i>Piksellar soni</i>
screenX	<i>Piksellar soni</i>
screenY	<i>Piksellar soni</i>
titlebar	yes no
z-lock	yes no

Siz bu hossalarning ma'nolarini JavaScript 1.2 tilining bayonidan topishingiz mumkin. Bundan keyin ulardan ayrimlariga tushintirishlar va ishlatishga misollar keltiramiz.

Masalan, endi bu yangi hossalardan foydalangan holda yangi oyna ekranning qaysi yerida joylashishi kerakligini aniqlashingiz mumkin. JavaScript tilining eski versiyasida siz buni amalga oshira olmas edingiz.

**Oynaning nomi.** Ko'rib turganingizdek, oyna ochish vaqtida biz uchta argumentdan foydalanishimiz kerak:

```
myWin= open("abc.html", "displayWindow",
"width=400, height=300, status=no, toolbar=no, menubar=no");
```

Ikkinchi argument nima uchun kerak? Bu oynaning nomi. Oldinroq biz uni **target** parametrada qanday ishlatilganini ko'rib o'tdik. Shunday qilib agar siz oynaning nomini bilsangiz, u holda unda quyidagi yozuv yordamida yangi sahifani yuklashingiz mumkin

```

```

Bunda siz mos oynaning nomini ko'rsatishingiz kerak (agar bunday nomdagi oyna mavjud bo'lmasa, shu nomdagi yangi oyna yaratiladi).

E'tibor beringki, *myWin* – bu yangi oyna nomi emas. Lekin shu o'zgaruvchi yordamidagina siz oynaga kirishingiz mumkin va u oddiy o'zgaruvchi bo'lganligidan uning qo'llanish sohasi u aniqlangan skript holos. Shu bilan birga oyna nomi (berilgan holatda u *displayWindow*) – bu brauzerning istalgan oynasida foydalanish mumkin bo'lgan o'ziga hos identifikator.

**Oynalarni yopish.** Siz yana JavaScript tili yordamida oynalarni yopishingiz ham mumkin. Buni amalga oshirish uchun sizga close() metodi kerak bo'ladi. Keling, oldinroq ko'rganimiz kabi yangi oyna ochamiz va unga navbatdagi sahifani yuklaymiz:

```
<html>
```

---

```

<script language="JavaScript">
<!-- hide
function closeIt() {
 close();
}
// -->
</script>
<center>
<form>
<input type="button" value="Yopish" onClick="closeIt()">
</form>
</center>
</html>

```

Endi agar siz yangi oynadagi tugmachani bossangiz u yopiladi. `open()` va `close()` – bu window ob’yektining metodlaridir. Biz shuni yodda tutishimiz kerakki, oddiygina qilib `open()` va `close()` shaklida emas, balki `window.open()` va `window.close()` ko‘rinishida yozish kerak. Ammo bizning holatda `window` ob’yektini tushirib qoldirishimiz mumkin. Agar siz bu o‘bektning metodlaridan birini chaqirmoqchi bo‘lsangiz (va u faqatgina shu ob’yekt uchungina mumkin), u holda siz window prefiksini yozishingiz shart emas.

**Hujjatni dinamik ravishda yaratish.** Endi biz hujjatlarni dinamik ravishda yaratish kabi JavaScriptning ajoyib imkoniyatini ko‘rib chiqishga tayyormiz. Yani JavaScriptda tuzilgan skriptning o‘ziga yangi HTML-sahifalarni yaratishga ruhsat berish mumkin. Bundan tashqari shu yo‘l bilan web-hujjatning VRML-ko‘rinishlar va h. kabi boshqa hujjatlarini ham yaratish mumkin. Qulaylik uchun bu hujjatlarni alohida oyna yoki freymga joylashtirish mumkin.

Avval biz oddiy HTML-hujjat yaratamiz va uni yangi oynada namoyish qilamiz.

Quyidagi skriptni ko‘ramiz.

```

<html>
<head>
<script language="JavaScript">
<!-- hide
function openWin3() {
 myWin= open("", "displayWindow",
 "width=500, height=400, status=yes, toolbar=yes, menubar=yes");
 // keyinchalik bosib chiqarish uchun hujjat ob’yektini ochish
 myWin.document.open();
 // yangi hujjatni ochish
 myWin.document.write("<html><head><title>Tect");
 myWin.document.write("</title></head><body>");
 myWin.document.write("<center>");
 myWin.document.write("Bu HTML-hujjat ochildi");
 myWin.document.write("JavaScript yordamida!");
 myWin.document.write("</center>");

```

---

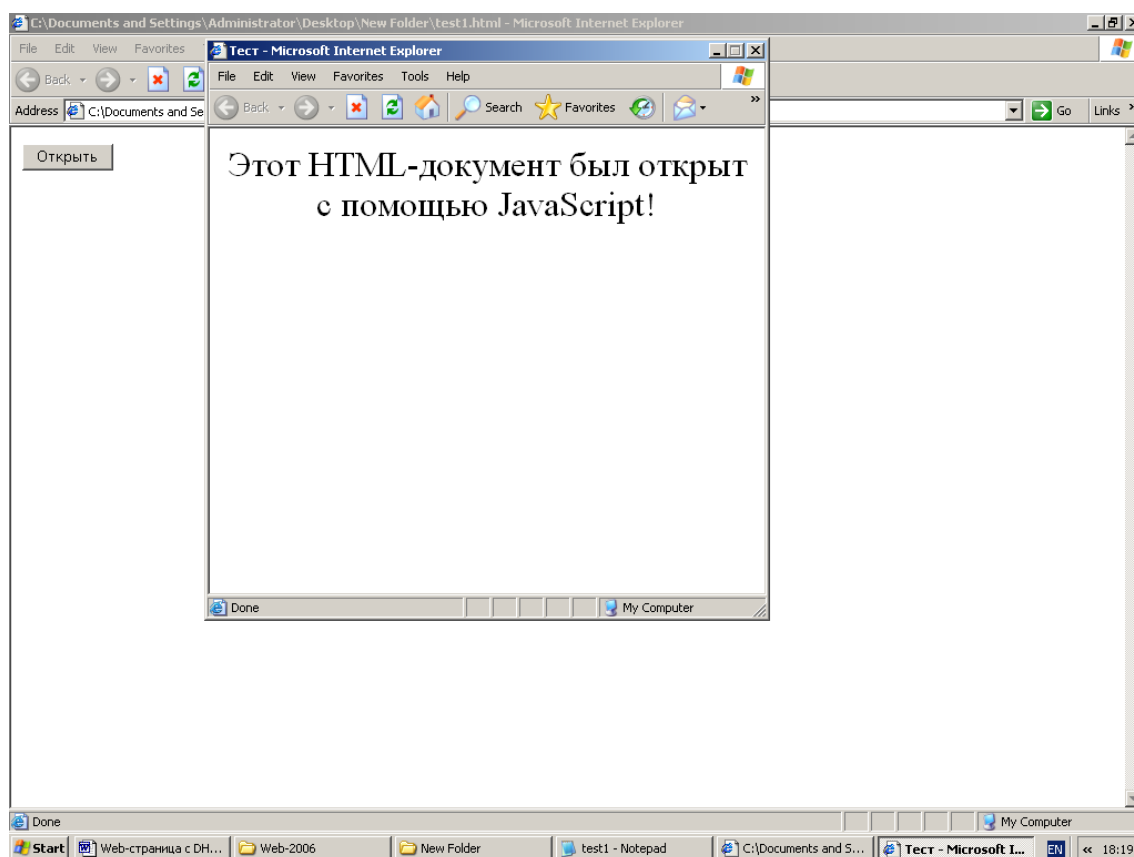
```
myWin.document.write("</body></html>");
// hujjatni yopish - (lekin oynani emas!)
myWin.document.close();
}
// -->
</script>
</head>
<body>
<form>
<input type=button value="Ochish" onClick="openWin3()">
</form>
</body>
</html>
```

Keling `winOpen3()` funksiyasini ko‘rib chiqaylik. Aniqki, biz avval brauzerning yangi oynasini ochamiz. `Open()` funksiyasining birinchi argumenti – bo‘sh satr (“”), bu esa shuni bildiradiki, biz bu holatda aniq bir URL adresini ko‘rsatishni xohlamaymiz. Brauzer faqatgina mavjud hujjatni qayta ishlamasligi kerak – JavaScript qo‘shimcha yangi hujjat yaratishi kerak.

Skriptda biz yangi *myWin* o‘zgaruvchini aniqlaymiz va uning yordamida biz yangi oynaga kirib bora olamiz. Shunga e‘tibor beringki, bu holda bu maqsad uchun oyna nomi (*displayWindow*) dan foydalana olmaymiz.

Yangi oynani ochganimizdan keyin, hujjat ob’yektini yozish uchun navbat keladi. Bu quyidagi buyruq yordamida amalga oshiriladi:

```
// keyinchalik bosmaga chiqarish uchun hujjat ob’yektini ochish
myWin.document.open();
```



Bu yerda biz `open()` ga- hujjat ob'jektining metodiga murojaat qilamiz. Lekin u window ob'jektining `open()` metodi bilan bir hil narsa emas. Bu buyruq yangi oyna ochmaydi – u bosib chiqarish uchun hujjatni tayyorlaydi. Bundan tashqari biz `document.open()` ning oldiga `myWin` ni yangi oynaga yozish imkoniyatini olish maqsadida qo'yishimiz kerak.

Skriptning keyingi satrlarida `document.write()` ni chaqirish yordamida yangi hujjatning matni hosil qilinadi:

```
// yangi hujjatni hosil qilish
myWin.document.write("<html><head><title>On-the-fly");
myWin.document.write("</title></head><body>");
myWin.document.write("<center>");
myWin.document.write("Bu HTML-hujjat ochildi");
myWin.document.write("JavaScript yordamida!");
myWin.document.write("</center>");
myWin.document.write("</body></html>");
```

Ko'rinib turibdiki, biz hujjatga HTML tilining oddiy teglarini yozamiz. Ya'ni amalda biz HTML bo'laklarini hosil qilamiz! Bunda HTML teglarining istalganidan foydalanish mumkin.

Buni yakunlagandan keyin biz hujjatni yana yopishimiz kerak. Bu quyidagi buyruq orqali amalga oshiriladi:

```
// hujjatni yopish - (oynani emas!)
myWin.document.close();
```

Nafaqat hujjatlarni dinamik ravishda yaratish mumkin, balki ularni o'z xohishicha u yoki bu freymlarga joylashtirish ham mumkin. Masalan, `frame1` va

---

*frame2* nomli ikki freymga ega bo'lsak va *frame2* ga yangi hujjatni hosil qilish kerak bo'lsa, u holda *frame1* ga quyidagini yozib qo'yish yetarli:

```
parent.frame2.document.open();
parent.frame2.document.write("Bu yerda sizning HTML-kod joylashgan");
parent.frame2.document.close();
```

*Ikkilamchi ob'yektlar.* JavaScriptda ichki qurilgan ob'yektlarni ko'rib chiqamiz. Foydalanuvchi tomonidan yaratilgan ob'yektlar va brauzerning ob'yekli modelini tashkil qilgan ob'yektlardan farq qilib, ichki qurilgan ob'yektlar xohlagan kontekstda – xoh u Microsoft Internet Explorer yoki Netscape Navigator bo'lsin chaqirilishi mumkin.

JavaScript tiliga bo'lgan asosiy talablarni belgilab beruvchi ECMAScript (ECMAScript Language Specification, Standard ECMA-262ni qarang) spetsifikatsiyasiga ko'ra tilda quyidagi ob'yektlar amalga oshirilgan bo'lishi kerak: Global, Object, Function, Array, String, Boolean, Number, Math va Date.

Array, Boolean, Date, Function, Math, Number va String ichki qurilgan ob'yektlarini ko'rib chiqamiz.

*Array ob'yekti*

JavaScript tilida massivlarni yaratish uchun ichki qurilgan ma'lumotlar tipi yo'q, shuning uchun bunday masalalarni hal qilishda Array ob'yektidan foydalaniladi. U massivlarni birlashtirish, tartiblash va o'rin almashtirish uchun metodlarga ega, yana massivning o'lchamini aniqlash imkoniyati ham mavjud.

*Massiv* – bu nomi va tartib raqami (indeksi) bo'yicha chaqiriladigan qiymatlarning tartiblashgan to'plamidir. Masalan, dasturda har biri o'z tartib raqamiga ega bo'lgan habarlar to'plami – allMsg dan tuzilgan massivni yaratish mumkin. Shunday qilib, allMsg[0] birinchi habar, allMsg[1] – ikkinchi habar bo'ladi.

Array ob'yektini hosil qilish uchun ikki bir-birini o'rnini olishi mumkin bo'lgan usulni qo'llash mumkin.

New konstruktorini chaqiring va massiv o'lchami (undagi elementlar soni)ni bering. Massivni to'ldirish keyinroq ro'y beradi.

Quyidagi misolni ko'rib chiqamiz.

```
<html>
<head><title> JavaScript sinovi.</title>
<script language="JavaScript">
// yangi massiv yaratish
allStr = new Array(5);
// massivni to'ldirish
allStr[0] = "Xabar №1";
allStr[1] = " Xabar №2";
allStr[2] = " Xabar №3";
allStr[3] = " Xabar №4";
allStr[4] = " Xabar №5";
// massiv elementlarini tasvirlash uchun funktsiya
function showMsg(ndx)
{
 alert(allStr[ndx]);
```

---

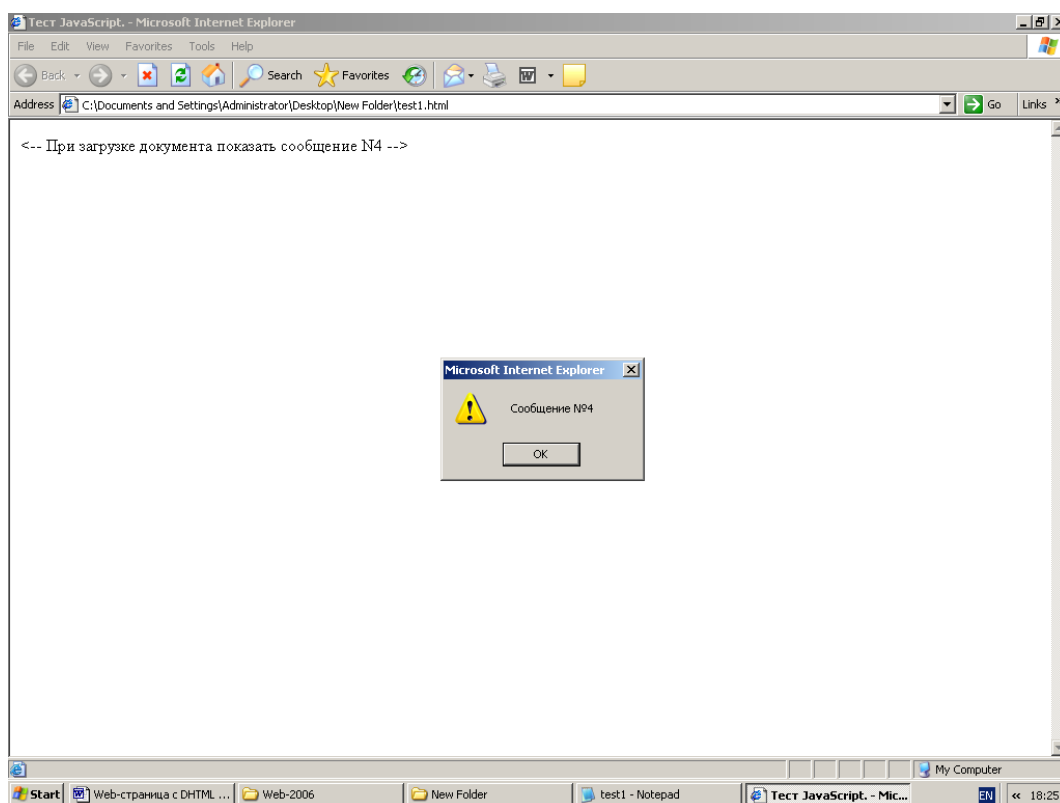
```
}
</script>
</head>
<!-- Hujjat yuklanishida №4 xabarni ko'rsatish -->
<body onLoad="showMsg(3);">
</body>
</html>
```

Yuqorida keltirilgan misolda 5 ta elementdan tashkil topgan massiv yaratiladi, keyin u to'ldiriladi.

New konstruktorini chaqirasiz va massivning barcha elementlarining qiymatlarini berasiz. Massiv o'lchami bu holda oshkora holda ko'rsatilmaydi.

Quyidagi misolni ko'ramiz.

```
<html>
<head><title> JavaScript sinovi.</title>
<script language="JavaScript">
// yangi massivni yaratish va uni to'ldirish
allStr = new Array("Xabar №1", "Xabar №2", "Xabar №3", "Xabar №4", "Xabar
№5");
// massiv elementlarini tasvirlash uchun funktsiya
function showMsg(ndx)
{
 alert(allStr[ndx]);
}
</script>
</head>
<!-- Hujjat yuklanishida №4 xabarni ko'rsatish -->
<body onLoad="showMsg(3);">
</body>
</html>
```



Bu yerda massiv elementlarining qiymatlari new konstruktorini chaqirish vaqtidayoq bevosita beriladi.

*Array ob'yektining metodlari*

Array ob'yekti quyidagi metodlarga ega.

Metod	Bayoni
join	Massivning barcha elementlarini satrga birlashtiradi
reverse	Massivdagi elementlar tartibini o'zgartiradi – birinchi element ohirgi elementga, ohirgisi esa birinchi elementga aylanadi
Sort	Massiv elementlarini tartiblaydi

Array ob'yekti metodlarining ishlatilishlarini ko'rib chiqamiz. Aytaylik bir nechta elementga ega bo'lgan massiv va ikkita – showAll va showElement funksiyalari berilgan bo'lsin. Birinchi, massivning barcha elementlarini namoyish qiluvchi funksiya sikl ichida massivning berilgan elementini namoyish qiluvchi showElement funksiyasini chaqiradi:

```
<html>
<head><title> JavaScript sinovi.</title>
<script language="JavaScript">
myArray = new Array("Ota", "Ona", "Aka", "Opa", "Boshqalar");
function showElement(ndx)
{
 alert(myArray[ndx]);
}
function showAll()
```



---

```

{
 for(i = 0; i <= myArray.length-1; i++)
 {
 showElement(i);
 }
}
</script>
</head>
<body onLoad="showAll();">
</body>
</html>

```

Agar yuqorida keltirilgan faylni yuklasak, har birida myArray massivining bitta elementi - Mother, Father, Sister, Brother va Uncle tasvirlangan habarlar paneli ketma-ketligini ko‘rishimiz mumkin.

Quyidagi kod yozilgan test funksiyasini yaratamiz:

```

function test()
{
 alert(myArray.join());
}

```

Va <body> tegini o‘zgartiramiz:

```

<body onLoad="test();">

```

Joint metodi massiv elementlarini ajratib turuvchini berish mumkin bo‘lgan majburiy bo‘lmagan parametrga ega. Ko‘rsatilmagan holda “,” belgisi ishlatiladi. Masalan

```

function test()
{
 alert(myArray.join(" _|_ "));
}

```

reverse metodi massive elementlarini o‘rin almashtirishlari uchun foydalaniladi. test funksiyasiga shu metodni chaqirilishini qo‘shamiz:

```

function test()
{
 myArray.reverse();
 alert(myArray.join(";"));
}

```

Massivning birinchi elementi ohirgi o‘rinni egalladi, ikkinchisi- ohiridan oldingi va hokazo.

Sort metodi massiv elementlarini tartiblash uchun foydalaniladi. test funksiyasiga shu metodni chaqirilishini qo‘shamiz:

```

function test()
{
 myArray.sort();
 alert(myArray.join(";"));
}

```

Ko‘p o‘lchamli massivlarni yaratish.

---

Array ob'yekti ko'p o'lchamli massivlarni yaratishga imkon beradi. Quyida ko'p o'lchamli massivni yaratishga misol keltirilgan.

```
<html>
<head><title> JavaScript sinovi. </title></head>
<body>
<center>
Ko'p o'lchamli massiv<p>
<script language="JavaScript">
 a = new Array(4);
 for(i=0; i < 4; i++)
 {
 a[i] = new Array(4);
 for(j=0; j < 4; j++)
 {
 a[i][j] = "["+i+", "+j+"]";
 }
 }
 for(i=0; i < 4; i++)
 {
 str = "Qator"+i+": ";
 for(j=0; j < 4; j++)
 {
 str += a[i][j];
 }
 document.write(str, "
");
 }
</script>
</center>
</body>
</html>
```

Yuqorida keltirilgan dasturda to'rtta elementdan tashkil topgan massiv hosil qilinadi: uning elementlarining har biri ham to'rtta elementdan iborat massivdir. Har bir elementga element indeksini beruvchi i, j juftlikning qiymatlari kiritiladi. Keyin sikl ichida berilgan massiv elementlari namoyish qilinadi.

#### *Boolean ob'yekti*

Ichki qurilgan Boolean ob'yekti mantiqiy bo'lmagan qiymatlarni mantiqiy qiymatlarga aylantirish uchun foydalaniladi. Boolean ob'yektining namunasini hosil qilish uchun new konstruktori ishlatiladi:

```
bfalse = new Boolean(false);
```

```
btrue = new Boolean(true);
```

valueOf metodi mantiqiy o'zgaruvchining mantiqiy ko'rinishdagi qiymatini ko'rsatadi, toString metodi esa satri ko'rinishdagi qiymatini ko'rsatadi. Bu metodlarni ishlatilishiga misol quyida keltirilgan.

```
<html>
<head><title> JavaScript sinovi.</title></head>
```

---

```
<body>
<script language="JavaScript">
 // ikki mantiqiy o'zgaruvchi yaratamiz
 bfalse = new Boolean(false);
 btrue = new Boolean(true);
 // ularning qiymatlarini chiqaramiz (mantiqiy qiymatlar)
 document.write(bfalse.valueOf()+"
");
 document.write(btrue.valueOf()+"
");
 // satrli qiymatlarni chiqaramiz
 document.write(bfalse.toString()+"
");
 document.write(btrue.toString()+"
");
</script>
</body>
</html>
```

*Date ob'yekti.*

Date ob'yekti va uning metodlaridan skript dasturlarda sana va vaqt bilan ishlashda foydalaniladi. Bu ob'yekt sanani o'rnatish, uning qiymatini olish va turli hil almashtirishlarni bajarish uchun katta sondagi metodlar to'plamiga ega. Date ob'yekti hossalarga ega emas.

Shuni ta'kidlab o'tamizki, JavaScriptda sana huddi Javadagi singari saqlanadi – u 1 yanvar 1970 yildan buyon o'tgan millisekundlar sonini ko'rsatadi. Shunday qilib undan oldingi sanalar qo'llanmaydi.

Date ob'yektining namunasini hosil qilish uchun Date konstruktoridan foydalaniladi:

```
MyDate = new Date([parametrlar]);
```

Quyidagi parametrlarni ko'rsatish mumkin:

- parametrlarsiz – namuna joriy sana va vaqtni ko'rsatadi. Masalan, today = new Date();

- sanani quyidagi formatda ko'rsatuvchi satr: "Oy, kun, yil vaqt:minutlar:sekundlar". Masalan, someDate = new Date("May 15, 1996"). Agar soat, minut yoki sekundlar soni ko'rsatilmagan bo'lsa, ularning qiymatlari 0 ga teng deb olinadi;

- yil, oy va kunlarning butun sondagi qiymatlari to'plami. Masalan, otherDay = new Date( 96, 4, 15);

- yil, oy, kun, soat, minut va sekundlarning butun sondagi qiymatlari to'plami. Masalan, sameDay = new Date( 96, 4, 15, 15, 30, 0);

Turli hil parametrlarning ishlatilishiga oid misol quyida keltirilgan.

```
<html>
<head><title> JavaScript test.</title></head>
<body>
<center>
<script language="JavaScript">
 today = new Date();
 document.write("today="+today+"
");
 someDate = new Date("May 16, 1996");
```

```

document.write("someDate="+someDate+"
");
otherDay = new Date(96, 4, 15);
document.write("otherDay="+otherDay+"
");
sameDay = new Date(96, 4, 16, 15, 30, 0);
document.write("sameDay="+sameDay+"
");
</script>
</center>
</body>
</html>

```

### *Date ob'yektining metodlari*

Date ob'yekti tomonidan sana va vaqtni boshqarish uchun ishlatiladigan metodlarni quyidagi kategoriyalarga bo'lish mumkin:

- *o'rnatish metodlari (set)* – Date ob'yekti namunalarida sana va vaqtni o'rnatish uchun metodlar;
- *aniqlash metodlari (get)* - Date ob'yekti namunalaridan sana va vaqtni olish uchun metodlar
- *shakl almashtirish metodlari (to)* – sana va vaqtni satrga aylantirish uchun metodlar;
- *sanani qayta ishlash uchun metodlar.*

O'rnatish va aniqlash metodlari sekund, minut, soat, oy kuni, hafta kuni, oy va yil qiymatlarini olish/o'zgartirish uchun foydalanilishi mumkin. Masalan, hafta kunini aniqlash mumkin bo'lgan `getDay` metodi mavjud, lekin `setDay` metodi mavjud emas, chunki hafta kuni avtomatik ravishda o'rnatiladi.

Bu metodlarning barchasi quyidagi jadvalda keltirilgan butun sonli qiymatlardan foydalanadi.

Qiymati	Oralig'i
Minut va sekundlar soni	0..59
Soatlar soni	0..23
Hafta kuni	0..6
Sana	1..31
Oy	0..11 (Yanvar..Dekabr)
Yil	1900 yildan boshlab

Quyidagi misolni ko'ramiz. Aytaylik, siz sanani 15 may 1996 yil deb berdingiz. Quyida Date ob'yektining bir nechta metodlarining ishlatilishi namoyish qilingan.

```

<html>
<head><title>JavaScript </title></head>
<body>
<center>
<p>
<script language="JavaScript">
someDate = new Date("May 15, 1996");
document.write("someDate="+someDate+"
");

```

---

```
document.write("getDay =" + someDate.getDay() + "
");
document.write("getMonth =" + someDate.getMonth() + "
");
document.write("getYear =" + someDate.getYear() + "
");
</script>
</center>
</body>
</html>
```

getTime va setTime metodlari sanalarni taqqoslash uchun qulay. getTime metodi millisekundlar sonini beradi. Masalan, bu yildagi qolgan kunlar sonini qanday aniqlash quyida keltirilgan.

```
<html>
<head><title>JavaScript </title></head>
<body>
<center>

<script language="JavaScript">
today = new Date();
// sanani berish
endYear = new Date("December 31, 1990");
// yilni o'zgartirish
endYear.setYear(today.getYear());
// kundagi millisekundlar sonini hisoblash
msPerDay = 24 * 60 * 60 * 1000;
// kunlar sonini olish
daysLeft = (endYear.getTime() - today.getTime()) / msPerDay;
// yahlitlash
daysLeft = Math.round(daysLeft);
// ko'rsatish
document.write("Number of days left in the year: " + daysLeft);
</script>
</center>
</body>
</html>
```

Parse metodi vaqtning satrli tasvirini shakl almashtirish uchun foydalanilishi mumkin. Masalan,

```
someDate = new Date();
someDate.setTime(Date.parse("May 15, 1996"));
```

Date ob'yektini ko'rishni kichkina bir amaliy misol bilan yakunlaymiz. Biz joriy vaqtni ekranga chiqaruvchi dasturni yaratamiz – xohlasangiz uni siz o'z HTML-sahifangizga joylashtirishingiz mumkin. Vaqt “tugma” interfeys elementining sarlavhasi sifatida chiqariladi. Mashq sifatida dasturni shunday to'ldirishni taklif qilamanki, tugmani bosishingiz bilan joriy sana namoyish qilinsin.

Vaqtning namoyish qiluvchi dastur matni quyida keltirilgan.

```
<html>
<head><title>JavaScript </title>
```

---

```

<script language="JavaScript">
function showTime()
{
 // joriy vaqtni olamiz
 var time = new Date();
 // soatlar sonini bilib olamiz
 var hour = time.getHours();
 // minutlar sonini bilib olamiz
 var minute = time.getMinutes();
 // sekundlar sonini bilib olamiz
 var second = time.getSeconds();
 // soatlarni ko'rsatamiz
 var temp = hour;
 // minutlarni ko'rsatamiz (0 ni 0-9ga qo'shib)
 temp += ((minute < 10) ? ":0" : ":") + minute;
 // sekundlarni ko'rsatamiz (0 ni 0-9ga qo'shib)
 temp += ((second < 10) ? ":0" : ":") + second;
 // namoyish qilamiz
 document.forms[0].clock.value = temp;
 // har bir sekundni ko'rsatamiz
 id = setTimeout("showTime()", 1000);
}
</script>
</head>
<body onLoad="showTime();">
<center>

JavaScript Date & Time Demo

<form name="DateDemo">
 <input type="button" name="clock" value="yyyyyyyyy">
</form>
</center>
</body>
</html>

```

Sanani namoyish qiladigan funktsiya quyidagi ko'rinishda bo'lishi mumkin:

```

function showDate()
{
 //joriy sanani olamiz
 var D = new Date();
 // hafta kunini bilib olamiz
 var DOW = D.getDay();
 // sanani bilib olamiz
 var Day = D.getDate();
 // oyni bilib olamiz

```

---

```

var Month = D.getMonth();
// yilni bilib olamiz
var Year = D.getYear();
// namoyish qilamiz
temp = Day + "/" + Month + "/" + Year;
document.forms[0].clock.value = temp;
}

```

DOW o'zgaruvchisida hafta kunining nomeri saqlanadi. Yuqorida ko'rib o'tilgan Array ob'yektidan foydalangan holda, hafta kunlari nomlarining massivini hosil qilishimiz mumkin:

```

dayNames = new
 Array("yakshanba", "dushanba", "seshanba", "chorshanba", "payshanba",
"juma", "shanba");

```

va oxiridan oldingi qatordagi showDate funksiyasiga quyidagilarni qo'shing:

```
temp = Day + "/" + Month + "/" + Year + dayNames[DOW];
```

*Function ob'yekti*

Ichki qurilgan Function ob'yekti satrga funktsiya deb baholash mumkin bo'lgan JavaScriptdagi kodni berishga imkon beradi. Function ob'yektining namunasini yaratish uchun new konstruktoridan foydalaniladi:

```
var newBgColor = new Function("c", "document.bgColor=c");
```

Bu funktsiyadan foydalanishga misol quyida berilgan.

```

<html>
<head><title>JavaScript </title></head>
<body>
<center>
<script language="JavaScript">
 var newBgColor = new Function("c", "document.bgColor=c");
 function fnDemo()
 {
 newBgColor("#a7b7c7");
 }
</script>
<form>
<input type="button" value="bgColor" onClick="fnDemo();">
</form>
</center>
</body>
</html>

```

Yana bir misolni ko'rib chiqamiz. Aytaylik, bizga ikki argumentning qiymatlarini ko'paytiradigan funktsiya kerak bo'lsin. Bu funktsiyani myMult deb ataymiz:

```
var myMult = new Function("x", "y", "return x*y");
```

Undan quyidagicha foydalanish mumkin:

```

<html>
<head><title>JavaScript </title></head>

```

---

```

<body>
<center>
<script language="JavaScript">
 var myMult = new Function("x", "y", "return x*y");
 document.write("10*20="+myMult(10, 20));
</script>
</center>
</body>
</html>

```

Biz hattoki kichkina forma hosil qilishimiz va uni turli hil sonlarning ko‘paytmasini topish uchun foydalanishimiz mumkin:

```

<html>
<head><title>JavaScript 12.97</title></head>
<body>
<center>
<script language="JavaScript">
 var myMult = new Function("x", "y", "return x*y");
 function calc()
 {
 document.forms[0].sum.value =
 myMult(document.forms[0].x.value, document.forms[0].y.value);
 }
</script>
</center>
<form>
<input type="text" name="x" value=10 size=4>
<input type="text" name="y" value=10 size=4>
<input type="text" name="mul" value=" " size=4>
<input type="button" value="calc" onClick="calc();">
</form>
</body>
</html>

```

Math ob’yekti

Ichki qurilgan Math ob’yekti turli hil matematik konstantalarni olish va matematik funktsiyalarni bajarish uchun xossa va metodlarga ega. Quyida shu qiymatlarni ekranga chiqaradigan dasturning matni berilgan.

```

<html>
<head><title>JavaScript </title></head>
<body bgcolor="#a7b7c7">
<center>Math Object Properties<p>
<script language="JavaScript">
 s = "<TABLE BORDER=2>" +
"<TR><TD>E</TD><TD>" + Math.E + "</TD></TR>" +
"<TR><TD>LN2</TD><TD>" + Math.LN2 + "</TD></TR>" +
"<TR><TD>LN10</TD><TD>" + Math.LN10 + "</TD></TR>" +

```



```

"<TR><TD>LOG2E</TD><TD>" + Math.LOG2E + "</TD></TR>" +
"<TR><TD>LOG10E</TD><TD>" + Math.LOG10E + "</TD></TR>" +
"<TR><TD>PI</TD><TD>" + Math.PI + "</TD></TR>" +
"<TR><TD>SQRT1_2</TD><TD>" + Math.SQRT1_2 + "</TD></TR>" +
"<TR><TD>SQRT2</TD><TD>" + Math.SQRT2 + "</TD></TR>" +
"</TABLE>";
document.write(s);
</script>
</center>
</body>
</html>

```

Keyingi jadvalda Math ob'yektida mavjud bo'lgan metodlar sanab o'tilgan.

Metod	Bayoni
abs	Argument qiymatining modulini beradi
sin, cos, tan	Standart trigonometrik funktsiyalar, argumentlar radianda beriladi
acos, asin, atan	Teskari trigonometrik funktsiyalar, radiandagi qiymatlarni beradi
exp, log	Eksponenta va natural logarifm, asos -e
ceil	Argumentga teng yoki undan katta bo'lgan butun sonni beradi
floor	Argumentga teng yoki undan kichik butun sonni beradi
min, max	Ikki argumentlardan kichigi yoki kattasini beradi
pow	Argument darajasini beradi
round	Argumentni unga eng yaqin butun songacha yaxlitlaydi
sqrt	Argumentning kvadrat ildizini beradi

Math ob'yektining ko'plab sondagi hossa va metodlaridan foydalanganda, masalan, qandaydir hisoblashlarni bajarganda, with operatoridan foydalanish qulay:

```

with(Math)
{
 a = PI * r*r;
 b = floor(c);
 x = r * sin(theta);
 y = r * cos(theta);
 z = round(b);
}

```

*Number ob'yekti*

Ichki qurilgan Number ob'yektining hossalari maksimal qiymat, "not-a-number" (NaN) va cheksizliklar kabi tipdagi turli hil sonli o'zgarmlarning qiymatlarini olish uchun foydalaniladi. Quyida Number ob'yekti hossalari qiymatlarini chiqaradigan dasturning matni berilgan.

```

<html>
<head><title>JavaScript </title></head>
<body bgcolor="#a7b7c7">
<center>Number Object Properties<p>
<script language="JavaScript">

```

```

s = "<TABLE BORDER=2>" +
 "<TR><TD>MAX_VALUE</TD><TD>" +
 Number.MAX_VALUE + "</TD></TR>" +
 "<TR><TD>MIN_VALUE</TD><TD>" +
 Number.MIN_VALUE + "</TD></TR>" +
 "<TR><TD>NaN</TD><TD>" +
 Number.NaN + "</TD></TR>" +
 "<TR><TD>NEGATIVE_INFINITY</TD><TD>" +
 Number.NEGATIVE_INFINITY + "</TD></TR>" +
 "<TR><TD>POSITIVE_INFINITY</TD><TD>" +
 Number.POSITIVE_INFINITY + "</TD></TR>" +
 "</TABLE>";
document.write(s);
</script>
</center>
</body>
</html>

```

### String ob'yehti

JavaScript tilida string ko'rinishidagi ichki qurilgan ma'lumotlar tipi yo'q. Shunga qaramasdan String ob'yehti va uning metodlaridan foydalangan holda siz skript dasturlarda satrlar bilan ishlashingiz mumkin. String ob'yehti satrlar ustida bajariladigan juda ko'plab metodlarga va satrning uzunligini aniqlashga imkon beruvchi bitta hossa (length)ga ega.

String ob'yektning namunasi new konstruktorini chaqirish orqali hosil qilinadi:

```
myString = new String("myString Object");
```

String ob'yehti ikki hil tipdagi metodlarni namoyish qiladi. Birinchi tipga shakl almashtirishlar yoki satrlar ustida boshqa operatsiyalarni bajaradigan metodlar kiradi: substring, toUpperCase va h. Shuni ta'kidlash joizki, HTML-versiyasi satrini beruvchi funktsiyalar ECMAScript standartiga kirmaydi.

String ob'yehti metodlari quyidagi jadvalda sanab o'tilgan.

Metod	Bayoni
anchor	anchor- HTML-elementini yaratadi
big, blink, bold, fixed, italics, small, strike, sub, sup	Mos HTML-elementni yaratadi
charAt	Satrning ko'rsatilgan o'rnida joylashgan belgini beradi
indexOf, lastIndexOf	Mos ravishda satrdagi qism-satrning o'rnini yoki satrdagi ohirgi qism-satrning o'rnini beradi
link	HTML-o'tishni yaratadi
split	String ob'yektini satrlarni qism-satrlarga aylantirgan holda satrlar massiviga bo'ladi
substring	Ko'rsatilgan qism-satrni beradi

---

toLowerCase, toUpperCase	Satrdagi belgilarni mos ravishda quyi va yuqori registrdagi belgilarga aylantiradi
-----------------------------	---------------------------------------------------------------------------------------

Bu ob'jekt metodlaridan foydalanishga bir nechta misollar ko'rib o'tamiz.

```
<html>
<head><title>JavaScript </title></head>
<body bgcolor="#a7b7c7">
<center>
<script language="JavaScript">
 //yangi satrni yaratamiz
 var s = new String('Netscape Navigator');
 //qism-satrni olamiz
 var n = s.substring(0,8);
 //satr va qism satrni ko'rsatamiz
 document.write("string="+s+"
substring="+n);
 //quyi registrga aylantiramiz
 var l = s.toLowerCase(s);
 // ko'rsatamiz
 document.write("
"+l);
 // yuqori registrga aylantiramiz va ko'rsatamiz
 document.write("
"+s.toUpperCase(s));
</script>
</center>
</body>
</html>
```

Bular shakl almashtirish metodlari edi. Endi HTML-versiyalar satrlarini beruvchi metodlarni ko'rib chiqamiz:

```
<html>
<head><title>JavaScript </title></head>
<body bgcolor="#a7b7c7">
<center>
<script language="JavaScript">
 // yangi satr yaratamiz
 var s = new String('Netscape Navigator');
 // va larni qo'shamiz
 document.write(s.bold(s)+"
");
 // <i> va </i> larni qo'shamiz
 document.write(s.italics(s)+"
");
 // <tt> va </tt> larni qo'shamiz
 document.write(s.fixed(s));
 // _{va} larni qo'shamiz
 document.write(s.sub(s)+" ");
 // ^{va} larni qo'shamiz
 document.write(s.sup(s)+"
");
 // <strike> va </strike> larni qo'shamiz
```

---

```
document.write(s.strike(s)+"
");
</script>
</center>
</body>
</html>
```

Satrni o‘tishga aylantirish uchun link metodini chaqirish kerak:  
var s = new String('Netscape Navigator');  
document.write(s.link("http://www.netscape.com"));  
s o‘zgaruvchi uchun hujjatda HTML-matn quyidagicha ko‘rinishga ega bo‘ladi:  
<A HREF="http://www.netscape.com">Netscape Navigator</A>



### Savol va topshiriqlar:

1. Ob'yekt metodlari nima?
2. Hodisalar nima?
3. String ob'yekti nima?
4. Math ob'yekti nima?
5. Number ob'yekti nima?



## 4.5. PHP tili

PHP – bu serverda qayta ishlanuvchi senariylar tilidir. ASP kabi PHP kodlar ham bevosita HTML-hujjatni tarkibiga qo‘shiladi. Ushbu tilning nomi Personal Home Page Tools so‘zlarining qisqartmasidan olingan. PHPda C va Perl tillarida uchragan bir qator muammolar hal etilgan. Bundan tashqari, PHP ma’lumotlar bazasi bilan ishlash uchun juda qulay vositadir. Umuman olganda Perl, PHP – ochiq tizimli tillar hisoblanadi va ularni dasturchilar modernizatsiyalashtira oladi.

Ko‘pgina boshqa dasturlash tillaridan farqli ravishda, PHP qandaydir tashkilot yoki kuchli dasturchi tomonidan yaratilgan emas. Uni oddiy foydalanuvchi Rasmus Lerdorf 1994 yili o‘zining bosh sahifasini interfaol uslubda ko‘rsatish uchun yaratgan. Unga Personal Home Page (PHP – shaxsiy bosh sahifa) deb nom bergan.

1995 yili Rasmus PHPni o‘zining HTML formalari bilan ishlaydigan boshqa dastur bilan umumlashtirib PHP/FI Version 2 («Form Interpretator») hosil qildi. 1997 yilga borib PHPdan foydalanuvchi saytlar 50 mingdan oshdi. Shundan so‘ng web texnologiya ustalari PHP g‘oyasi asosida mukammal til yaratishga Ziva Suraski va Endi Gutmans asoschiligida kirishildi. PHPni samarali deb hisoblanmagani uchun deyarli noldan boshlab, mavjud C va Perl tillaridan ibrat olib PHP3 talqinini yaratildi. 1999 yilga kelib PHP asosida qurilgan saytlar milliondan oshib ketdi. 2000 yilda esa Zend Technologies shirkati yangi ko‘pgina funktsiyalarni qo‘shgan holda PHP4 sharhlovchisini yaratdi.

PHP – web texnologiya tili. PHPni o‘rganish uchun avval HTML va dasturlash tilidan habardor bo‘lish talab qilinadi. HTML/CSS va JavaScript- larni mukammal

---

biladiganlar uchun PHPni o'rganish murakkablik tug'dirmaydi. PHPning vazifasi HTML faylini yaratib berish. JavaScript yordamida bajariladigan ko'pgina operatsiyalarni PHP orqali ham amalga oshirish mumkin.

JavaScript – mijoz tomonida bajariladi, PHP esa serverda. PHPda yozilgan kod serverning o'zida bajarilib, klientga HTML shaklida etib boradi. Bu havfsizlik jihatdan ancha maqsadga muvofiq.

JavaScript yordamida kod yozish, ma'lumot uzatish va qabul qilishni biroz tezlashtirsa-da, kodni mijoz ko'rish imkoniyatiga ega bo'ladi.

«PHPda har qanday dasturni bajarsa bo'ladi» – degan edi uning yaratuvchisi. Birinchi navbatda PHP tili server tomonidan bajariladigan skriptlar yaratish uchun foydalaniladi va aynan shuning uchun u yaratilgan. PHP tili ixtiyoriy CGI-skriptlari masalalarini echishga va bundan tashqari html formali ma'lumotlarni qayta ishlashga hamda dinamik ravishda html sahifalarni ishlab chiqishga qodir. Biroq PHP tili foydalaniladigan boshqa sohalari ham mavjud. Bu sohalarni biz uchta asosiy qismga bo'lamiz:

Birinchi soha – biz yuqorida aytib o'tganimizdek, server tomonidan bajariladigan ilovalar (skriptlar) yaratish. PHP tili bunday turdagi skriptlarni yaratish uchun juda keng qo'llaniladi. Bunday ish ko'rsatish uchun PHP-skriptlarni qayta ishlovchi va skriptlarni qayta ishlovchi web-server, skriptlarni natijasini ko'rish uchun brauzer va albatta php-kodini yozish uchun qanday bo'lsa ham matn muharriri kerak bo'ladi. PHP- skriptlarni qayta ishlovchi CGI-dasturlar ko'rinishida yoki server modullari ko'rinishida tarqalgan.

Ikkinchi soha – buyruqlar satrida bajariladigan skriptlarni yaratish. Ya'ni PHP tili yordamida biror-bir kompyuterda brauzer va web-serverlardan mustaqil ravishda o'zi bajariladigan skriptlarni ham yaratish mumkin. Bu ishlarni bajarish uchun hech bo'lmaganda PHP-skriptlarni qayta ishlovchi (bu holatda biz uni buyruqlar satri interpretatori (CLI, command line interpreter) deb ataymiz) talab etiladi. Bunday ishlash uslubi turli masalalarni rejalashtirish yordamida bajarilishi uchun kerak bo'lgan skriptlar yoki oddiy matnni qayta ishlash uchun kerak bo'lgan masalaga o'xshash ishlaydi. Va nihoyat oxirgi uchinchi soha – bu mijoz tomonidan bajariladigan GUI- ilovalarni (grafik interfeys) yaratish. Bu soha PHP tilini endigina o'rganayotgan foydalanuvchilar uchun uncha muhim bo'lmagan sohadir.

PHP tilini qo'llanilish sohalari keng va turlichadir. Yuqoridagi masalalarni echa oladigan boshqa turlicha dasturlash tillari ham mavjud. Unda nima uchun PHP tilini o'rganishimiz kerak? U til bizga nima beradi? Birinchidan, PHP tili o'rganish uchun juda qulay. PHP tilini sintaksisi asosiy qoidalari va ishlash printsiplari bilan yetarlicha tanishib chiqib o'zingizni shaxsiy dasturingizni tuzib ko'rib, so'ngra uni boshqa dasturlash tillarida tuzilgan variantlari bilan solishtirsangiz bunga guvoh bo'lasiz. Ikkinchidan, PHP tili barcha bizga ma'lum platformalarda, barcha operatsion tizimlarda hamda turlicha serverlarda erkin ishlay oladi.

PHP dasturlash tilini turli dasturlash tillari o'rtasidagi o'zaro aloqasiga diqqatni qaratsak, bunga Java dasturlash tilini aytib o'tish kerakki, Java dasturlash tili ob'yektlarini PHP tili o'z ob'yektlari sifatida qaraydi. Ob'yektlarga murojaat sifatida CORBA kengaytmasidan foydalaniladi.



## Savol va topshiriqlar:

1. PHP nima?
2. PHP qaerda va kim tomonidan ishlab chiqilgan?

### PHP asosiy tuzulishi

Ko‘p hollarda PHP tilini interpretatori ishlayotganligini tekshirib ko‘rish uchun tuziladigan dastur eng sodda dastur deb ataladi. Quyida PHP dasturlash tilining maxsus kodli teglari yordamida tuzilgan sodda html-fayliga misol keltirilgan:

```
<html>
<head>
<title> Misol </title>
</head>
<body>
<?php
echo "<p> Salom, bu men – PHP skript! </p>";
?>
</body>
</html>
```

Yuqorida aytib o‘tganimizdek, PHP dasturlash tili C va Perl dasturlash tiliga o‘xshash. Biroq keltirilgan dastur C va Perl dasturlash tilidagi dasturdan ancha katta farq qiladi. Bunda HTML sahifaga chiqarish uchun bir qator maxsus buyruqlarni yozish shart emas. Bevosita PHP-kod asosida qurilgan biror vazifani bajaradigan HTML- skript yoziladi (bizni misolda ekranda chiqarilgan matn). PHP dasturlash tilining C va Perl dasturlash tillaridan kamchiligi shuki, murakkab skriptlarni PHP dasturlash tili ancha sekin bajaradi.

PHP-skriptlar – bu serverda bajariladigan va qayta ishlanadigan dasturlardir. Bu skriptlarni JavaScript tipidagi skriptlar bilan taqqoslash mumkin emas, chunki JavaScript tilidagi skriptlarda yozilgan buyruqlar faqat mijoz kompyuteridagina bajariladi. Klient kompyuterida va server kompyuterida bajariladigan skriptlarning farqi nimada? Agarda skript serverda qayta ishlansa, mijoz kompyuteriga faqatgina natija yuboriladi. Masalan, agarda serverda skript bajarilayotgan bo‘lsa, yuqorida keltirilganga o‘xshab mijoz HTML-sahifa ko‘rinishdagi natijani oladi:

```
<html>
<head>
<title> Misol </title>
</head>
<body>
<p> Salom, bu men – PHP skript! </p>
</body>
</html>
```

Bu holatda mijoz qanday kod bajarilayotganini bilmaydi. O‘z serveringizni HTML-fayllarni PHP muharriri qayta ishlaydigan qilib sozlab olishingiz ham mumkin. Ya’ni klientlar oddiy HTML-faylni qabul qildimi yoki skript natijasini

---

koʻrdimi buni bila olmaydi. Agarda skript klient kompyuterida qayta ishlansa (masalan, JavaScript tilidagi dastur), u holda klient skript kodidan iborat HTML-sahifani koʻradi.



### Savol va topshiriqlar:

1. PHP asosiy tuzilishi nimadan iborat?
2. PHPning boshqa dasturlash tillari bilan munosabatlari?

#### *PHP dasturlari*

PHP dasturlari ikki usulda bajarilishi mumkin: Web-server tomonidan senariy ilovasi va konsol dasturi sifatida.

Bizning maqsadimiz web-ilovalarni dasturlash boʻlgani uchun asosan birinchi usulni koʻramiz.

PHP odatda Internet bilan bogʻliq dasturlar yaratish uchun ishlatiladi. Lekin PHP dan buyruq satrlar interpretatori, asosan \*nix tizimlarda foydalanish mumkin. Oxirgisi CORBA va COM interfeyslar hamda PHP-GTK kengaytmasi yordamida mumkin. Bu xolda quyidagi masalalarni echish mumkin:

- Interaktiv komanda qatorlari yordamida ilovalar yaratish;
- Kross-platformali GUI ilovalarni PHP-GTK kutubxonasi yordamida yaratish;
- Windows va Linux uchun baʼzi masalalarni avtomatizatsiya qilish

Serverga brouzerning murojlat qilishida yordamida php-stsenariylari bajarilishini koʻrib chiqamiz. Avval brauzer .php kengaytmali sahifani soʻraydi, soʻngra web-server dasturni PHP mashinadan oʻtkazadi va natijani html-kod shaklida qaytaradi. Agar standart HTML sahifani olib, kengaytmasini .phpga oʻzgartirilsa va PHP mashinadan oʻtkazilsa, foydalanuvchiga oʻzgartirmasdan qaytaradi. Bu faylga PHP komandani qoʻshish uchun, PHP komandalarni maxsus teglar ichiga olish kerak. Bu teglarning 4 xil shakli mavjud boʻlib, ixtiyoriysidan foydalanish mumkin:

1. **XML qayta ishlash instruktsiyasi:**

2. `<?php`

3. ...

`?>`

4. **SGML qayta ishlash instruktsiyasi:**

5. `<?`

6. ...

`?>`

7. **HTML stsenariylari qayta ishlash instruktsiyasi:**

8. `<script language = "php">`

9. ...

10. `</script>`

11. **ASP uslubidagi instruktsiya:**

12. `<%`

13. ...

14. `%>`

---

Biz XML yoki SGML uslubiga amal qilamiz.

Xususan biror blok ichida PHPdan chiqish mumkin, faqat keyinchalik yana uning ichiga kirib kodni tugatish sharti bilan, quyidagi konstruktsiya mumkin:

```
<?
 if(5<3){
 echo("<p>Hello, world!<p>");
 }
 ?>
 <p>Hello!</p>
 // bu qator PHP kodi sifatida qaralmaydi
 // va kod bloki bajarilayotgan bo'lsa chiqariladi
<?
 echo("<p>Hello, world!<p>");
}
?>
```

PHPda echo komandasi web-sahifalarda uchraydigan har qanday ma'lumotni (matn, HTML ajratuvchi simvoli, son) chiqarish uchun qo'llanadi. Uning ma'nosi misolda lo'rsatilgan.

### ***Izohlar***

PHP tilida izohlarni joylash uchun bir necha usullar mavjud. Eng soddasi ikkilik slash (//) dan foydalanish, shundan so'ng PHP satrlar oxirigacha yozilganni o'tkazib yuboradi. Bundan tashqari C (/\*...\*/) uslubidagi ko'p qatorli izohlardan foydalanish mumkin. Bir qatorli izohlar uchun (#) simvoldan foydalanish qulay (UNIX script tillaridagi izoh).

```
<php
 echo("<p>Hello</p>"); // izoh
 echo("<p>Hello</p>"); # izoh
 /*
 bu ham izoh
 */
?>
```

Shuni esdan chiqarmaslik lozimki PHP uslubi izohlari faqat PHP cheklanishlari orasida ta'sir qiladi. Agar PHP bu izohlar simvollarini cheklanishlari tashqarisida uchratsa, ularni boshqa matnga o'xshab, html- sahifaga joylashtiradi.

Masalan:

```
<php
 echo("<p>Hello</p>"); // normal izoh
?>
// bu izoh brauzerda ko'rinadi.
<!-- HTML izohi.
```

Bu izoh HTML kodda ko'rinadi, brauzerda emas -->

Izohlarni faqat operator ohiriga emas, quyidagicha joylash ham mumkin:

```
<?
 $a = "Hello, world";
 echo strstr($a,"H");
 // bu funktsiyani keyinchalik qarab chiqamiz
```



?>



## Savol va topshiriqlar:

1. PHPda dasturlar necha usulda bajariladi?
2. PHPda izohlar nima va ular qanday joylanadi?



## 4.6. PHP tilida o‘zgaruvchilar va konstantalar

O‘zgaruvchilar haqida siz maktab kurslaridan tushunchaga egasiz. O‘zgaruvchi bu o‘zida ma’lumotni, ya’ni biror bir so‘zni, biror bir sonni saqlovchi belgi.

O‘zgaruvchilar dastur bajarilayotgan vaqtda, ustida amallar bajarish mumkin bo‘lgan ma’lumotlarni o‘zida jamlagan xotira maydoni nomini ifodalaydi. PHP da o‘zgaruvchilar \$(dollar) belgisi va o‘zgaruvchi nomini yozish orqali yaratiladi. O‘zgaruvchi nomi biror bir harf yoki so‘z bilan ifodalanadi, lekin son bilan ifodalanishi mumkin emas. **Masalan:** \$name, \$n, \$a2 va h.

O‘zgaruvchiga biror qiymat berish uchun uni o‘sha qiymatga tenglashtirish lozim. Tenglashtirish “=” amali yordamida amalga oshiriladi. Bir narsani yoddan chiqarmang, PHP da “=” amali aynan teng ma’nosini emas, balki tenglashtirish ma’nosini bildiradi. Agar bu ikki tushunchani aralashtirilib yuborilsa, dastur kodida xatoliklar yuzaga keladi.

Endi o‘zgaruvchini biror bir qiymatga tenglashtiramiz. **Masalan,** belgili qiymatga, u holda belgili qiymat “” ichida yozilishi kerak. Va har bir satr “;” belgisi bilan yakunlanishi kerak.

```
<?
$name="Lola";
?>
```

Agar o‘zgaruvchiga son qiymat beradigan bo‘lsak, u holda “” belgisi qo‘yilmasa ham bo‘ladi.

```
<?
$name="Lola";
$age=22;
?>
```

### 4.1 - misol. O‘zgaruvchiga qiymat berish

```
<?php
$first = ' Matn '; // $first ga
// ' Matn ' qiymatini tenglaymiz
$second = $first; // $second ga
// $first o`zgaruvchisi qiymatini tenglaymiz
$first = ' Yangi matn '; // $first ning qiymatini
// ' Yangi matn ' ga almashtiramiz
echo " first nomli o`zgaruvchi " .
"$first qiymatiga teng
";
// $first ning qiymatini chiqazamiz
echo " second nomli ozgaruvchi" .
" $second qiymatiga teng";
// $second ning qiymatini chiqazamiz
?>
```

**Natija:**

---

first nomli o`zgaruvchi Yangi matn qiymatiga teng  
second nomli o`zgaruvchi Matn qiymatiga teng

O`zgaruvchi tipi o`zida saqlanuvchi ma'lumotlarni tipiga bog`liq. PHP quyidagi ma'lumot tiplarini qo`llaydi:

**Skalyar tiplar:**

- boolean ( mantiqiy)
- integer, int(butun)
- float, real, double (haqiqiy)
- string (satri)

**Aralash tiplar:**

- array(massiv)
- object(ob'yekt)

**Maxsus tiplar:**

- resource(resurs)
- NULL

***Skalyar tipdagi o`zgaruvchilar***

Skalyar tipdagi o`zgaruvchilarni ma'lumotlarni saqlovchi bir quti desak ham bo`ladi. U o`ziga sonli, belgili va matnli qiymatlarni qabul qiladi. Skalyar tipdagi o`zgaruvchilar haqida yanada aniqroq tasavvur hosil qilish uchun quyidagi sxemani ko`rib chiqamiz.

**\$a=** 12

**\$b=** **Bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim, bilim**

***Boolean (bulev yoki mantiqiy) tipi***

Bu oddiy tip qiymatni rostligini aniqlaydi, bu tipdagi o`zgaruvchi faqatgina ikkita qiymat qabul qiladi- rost (TRUE) yoki yolg'on (FALSE).

Mantiqiy tipni aniqlash uchun, ikkita kalit so`z TRUE yoki FALSE ishlatiladi.

**4.2 – misol. Mantiqiy tip**

```
<?php
$test = True;
?>
```

***integer yoki int (butun) tipi***

Integer (butun) tipi butun sonlar to`plamida ( $Z=\{\dots, -2,-1,0,1,2,\dots\}$ ) gi qiymatlar uchun ishlatiladi. Butun tiplar o`nlik, o`n oltilik va sakkizlik sanoq sistemalaridagi qiymatlarni «-» yoki «+» belgilarini ishlatgan holda ham qabul qiladi.

Agar sakkizlik sanoq sistemasi ishlatilgan bo`lsa, raqamdan avval 0(nol) sonini, o`n oltilik sanoq sistemasida esa raqamdan avval 0x qo`shib yozish kerak.

```
<?php
o`nli sanoq sistemasidagi son
$a = 1234;
manfiy son
$a = -123;
sakkizlik sanoq sistemasidagi son(o`nlik sanoq sistemasidagi
sistemasidagi 83 soniga ekvivalent)
$a = 0123;
o`n oltilik sanoq sistemasidagi son (o`nlik sanoq
sistemasidagi 26 soniga ekvivalent)
$a = 0x1A;
?>
```

Butun tipning eng katta qiymati 2 milliardga yaqin (bu 32 bit belgi). Belgisiz butun qiymatlarni PHP qabul qilmaydi.

### ***Float, real, double (nuqta bilan ajratilgan son qiymati) tipi***

Nuqta bilan ajratilgan son qiymati (haqiqiy son qiymati) tipi quyidagicha sintaksislarga ega.

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

Double(ikkilik qiymat) float, real tiplari bir-biriga o`xshash bo`lib– haqiqiy son qiymatlari uchun ishlatiladi.

### ***String (satrli) tipi***

Satr – bu belgilar to`plami. PHP da boshqa dasturlardagi kabi bayt bu turli 256 ta belgini qabul qiladi. String tipi satrli qiymatlar uchun ishlatiladi.

PHP da satr turli xil usullar bilan beriladi:

- bittalik qo`shtirnoq yordamida
- ikkitalik qo`shtirnoq yordamida
- heredoc – sintaksisi yordamida

### **Bittalik qo`shtirnoq**

Satrlarni aniqlashni oddiy usuli – bu uni bittalik qo`shtirnoq «\» ichiga olib yozish kerak. Satr ichida bittalik qo`shtirnoq ishlatish uchun boshqa dasturlash tillaridagi kabi, uning oldidan teskari chiziq belgisi ko`yilishi kerak. Agar teskari chiziq belgisini satrning oxirida bittalik qo`shtirnoq oldidan quyishga to`g`ri kelsa, u holda uni yana bittaga oshirib qo`yish kerak «\\».

Agar bittalik qo`shtirnoq ichiga olingan satr ichida teskari chiziq «\» belgisi ixtiyoriy belgi oldidan ( «\» va «\« dan farqli) uchrasa, u holda bu oddiy belgi sifatida qaraladi va ekranga shu ko`rinishda chiqadi. Shuning uchun teskari chiziq satr oxirida kelgan bo`lsa, unga yana bitta teskari chiziq qo`shib yozish kerak. PHP ning teskari chiziq bilan boshlanuvchi belgilar to`plami mavjud bo`lib, ular boshqaruvchi ketma-ketliklar deb ataladi va maxsus qiymatlarga ega(/n, :// va h). Shunga ko`ra, ikkita turli sintaksislar bilan farq qiluvchi o`zgaruvchilar va maxsus belgilar uchun boshqaruvchi ketma-ketliklar satrlarda bittalik qo`shtirnoq ichida uchrasa, ular qayta ishlanmaydi.

### **4.3 - misol. Boshqaruvchi ketma-ketliklarni ishlatilishi**

```
<?php
echo ' Satrda qo`shtirnoq shu ko`rinishda ishlatiladi';
echo Agar satrda
Warning: Unexpected character in input: '\' (ASCII=92) state=1 in
D:\xampplite\htdocs\learn_php\code.php on line 2
```

```
' u holda ' .
' uni oldidan quyidagi belgi qo'yiladi \\';
// Chiqadi: Agar satrda ' ishlatilsa, u holda
// uni oldidan quyidagi belgi qo'yiladi \
echo 'Ushbu diskni o'chirasizmi C:*..*?';
// Chiqadi: Ushbu diskni o'chirasizmi C:*..*?
echo 'Yangi satrni: \n qo'ymaydi';
// Chiqadi: Yangi satrni : \n
// qo'ymaydi
echo ' O'zgaruvchi $expand $either' .
'kabi qo'yilmaydi';
// Chiqadi: O'zgaruvchi $expand
// $either kabi qo'yilmaydi
?>
```

### Ikkitalik qo'shtirnoq

PHP da ikkitalik qo'shtirnoq «"» bilan chegaralangan satrlarda maxsus belgilar uchun bir qancha boshqaruvchi ketma-ketliklar ishlatiladi. Ulardan ba'zilar 4.1 - jadvalda keltirilgan

#### 4.1 - jadval. Satrlar uchun ishlatiladigan boshqaruvchi ketma-ketliklar

Boshqaruvchi ketma-ketliklar	
Ketma - ketlik	Qiymati
\n	Yangi satr (ASCII da LF yoki 0x0A (10))
\r	Kursorni qaytishi
\t	Gorizontaal tabulyatsiya (ASCII da HT yoki 0x09 (9) )
\\	Teskari qiyshiq chiziq
\\$	Dollar belgisi
\"	Ikkitalik qo'shtirnoq

### Heredoc sintaksisi

Satrlar katta hajmdagi ma'lumotlar (jadvallar, satrlar, murojaatlar) ham bo'lishi mumkin. Bunday satrlarni echo operatoridan foydalanib ekranga chiqazish bir qancha muammolar keltirishi mumkin. Ushbu holatlarda satriy o'zgaruvchilarni *heredoc*-sintaksisi usulidan foydalanib aniqlash mumkin. Bu usulda satrlar <<< markeri bilan boshlanib, so'ng identifikator so'zi yoziladi va satr yana shu identifikator so'zi bilan tugatiladi.

#### Sintaksisi:

```
echo <<<MATN
```

```
.....
```

```
MATN;
```

Heredoc matni qo'shtirnoq ichida joylashgan matn kabi bo'ladi va heredoc sintaksisi ishlatilganda qo'shtirnoq ishlatishga xojat yo'q. Heredoc matni ichida o'zgaruvchilarni ham berish mumkin va bu holda heredoc ichidagi o'zgaruvchilar ham qayta ishlanadi.

#### 4.4 - misol. heredoc sintaksisining ishlatilishi

```
<?php
$str = <<<EEE
<table cellpadding="1" cellspacing="1" border="0" >
<tr><td>
```

```
<div class="body_txt">
$fak fakultetining tarixi
</div>
</td></tr>
</table>
EEE;
// EEE - bu yerda identifikator.
?>
```

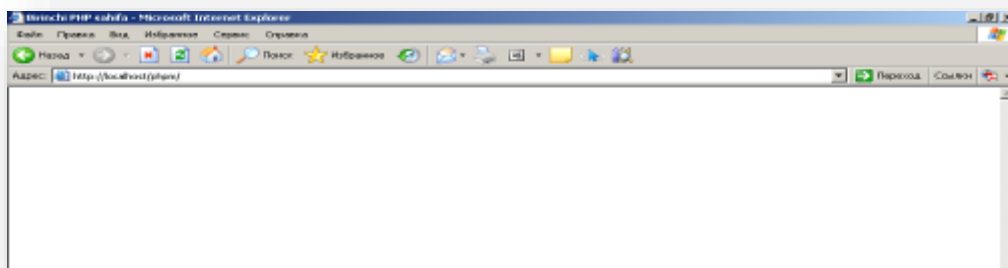
Skalyar tipdagi o‘zgaruvchilarni boolean, integer, float, string turlarini ko‘rib o‘tdik.

#### 4.5 - misol. O‘zgaruvchilarning qiymatlarini chiqazish

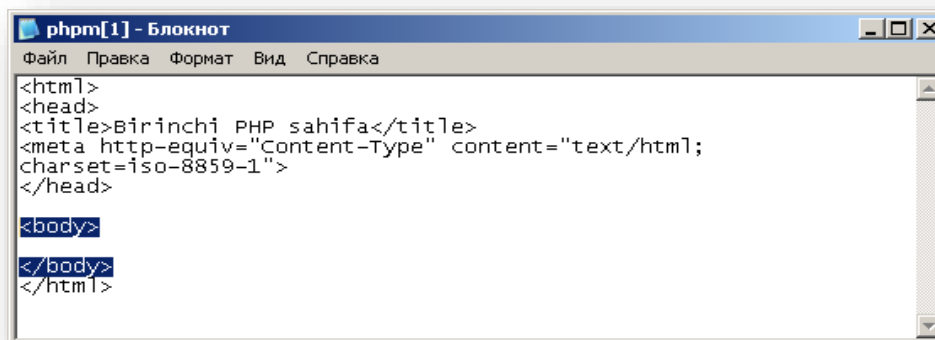
```
<?
$name="Lola";
$age=22;
$price=2.11;
?>
```

Bu yerda, \$name o‘zgaruvchisi string tipdagi, \$age o‘zgaruvchisi integer tipdagi, \$price o‘zgaruvchisi esa double tipdagi qiymatlarni qabul qiladi.

Endi sahifani saqlab, brauzerda ko‘ramiz.



Sahifada hech qanday yozuv yoki ma‘lumot yo‘q. Ichki (html) kodini ko‘radigan bo‘lsak.



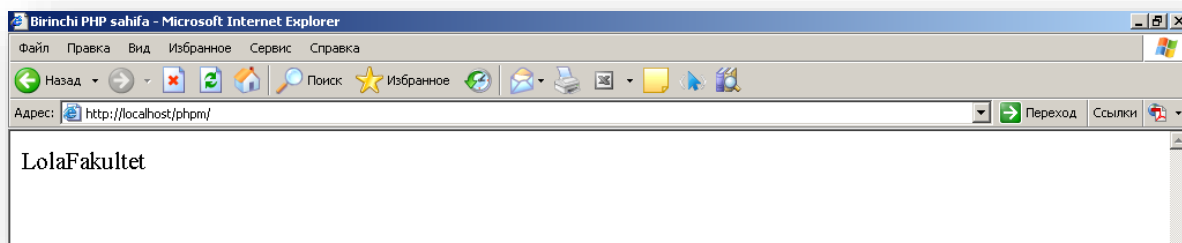
Biz yozgan PHP kodlar brauzerda ham, ichki kodda ham chiqmadi. Nimaga? Chunki biz faqatgina o‘zgaruvchilarga qiymat berdik xolos, brauzerga chiqazishga esa buyruq bermadik.

echo operatoridan foydalanib o‘zgaruvchilarning qiymatlarini brauzerga chiqazamiz.

```
<?
$name="Lola";
$age=22;
echo "$name";
?>
```

Fakultet soʻzini ham brauzerga chiqazmoqchimiz, u holda

```
<?
$name="Lola";
$age=22;
echo "$name";
echo "Fakultet";
?>
```



Bu yerda qiymatlar ketma-ket holda chiqdi. Ularni ajratish uchun bizga tanish boʻlgan `<br>` tegidan foydalanishimiz mumkin.

```
<?
$name="Lola";
$age=22;
echo "$name";
echo "
Fakultet";
?>
```

### *Aralash tipdagi oʻzgaruvchilar*

#### *Array tipi (Massivlar)*

Skalyar **tipdagi** oʻzgaruvchi – bu yagona *qiymat* saqlanadigan katakcha xotirasini nomi. Massiv **tipdagi** oʻzgaruvchi esa *qiymatlar toʻplami* saqlanadigan katakcha xotirasini nomi. Boshqacha qilib aytganda, **bir nechta tipdagi** maʼlumotlarning bir tipdagi oʻzgaruvchida **saqlanishi array (massiv)** deb ataladi. Massivda saqlanayotgan qiymatlar massiv elementlari deb ataladi. Bitta massiv bir nechta elementlardan tashkil topgan **boʻlib**, har bitta element oʻzida yagona qiymat(matn, son yoki biror boshqa massiv)ni saqlaydi. Massivning har bir elementi massiv bilan kalit orqali bogʻlanadi. Boshqa tipdagi oʻzgaruvchilarda bitta qiymat saqlansa, massiv tipdagi oʻzgaruvchilarda bir nechta qiymatlar saqlanadi. Element identifikatsiyasi usuli bilan farq qiluvchi ikkita tipdagi massiv mavjud. **Birinchi** tipdagi massiv elementi kalitlari ketma-ket keluvchi **sonli** indeks sifatida aniqlangan sonli indeksli massivlar. Massivning **ikkinchi tipi assotsiativ** shaklda boʻlib, elementga murojaat etish uchun qiymat bilan mantiqiy bogʻlangan kalitlar ishlatiladi. Oʻzida boshqa massivlarni saqlagan massiv **koʻp oʻlchovli massiv** deb ataladi.

Massivlar `array ()` konstruksiyasi yordamida eʼlon qilinadi.

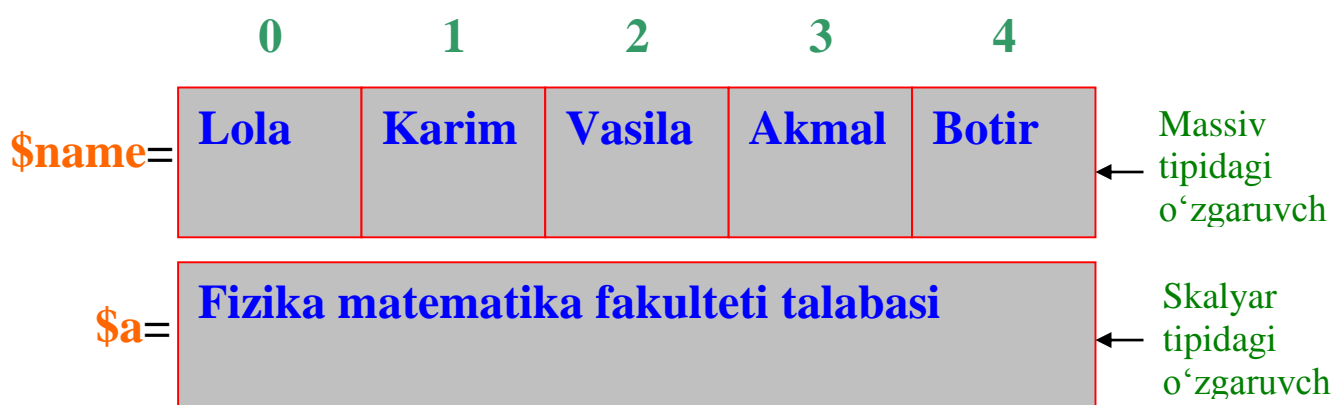
#### **Sintaksisi :**

```
array ([kalit] => qiymat,
[kalit1] => qiymat1, ...)
```

`array()` massivi juftlik *kalit=> qiymat* parametrlarini vergul bilan ajratib qabul qiladi. `=>` belgisi qiymat va uni kaliti orasidagi bogʻlanishni oʻrnatadi. Kalit tipi butun va satrli, qiymat esa PHP ning ixtiyoriy tipida boʻlishi mumkin.

Masalan, `$name=array("Lola", "Karim", "Vasila", "Akmal", "Botir");`

### Sonli indeksli massivlar



\$a o'zgaruvchisini nashr qilishga bersak, bitta qiymatni ekranga chiqazadi. \$name o'zgaruvchisi massiv bo'lganligi sababli, kalitni ko'rsatishimiz kerak. YA'ni Akmal ni ekranga chiqazmoqchi bo'lsak, \$name[3] deb kalitini ko'rsatishimiz kerak. Sonli indekslangan massivlar noldan boshlanadi.

```
$name=array (0=>"Lola", 1=>"Karim", 2=>"Vasila", 3=>"Akmal",
4=>"Botir");
```

yoki

```
$name=array("Lola", "Karim", "Vasila", "Akmal", "Botir");
```

yoki

```
$name[0]= "Lola";
```

```
$name[1]= "Karim";
```

```
$name[2]= "Vasila";
```

```
$name[3]= "Akmal";
```

```
$name[4]= "Botir";
```

Yoki

```
$name[]= "Lola";
```

```
$name[]= "Karim";
```

```
$name[]= "Vasila";
```

```
$name[]= "Akmal";
```

```
$name[]= "Botir";
```

### Assotsiativ massivlar

Ko'pgina dasturlash tillarida massiv indeksleri 0 va 1 dan boshlanuvchi sonlardan iborat. Bu dasturlash tillaridan farqli ravishda PHP assotsiativ massivlarni ham qo'llaydi. Assotsiativ massivlarni sonli indekslangan massivlardan farqi assotsiativ massivlarda kalit ixtiyoriy qiymatlar(matn, son va h.)ni qabul qiladi. Bu massivda massiv elementlari kalit sifatida berilgan sonlar bilan emas, balki satrlar bilan bog'lanadi.

**4.1 - misol.** PHP da assotsiativ massivlarni ishlatilishi

	0	1	2	3	4	
<b>\$name=</b>	Lola	Karim	Vasila	Akmal	Botir	← Sonli indeksli massiv

O'zbekiston Rossiya Fransiya Germaniya Angliya

<b>\$poytaxt=</b>	Toshkent	Moskva	Parij	Berlin	London	← Assotsiativ massiv
-------------------	----------	--------	-------	--------	--------	----------------------

```
<?
$poytaxt["Uzbekiston"]="Toshkent";
$poytaxt["Rossiya"]="Moskva";
$poytaxt["Fransiya"]="Parij";
$poytaxt["Germaniya"]="Berlin";
$poytaxt["Angliya"]="London";
echo $poytaxt["Fransiya"];
?>
```

**Natija:** Parij

Shu davlatlarni aholisi sonini ham massiv sifatida berishimiz mumkin.

#### 4.7 - misol. PHP da assotsiativ massivlarni ishlatilishi

```
<?
$poytaxt["Uzbekiston"]="Toshkent";
$poytaxt["Rossiya"]="Moskva";
$poytaxt["Fransiya"]="Parij";
$poytaxt["Germaniya"]="Berlin";
$poytaxt["Angliya"]="London";
$aholisi=array ("Uzbekiston"=>"25", "Rossiya"=>"150", "Fransiya"=>"40",
"Germaniya"=>"35", "Angliya"=>"35");
echo "Uzbekistonning poytaxti $poytaxt["Uzbekiston"], aholisi $aholisi["Uzbekist
on"] mln kishidan iborat.";
?>
```

**Natija:** Uzbekistonning poytaxti Toshkent, aholisi 25 mln kishidan iborat.

#### Ko'p o'lchovli massiv

Massiv kalit va qiymatlarni o'zida mujassamlashtirgan oddiy ro'yxatdan iborat bo'lishi shart emas, balki massiv yacheykasi o'zida boshqa massivlarni ham saqlashi mumkin. Bunday massivlar ko'p o'lchovli massivlar deyiladi. Bu massivda aralash (sonli va assotsiativ) indekslashga ruxsat beriladi.

Ko'p o'lchovli massiv sifatida, ikki o'lchovli massivlarni ham yaratish mumkin. Ikki o'lchovli massiv kengligi va balandligi satrlar va ustunlarga akslanuvchi matritsa yoki jadval ko'rinishida bo'ladi.

#### 4.2 -misol. Ko'p o'lchovli massivlarni ishlatilishi

Assotsiativ massiv

	O'zbekiston	Rossiya	Fransiya	Germaniya	Angliya
<b>\$poytaxt=</b>	Toshkent	Moskva	Parij	Berlin	London



---

\$auto massivini qaraydigan bo'lsak, Nexia indeksli massiv qiymati boshqa bir massivga teng. Demak bu ko'p o'lchovli massivni PHP kodini yozib chiqamiz.

```
<?
$auto["nexia"]= array("yili"=> "1999", "rang"=> "oq");
$auto["damas"]= array("yili"=> "2003", "rang"=> "qora");
$auto["matiz"]= array("yili"=> "2005", "rang"=> "oq");
$auto["tiko"]= array("yili"=> "1999", "rang"=> "qizil");
?>
```

Ko'p o'lchovli massiv elementlarini kiritdik. Endi shu massiv elementlarini ekranga chiqazib ko'ramiz.

```
<?
$auto["nexia"]= array("yili"=> "1999", "rang"=> "oq");
$auto["damas"]= array("yili"=> "2003", "rang"=> "qora");
$auto["matiz"]= array("yili"=> "2005", "rang"=> "oq");
$auto["tiko"]= array("yili"=> "1999", "rang"=> "qizil");
echo "Matiz avtomobilini ishlab chiqarilgan yili - ". $auto["matiz"]["yili"];
?>
Natija.
Matiz avtomobilini ishlab chiqarilgan yili - 2005
```

### Massiv kalitlari

Agar kalit berilmagan bo'lsa, u holda kalit o'rniga kalitning eng katta qiymatiga 1 sonini qo'shib yozib ketiladi. Kalit biror - bir qiymatga teng bo'lsa, u holda o'sha qiymat o'zi yoziladi. PHP ning 4.3.0 dan keyingi versiyalarida kalitning eng katta qiymati manfiy son bo'lsa, massivning keyingi qiymati nol(0) deb olinadi.

Agar kalit o'rnida TRUE yoki FALSE qiymatlari ishlatilsa, u holda kalit tipi integer bo'lgan 0 yoki 1 qiymatni beradi. Kalit o'rnida NULL ishlatilsa, bo'sh satrni beradi. Kalit o'rnida bo'sh satr ham ishlatilishi mumkin, faqat uni qo'shtirnoq ichiga olib yozish kerak yoki alohida yozilganda bo'sh to'rtburchak qavs ishlatiladi. Kalit o'rnida ob'jekt va massivlar ishlatilmaydi.

#### 4.9 - misol. PHP da massivlarni ishlatilishi

```
<?php
// $arr va $arr1 massivlari ekvivalent
$arr = array(-3 => 43, 32, 56, 86, 97, "b" => 12);
$arr1 = array(-3 => 43, 0 => 32, 1 => 56, 2=> 86,
3 => 97, "b" => 12, "s" => true, " " => 28);
Echo $arr1[null];
```

```
?>
```

Agar yangi kalit va yangi qiymat berilsa(`$name["new_key"]="new_value"`), u holda massivga yangi element qo'shilgan hisoblanadi. Agar biz kalitni ko'rsatmasak, faqatgina qiymatga tenglasak, `$name[]="new_value"`, u holda massivning yangi elementi sonli kalit bo'lib, mavjud kalitning eng katta qiymatidan bir qiymatga oshgan bo'ladi.

```
<?
$name=array(2=>5, 3=>6,4=>15);
$name[] = value1;
/* qiymati value1 kaliti 5 (bizda kalitning
eng katta qiymati 4 bo'lganligi uchun)ga
teng massiv kiritdik */
echo $name;
?>
```

Massivning elementini o'zgartirish uchun, kalitini yangi qiymatga tenglashtirish yetarli. Massiv kalitini o'zgartirish mumkin emas, faqatgina element(kalit/qiymat juftligini)ni o'chirib yuborish va yangisini kiritish mumkin. Massiv elementini o'chirish uchun, **unset()** funksiyasidan foydalaniladi.

```
<?php
$name=array(2=>5, 3=>6,4=>15);
$name[] = value1;
/* qiymati value1 kaliti 5 (bizda kalitning
eng katta qiymati 4 bo'lganligi uchun)ga
teng massiv kiritdik */
echo $name;
$name[] = "Sobir";
// kaliti(indeksi)5 ga teng
// $name[5] = "Sobir" ga ekvivalent
// element yaratildi
Echo $name.'
```

Massiv elementi kiritilayotganda bo'sh to'rtburchak qavs ishlatilsa, massiv elementiga kalit qo'yish uchun oxirgi qayta indekslangan massiv elementlari orasidan kaliti eng kattasini qidiradi. Massivni qayta indekslash uchun **array\_values()** funksiyasidan foydalaniladi.

#### 4.10 - misol. Massivlarni qayta indekslash.

```
<?php
$arr =
array ("a","b","c");
/* qiymati "a", "b" va "c" ga teng.
Kalit ko'rsatilmaganligi uchun
mos ravishda 0,1,2 ga teng bo'ladi.*/
print_r($arr);
// massiv(kalit, qiymat)ni ekranga chiqazamiz
unset($arr[0]);
unset($arr[1]);
unset($arr[2]);
// massivni barcha qiymatlarini o'chiramiz.
```

```

print_r($arr); // massiv(kali, qiymat)ni ekranga chiqazamiz
$arr[] = "aa"; // massivga yangi element qo'shamiz,
// uni indeks (kaliti) nol emas, 3 bo'ladi
print_r($arr);
$arr = array_values($arr); // massivni qayta indekslaymiz
$arr[] = "bb"; // bu element kaliti 1 bo'ladi
print_r($arr);
?>
Natija:
Array ([0] => a [1] => b [2] => c) Array () Array ([3] => aa) Array ([0] =
> aa [1] => bb)

```

### **Object tipi (ob'yektlar)**

**Ob'yektlar** – ob'yektga yo'naltirilgan dasturlashdan(OYD) olingan ma'lumotlar tipi. OYD prinsipiga asoslanib, klass - bu xususiyati va ishlash metodi aniqlangan ob'yektlar to'plami, ob'yekt esa klassdan olingan bir nusxa.

Biror bir vazifani bajaruvchi amallardan tashkil topgan klassni e'lon qilamiz. Klass e'lon qilib bo'lingandan so'ng, klassni a'zosi bo'lgan alohida aniq element – **ob'yektni** yaratamiz. Bu jarayon klassni nusxasini yaratish deb nomlanadi. Ob'yekt new kalit so'zi yordamida yaratiladi. Buning uchun, ob'yekt qaysi klassni nusxasi ekanligini va barcha parametrlarni ko'rsatish kerak. Ob'yektlar bilan ishlashda klasslarni chaqirish uchun → operatoridan foydalaniladi.

**Misol uchun**, dasturchilar klassi - bu dastur tuza oladigan, kompyuterda ishlay oladigan va bundan tashqari boshqa insonlar kabi ism va familiyaga ega insonlar klassi. Endi, agar birorta aniq bir dasturchi, masalan, Odil Karimovni oladigan bo'lsak, u holda u boshqa dasturchilar kabi dastur tuza oladigan, ism - familiyaga va boshqa xususiyatlarga ega dasturchilar klassining ob'yekti hisoblanadi.

#### **4.11 – misol. PHP da ob'yektlar.**

```

<?php
//insonlar klassini e'lon qilamiz
class person
{
 function prog_learn($name)
 {
 echo "$name PHP da dastur tuza oladi";
 }
 function lang_learn($name1)
 {
 echo "$name1 ingliz tilidi gaplasha oladi";
 }
}
// klassni obektlarini yaratamiz
$a = new person();
$aa=$a->prog_learn("Lola");
$b=$a->lang_learn("Karim");
$c= $a-> prog_learn();
?>

```

**Klass har bitta obekt yaratilganda quyidagi natijani chiqaradi:**  
Lola PHP da dastur tuza oladi  
Karim ingliz tilida gaplasha oladi  
PHP da dastur tuza oladi

### **Maxsus tipdagi o'zgaruvchilar**

#### **resource tipi (resurslar)**

---

**Resurs** – bu tashqi resursga (masalan, ma'lumotlar bazasi bilan bog'lanishga) murojaatni ta'minlovchi maxsus o'zgaruvchi. Resurslar maxsus funksiyalar (masalan, `mysql_connect()`, `pdf_new()` va boshqalar) bilan yaratiladi va ishlatiladi.

### **Null tipi**

Maxsus qiymat `NULL` o'zgaruvchi qiymatga ega emasligini anglatadi.

O'zgaruvchi `NULL` hisoblanadi, agar:

u `NULL` konstantasiga tenglashtirilgan bo'lsa (`$var = NULL`);

u hali biror – bir qiymatga tenglashtirilmagan bo'lsa;

`unset()` yordamida o'chirilgan bo'lsa.

Faqatgina bitta registrga bog'liq bo'lmagan `NULL` kalit so'zli `NULL` tipidagi qiymat mavjud.

### ***O'zgaruvchilarning tiplarini tekshirish va o'rnatish***

#### **Sintaksisi:**

`string gettype(mixed var);`

`int settype(string var, string type);`

**gettype()** funksiyasi unga uzatilgan o'zgaruvchining tipini aniqlab, tip nomini yoki agar o'zgaruvchi tipi standart tiplardan (`integer`, `double`, `string`, `array` yoki `object`) birortasi ham bo'lmasa, "unknown type" ni qaytaradi.

**settype()** funksiyasiga parametrlar sifatida o'zgaruvchi va standart tiplardan biri beriladi. Bu funksiya parametr sifatida berilgan o'zgaruvchi tipini parametr sifatida berilgan yangi tipga o'zgartiradi.

#### **Misol:**

```
<?
$a = 56;
echo gettype($a)."
";
settype ($a, "double");
echo gettype ($a)."
";
?>
```

Natija:

integer

double

**gettype()** funksiyasiga murojaat etganda, `$a` o'zgaruvchisining tipi **integer** ekanligini chiqazadi. **settype()** funksiyasiga murojaat etgandan so'ng, uning tipi **double** ga o'zgaradi.

PHP ning tiplarni tekshirish uchun bir qancha funksiyalari mavjud. Bu funksiyalarning har biri parametr sifatida o'zgaruvchini qabul qiladi va `true` yoki `false` qiymatlarini qaytaradi.

- **is\_array()**
- **is\_double(), is\_float(), is\_real()** (bir xil funksiya)
- **is\_long(), is\_int(), is\_integer()** (bir xil funksiya)
- **is\_string()**
- **is\_object()**

### ***O'zgaruvchi holatini tekshirish***

PHP ning o'zgaruvchi holatini tekshiruvchi bir qancha funksiyalari mavjud. Ulardan biri – **isset()**.

#### **Sintaksisi:**

`int isset(mixed var);`

---

bu funksiya parametr sifatida o'zgaruvchi nomini oladi va natija sifatida agar o'zgaruvchi mavjud bo'lsa, true, aks holda false ni qaytaradi.

O'zgaruvchini **unset()** funksiyasi yordamida o'chirish mumkin. Uning **sintaksisi** quyidagicha:

*int unset(mixed var);*

bu funksiya o'zgaruvchini o'chiradi va natija sifatida true qiymatini qaytaradi.

**empty()** funksiyasi qiymati bo'sh va nol bo'lmagan o'zgaruvchining mavjudligini aniqlaydi va mos holda true yoki false qiymatlarini qaytaradi.

**Sintaksisi:**

*int empty(mixed var);*

Ushbu uchta funksiyaga misol ko'ramiz. Vaqtinchalik yaratilgan sahifaga ushbu kodlarni kiritamiz:

```
<?
echo isset($bor);
echo isset($yuq);
echo empty($bor);
echo empty($yuq);
?>
```

**isset()** funksiyasi **\$bor** o'zgaruvchisi uchun forma maydonida qiymat kiritilgan yoki kiritilmaganligiga bog'liq bo'lmagan holda, true qiymatini qaytaradi. **empty()** funksiyasi esa kiritilgan qiymatga bog'liq holda natijani qaytaradi.

\$yuq o'zgaruvchisi mavjud emas, shuning uchun bu o'zgaruvchi uchun **isset()** funksiyasi false qiymatini, **empty()** funksiyasi esa – true ni qaytaradi.

Ko'rib o'tilgan funksiyalar formadagi mavjud maydonlar foydalanuvchi tomonidan to'ldirilganligini tekshirish uchun ishlatiladi.

### ***O'zgaruvchilarni qayta interpretatsiyalash***

**Sintaksisi:**

*int intval(mixed var);*

*double doubleval(mixed var);*

*string strval(mixed var);*

ushbu har bir funksiya parametr sifatida o'zgaruvchini qabul qiladi va mos tipga o'zgartirilgan o'zgaruvchi qiymatini qaytaradi.



### **Savol va topshiriqlar:**

1. PHPda o'zgaruvchilar nima?
2. PHPda konstantalar nima?
3. PHPda tashqi o'zgaruvchilar nima?



## **4.7. PHP tilining amallari**

**Amal belgilari** – bular o'zgaruvchilar va qiymatlar ustida amallar bajarish uchun ishlatish mumkin bo'lgan belgilar.

## 9.1 - jadval. PHP amallari

Amallar	Mazmuni
()	Qavs
new	Ob'yekt nusxasini yaratish, yangi ob'yekt hosil qilish
!~	Mantiqiy rad etish
++,--	Bitta songa oshirish (inkrement), kamaytirish (dekrement)
@	Xatolarni yashirish
/,*,%	Bo'lish, ko'paytirish, qoldiq
+,-,.	Qo'shish, ayirish, konkatenatsiya
<<,>>	Chapga surish, o'ngga surish(razryadli)
<,<=,>,>=	Kichik, kichik yoki teng, katta, katta yoki teng
==,!==,===,<>	Teng, teng emas, aynan teng, teng emas
&,&^,&	Razryadli amallar AND, XOR va OR
&&,&	Mantiqiy amallar AND va OR
?:	Ternar amal
=,+==,*==,/==,.=,%==, ?==,/==,:==,<<==,>>==,	Tenglik amallari
AND, XOR, OR	Mantiqiy amallar AND, XOR va OR

## Satrlar bilan ishlash amallari

*O'zgaruvchilar va literallar.* O'zgaruvchi ma'lumotlar belgisi. \$name – o'zgaruvchi yoki biror bir ma'lumotni o'zida saqlovchi belgi. Satrlar ham ma'lumotlar. "talaba"- bu literal, u qanday bo'lsa shunday qabul qilinadi.

### Satrlarni birlashtirish

PHP da satrlar bilan ishlash uchun eng qulayi konkatenatsiya(satrlarni birlashtirish) amali. Konkatenatsiya ikki yoki undan ortiq ob'yektlarni bir butun qilib birlashtirish degan ma'noni bildiradi. Satrlar bilan ishlash uchun ikkita amal mavjud: konkatenatsiya amali(.) va konkatenatsiya bilan tenglik amali(.=).

### 5.2 - jadval. Satrlar bilan ishlash amali

Belgisi	Nomi	Misol	Natija
.	Konkatenatsiya	\$a="abc"."def"	\$a o'zgaruvchisi abc va def larning birlashtirish natijasiga teng.
.=	Konkatenatsiya bilan tenglik	\$a.="ghijkl"	\$a o'zgaruvchisi uning mavjud qiymati bilan "ghijkl" satrini birlashtirish natijasiga teng.

Konkatenatsiya amali yordamida satrlar bilan satrlarni, o'zgaruvchilar bilan satrlarni birlashtirish mumkin. Bu amaldan brauzerga matnlarni chiqazishda tez-tez foydalaniladi va echo buyrug'ini bir necha marotaba berishga zarurat tug'ilmaydi.

### Satrlar bilan ishlash amallariga misollar keltiramiz:

#### 5.1- misol. Satrlar bilan ishlash.

```
<? $a="Olma"." va Nok";
// $a "Olma va Nok" satrli qiymatga teng
```

```
$a=" mevalar";
// $a "Olma va Nok mevalar" satrli qiymatga teng
?>
```

### 5.2-misol. Satrlar bilan ishlash.

```
<?
$name="Lola";
$age=22;
echo "$name";
echo $name." talaba
";
?>
```

### Yoki

```
<?
$name="Lola";
$age=22;
echo "$name";
echo "$name talaba
";
?>
```

deb ham yozish mumkin. Ikkala hol ham bir xil natija beradi.

Conli o'zgaruvchilar ustida ham konkatenatsiya amalini bajarish mumkin.

### 5.3- misol. Satrlar bilan ishlashda konkatenatsiya amali

```
<?
$a=39;
$b=10;
$n=$a.$b;
echo $n;
?>
Natija:3910
```

Demak, (.) konkatenatsiya amali \$a va \$b o'zgaruvchilarni qiymatlarini bir-biriga qo'shib qo'ymaydi, balki ularni satriy kattalik sifatida ketma-ket joylashtiradi.

Endi o'zgaruvchi qiymatlari, matnlarni va HTML teglarini birgalikdagi kombinatsiyasini chiqazishini ko'ramiz.

### 5.4- misol. Satrlar bilan ishlash

```
<?
$name="Lola";
$age=22;
echo "
$name Fizika-
matematika fakultetida o'qiydi.
Uni yoshi $age da."
?>
Natija:
Lola Fizika-matematika fakultetida o'qiydi.
Uni yoshi 22 da.
```

O'zgaruvchilarga ba'zi xollarda qiymat berilayotganda quyidagi kabilarga duch kelish mumkin.

```
<?
$a= "Lola "Ulug'bek stipendiyasi" sovrindori";
echo $a;
?>
```

Agar o'zgaruvchiga shu tartibda qiymat beriladigan bo'lsa, u holda brauzerda xatolik haqida xabar chiqaziladi, ya'ni ikkitalik qo'shtirnoq ichida ikkitalik qo'shtirnoq bu xolatda ishlatilmaydi. Agar ikkinchi qo'shtirnoqning ochilishi va yopilishi oldidan \ belgisi qo'llanilsa, u holda xatolik yuzaga kelmaydi.

```
<?
$a= "Lola \"Ulug'bek stipendiyasi\" sovrindori";
echo $a;
?>
```

PHP sahifada HTML teglarini qo'llanilishini ko'rib chiqamiz.

```
<?
echo "<p align="center">Fizika - matematika fakulteti</p>";
?>
```

HTML tegini PHP sahifada bunday holda qo‘llashimiz xatolikka olib keladi. Bu yerda yuqoridagi singari ikkinchi qo‘shirnoq ochilishi va yopilishi oldidan \ belgisi ishlatilsa bo‘ladi, lekin uni o‘rniga quyidagi holda yozish ham mumkin.

```
<?
echo "<p align='center'>Fizika - matematika fakulteti</p>";
?>
```

### Arifmetik amallar.

Biz ikkita amal bilan tanishmiz: (=) tenglik amali va (.) satrlarni birlashtirish amali. Endi arifmetik amallar bilan tanishib chiqamiz. O‘zgaruvchilar ustida barcha arifmetik amallarni bajarish mumkin.

#### 5.3 - jadval. Arifmetik amallar

Belgisi	Ko‘rinishi	Nomi
+	$a+b$	qo‘shish
-	$a-b$	ayirish
*	$a*b$	ko‘paytirish
/	$a/b$	bo‘lish
%	$a\%b$	modul olish

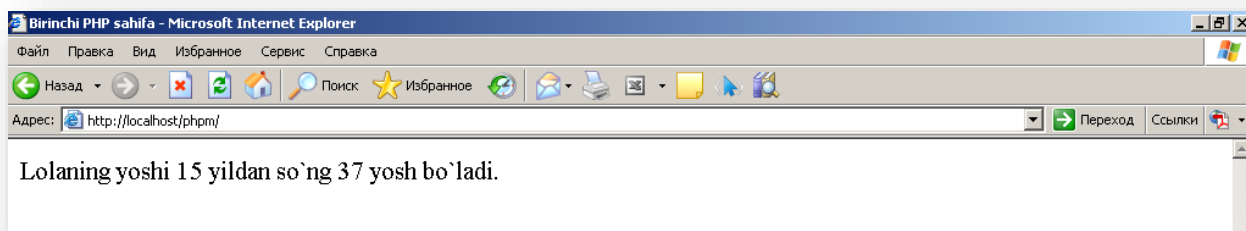
Qo‘shish va ayirish amallari oddiy bajariladi. Ularni natijasi  $a$  va  $b$  o‘zgaruvchilarda saqlanayotgan qiymatlarning yig‘indisi va ayirmasiga teng. Ayirish amalidan unar amal sifatida ham foydalanilsa bo‘ladi. Bunda o‘zgaruvchiga manfiy qiymat beriladi,  $a=-1$ .

Ko‘paytirish va bo‘lish amallari ham oddiy ishlatiladi. E’tibor bering, belgilar an’anaviy matematikadagi belgilarni, ya’ni yulduzcha ko‘paytirish amalini belgisini va egilgan tayoqcha bo‘lish amalini belgisini beradi.

#### 5.5 – misol. Qo‘shish amali.

```
<?
$name="Lola";
$age=22;
$n=15;
$sum=$age+$n;
echo $name. "ning yoshi $n yildan so'ng $sum yosh bo'ladi."
?>
```

#### Natija:



#### 5.6 – misol. Modul olish amali.

```
<?
```



```

$a=39;
$b=10;
$natija=$a%$b;
/* Bu yerda $natija o'zgaruvchisi 39 ni 10 ga bo'lingandagi qoldiq 9 ga teng bo'ladi. */
?>
Natija:
9

```

## 5.6 – misol. Arifmetik amallarning ishlatilishi

```

<?
$a=10;
$b=17;
$qushish=$a+$b;
$ayirish=$a-$b;
echo $qushish." - $a va $b ning yig'indisi.
";
echo $ayirish." - $a va $b ning ayirmasi.
";
$a=20;
$b=5;
$kupay=$a*$b;
$bulish=$a/$b;
echo $kupay."- $a va $b ni kupaytmasi.
";
echo $bulish."- $a ni $b ga bulinmasi.";
?>
Natija:
27 - 10 va 17 ning yig'indisi.
-7 - 10 va 17 ning ayirmasi.
100 - 20 va 5 ni kupaytmasi.
4- 20 ni 5 ga bulinmasi.

```

Arifmetik amallarni tenglashtirish amali bilan birga boshqacha ko‘rinishda ham ishlatish mumkin. **Masalan,**

$$b=12+(a=4);$$

Bu holda  $b$  o‘zgaruvchisi 16 ga teng bo‘ladi. Chunki,  $a$  o‘zgaruvchisi 4 qiymatni qabul qilyapti va 12 ga qo‘shilyapti, shuning uchun  $12+4=16$  ga teng.

Oddiy tenglashtirish amallaridan tashqari kombinatsiyalangan tenglashtirish amallari yoki qiymat uzatish amallari ham mavjud.

Qiymat uzatish amali har bir arifmetik amal va konkatenatsiya amali uchun ham mavjud.

### 5.4 - jadval. Qiymat uzatish amali

Belgisi	Ishlatilishi	Ekvivalenti
+=	$a+=b$	$a=a+b$
-=	$a-=b$	$a=a-b$
*=	$a*=b$	$a=a*b$
/=	$a/=b$	$a=a/b$
%=	$a%=b$	$a=a\%b$
.=	$a.=b$	$a=a.b$

**Masalan,**

$$a+=8$$

bu yozuv

$a=a+8$  yozuviga ekvivalent.

inkrement(++ ) va dekrement(-- ) yordamchi amallari dasturda o‘zgaruvchi qiymatini 1 ga kamaytirish yoki oshirish uchun xizmat qiladi.

### 5.5 - jadval. Inkrement va dekrement amallari

Misol	nomi	Natija
++\$a, \$a++	Inkrement	\$a o'zgaruvchisini 1 ga oshiradi
--\$a, \$a--	dekrement	\$a o'zgaruvchisini 1 ga kamaytiradi

Prefiksli va suffiksli inkrement(++ ) va dekrement(-- ) amallari += va -= amallariga o'xshash. **Masalan,**

```
<?
$a=4;
echo ++$a;
?>
Natija:5
```

Ikkinchi satrda prefiksli inkrement amali ishlatilyapti, chunki \$a dan oldin (++) belgisi mavjud. Natijada \$a ning qiymati 1 ga oshiriladi va oshirilgan natija chiqaziladi. Bu holda, \$a ning qiymati 5 ga oshiriladi va 5 ekranga chiqaziladi, ya'ni \$a o'zgaruvchining qiymati avval oshirilib, keyin natijaga beriladi.

Agar (++) belgisi \$a dan keyin ishlatilsa, demak suffiksli inkrement amali ishlatilyapti, u holda natija boshqacha bo'ladi. **Masalan,**

```
<?
$a=4;
echo $a++;
?>
Natija:4
```

bu holda amal teskari tartibda bajariladi. Avval \$a qiymatni qabul qiladi va natijaga chiqazadi, shundan so'ng uni qiymati 1 taga oshiriladi. Bunda natijaga 4 soni chiqaziladi, keyin \$a ning qiymati 5 ga teng bo'ladi.

Inkrement va dekrement amallari bir-biriga o'xshash. Faqatgina, prefiksli va suffiksli inkrement amalida \$a ning qiymati 1 taga oshirilsa, dekrement amalida esa \$a ning qiymati 1 taga kamaytiriladi.

### Mantiqiy binar amallar.

Mantiqiy amallar PHP da ixtiyoriy dastur tuzish jarayonida, o'zgaruvchi qiymatlari bilan bog'liq bo'lgan hollarda muhim rol o'ynaydi. Mantiqiy amallar dasturda buyruqlar bajarilish tartibini boshqarish imkoniyatini beradi va boshqaruvchi konstruksiyalar (shartli if operatori va for, while sikllari) bilan tez-tez ishlatiladi.

Mantiqiy shartlarni natijalarini birlashtirish uchun mantiqiy amallar ishlatiladi. Misol uchun \$a o'zgaruvchisi 0 va 100 orasidagi qiymatlarni qabul qiladi deylik. Bu holda AND(VA) amalini ishlatib, \$a>=1 va \$a<=100 shartlarni tekshiramiz.

\$a>=1 && \$a<=100

PHP texnologiyasi AND(VA), OR(YOKI), XOR(yoki mustasno) va NOT(EMAS) kabi mantiqiy amallarni ishlatadi.

### 5.6 - jadval. Mantiqiy amallar.

Belgisi	nomi	ishlatilishi	natija
&&	Va	\$a&&\$b	\$a va \$b o'zgaruvchilarning ikkalasi ham rost

			qiymat qabul qilsa, rost, aks holda yolg'on
AND	Va	$\$a \text{ AND } \$b$	&& kabi mantiqiy amal
	Yoki	$\$a    \$b$	$\$a$ yoki $\$b$ o'zgaruvchilarning ikkalasidan birortasi rost qiymat qabul qilsa, rost, aks holda yolg'on
OR	yoki	$\$a \text{ OR } \$b$	kabi mantiqiy amal
XOR	yoki	$\$a \text{ XOR } \$b$	o'zgaruvchilarning ikkalasidan faqatgina bittasi rost bo'lsa, rost
!	Emas	$!\$a$	agar $\$a$ yolg'on bo'lsa, natija rost va aksincha

### Tenglik amali

Tenglik amali = =(ikkita teng belgisi) ikkita qiymatni bir-biriga taqqoslaydi va ularni tengligini tekshiradi. masalan,

$\$a = \$b$ ;

$\$a$  va  $\$b$  o'zgaruvchilarda saqlanayotgan qiymatlarning tengligini tekshirish uchun ishlatiladi. Agar ikkala o'zgaruvchining qiymati bir-biriga teng bo'lsa, natija true ni, aks holda false ni beradi.

Bu amalni tenglashtirish amali bilan adashtirish mumkin. Bu vaqtda dasturda xatolik yuzaga kelganligi haqida xabar chiqmasligi mumkin, lekin istalgan natijaga erishib bo'lmaydi. Umumiy hollarda nol bo'lmagan qiymat true ni, nol qiymatlar esa – false ni beradi. Quyidagi holatni ko'rib chiqamiz:

$\$a = 5$ ;

$\$b = 7$ ;

$\$a = \$b$  ni tekshiradigan bo'lsak, natija true ni beradi. Nima uchun?

Chunki  $\$a = \$b$  qiymatga ega, ya'ni o'ng tomondagi operandning qiymati ayni vaqtda 7 ga teng, bu esa nol bo'lmagan qiymat. Shuning uchun ifoda natijasi true ni beradi.

Agar  $\$a = \$b$  ifodani tekshiradigan bo'lsak, bu holda natija false ni beradi, chunki  $\$a$  va  $\$b$  ning qiymati bir-biriga teng emas.

Demak, = va = = belgilari ikki hil natija berdi. Shuning uchun dastur kodlari yozayotgan vaqtda qaysi taqqoslash amali kerakligini aniqlab, keyin bu belgilarni ishlatish kerak.

#### 5.7 - jadval. Tenglik amallari.

Amal belgisi	Nomi	Ishlatilishi	Natija
= =	Teng	$\$a = \$b$	$\$a$ va $\$b$ teng bo'lsa, rost
= = =	Aynan teng	$\$a = = \$b$	$\$a$ va $\$b$ aynan teng(teng va bir xil tipli) bo'lsa, rost
!=	Teng emas	$\$a != \$b$	$\$a$ va $\$b$ teng bo'lmasa, rost
0	Teng emas	$\$a 0 \$b$	$\$a$ va $\$b$ teng bo'lmasa, rost

### Munosabat(taqqoslash) amali.

Taqqoslash amali ikkita qiymatni bir-biri bilan solishtirish uchun ishlatiladi. Taqqoslash amallari xuddi mantiqiy amallar kabi ishlaydi, dastur mantig'ini boshqaradi va ikki yoki undan ortiq o'zgaruvchilarni taqqoslab xulosa qabul qiladi.

Bu amallar ishlatilganda taqqoslash natijasi true(rost) yoki false(yolg'on) mantiqiy qiymatlarga teng bo'ladi.

### 5.8 - jadval. Taqqoslash amallari

Amal belgisi	Nomi	Ishlatilishi	Natija
<	Kichik	$\$a < \$b$	$\$a$ o'zgaruvchisi $\$b$ dan kichik bo'lsa, rost
>	Katta	$\$a > \$b$	$\$a$ o'zgaruvchisi $\$b$ dan katta bo'lsa, rost
<=	Kichik yoki teng	$\$a <= \$b$	$\$a$ o'zgaruvchisi $\$b$ dan kichik yoki teng bo'lsa, rost
>=	Katta yoki teng	$\$a >= \$b$	$\$a$ o'zgaruvchisi $\$b$ dan katta yoki teng bo'lsa, rost
$(\$a=12)?5:-1$	Ternar amal		Agar $\$a$ o'zgaruvchisi 12 ga teng bo'lsa, 5 qiymatini, aks holda -1 qiymatini chiqazadi

### Ternar amali

*Ternar amali* (?:) xuddi S dagi kabi ishlatiladi.

#### Sintaksisi:

*shart ? shart rost bo'lgandagi qiymat: shart yolg'on bo'lgandagi qiymat*

Ternar amali if-else operatoriga o'xshash.

### 5.7 – misol. Ternar amalining ishlatilishi

```
<?
($natija>50 ? "Imtihondan o'tdingiz" : " Imtihondan o'tmadingiz")
?>
```

Bu ifoda talabalarning imtihondan o'tgan yoki o'tmaganligini aniqlovchi shart sifatida ishlatiladi.

### Xatolarni yashirish amali

Xatolarni yashirish amali @ biror qiymatga ega yoki ishlab chiqilayotgan yozuv, ixtiyoriy ifoda oldidan ishlatilishi mumkin.

### 5.8 – misol. Xatolarni yashirish amali

```
<?
$a=@(57/0)
?>
```

Bu qator @ belgisisiz yozilsa, nolga bo'lish mumkin emasligi haqidagi ogohlantirishni beradi. Agar @ belgisi ishlatilsa, u holda bu xatolik haqidagi xabar yashiriladi. Agar PHP track\_errors funksiyasi bilan o'rnatilgan bo'linsa, xatolar haqidagi xabarlar \$php\_errormsg global o'zgaruvchisida saqlanadi.



## 4.8. PHP tilining operatorlari

Operatorlar quyidagi turlarga bo'linadi:

- Tanlash operatorlari – shartli o'tish operatorlari(if-else) va qayta ulovchi (switch-case) operatorlar

---

- Takrorlash(sikl) operatorlari - sharti oldindan berilgan operator (while), parametrli takrorlash opertori(for) va sharti keyin keluvchi operatorlar(do...while).

- O'tish operatori – boshqarishning shartsiz uzatishini ta'minlaydi. Ularga continue(takrorlashning hozirgi iteratsiyasini yakunlash), break(takrorlashdan chiqish) va return(funksiyalardan chiqish) lar kiradi.

- Qamrab olish operatorlari (include, require).

## Tanlash operatorlari

### *if, else – shartli o'tish operatorlari*

Boshqa tillardagi kabi if shartli o'tish operatori PHP ning asosiy operatorlaridan biri. IF operatori shartning berilishiga qarab, kod blokini bajaradi. If shartli o'tish operatorining sharti ( ) qavs ichiga olib yoziladi.

Sintaksisi:

*if (shart) bajarish\_bloki;*

bu yerda shart biror bir mazmunga ega bo'lib, skriptni qayta ishlash jarayonida mantiqiy qiymatga o'zgartiriladi. Agar shart bajarilsa, true(rost) qiymatni qabul qiladi, ya'ni shart mantiqiy true(rost) qiymatga o'zgaradi va keyingi qadam (*bajarish\_bloki*) bajariladi. Aks holda *bajarish\_bloki* bekor qilinadi.

Shartlarning mantiqiy qiymatlarga aylanish holatlari

1. False qiymatga o'zgaradigan holatlar:

- Mantiqiy false
- Butun nol (0)
- Bo'sh qator va «0» qatori
- Elementlarsiz massiv
- O'zgaruvchilarsiz ob'yekt
- Maxsus NULL

2. Qolgan barcha holatlar True qiymatga o'zgaradi.

### **7.1 –misol.** If shartli o'tish operatori

Ma'lumot kiritish uchun forma berilgan. Ismi kiritilganmi yoki yo'qligi tekshiriladi va mos javob qaytariladi. If shartli o'tish operatori shart yordamida foydalanuvchi logini kiritilgan yoki kiritilmaganligini tekshiradi va mos xabarni chiqazadi.

```
<?
if (!$name)
echo "Ismingizni kiritmadingiz!";
?>
```

Bu misolda !\$name sharti ishlatilmoqda.

Agar \$name o'zgaruvchisining qiymati bo'sh bo'lsa, !\$name sharti true(rost) qiymatni qabul qiladi va shart bajarilganligi uchun keyingi qadamga o'tiladi.

IF shartli o'tish operatoridan so'ng faqatgina bittadan ortiq bo'lmagan *bajarish\_bloki* bajariladi. Agar *bajarish\_bloki* bir nechta buyruq satridan iborat bo'lsa, har bir buyruq satri oldidan if shartli o'tish operatori qo'llaniladi yoki buyruq satrlari {} qavs ichiga olib yoziladi, ya'ni bir blokka birlashtiriladi.

```
<?
if (!$name)
```

```

{
echo "";
echo "Ismingizni kiritmadingiz";
echo "";
}
?>

```

### ***Else operatori***

Odatda faqatgina shart bajarilgan holatlar qaralmaydi, balki u bajarilmay qolgan holatlarda ham biror bir xulosa chiqariladi. Shart bajarilmagan holda, qandaydir harakatni bajarish uchun if operatori Else operatori yordamida kengaytiriladi.

### **If ning Else operatori yordamida kengaytirilgan sintaksisi:**

*if(shart) bajarish\_bloki*

*else bajarish\_bloki1*

Agar shart bajarilgan bo'lsa, ya'ni *shart* true(rost) qiymat qabul qilsa, *bajarish\_bloki* bajariladi, aks holda *bajarish\_bloki1* bajariladi.

**<?php**

**if (shart) – Agar shart rost bo'lsa, u holda**

**{**

**amal**

**agar shart yolg'on bo'lsa, u holda**

**}**

**else**

**{**

**amal**

**}**

**?>**

if operatorini yakka holda yoki else operatori bilan birga ishlatish mumkin. Lekin else operatorini yakka holda ishlatish mumkin emas, faqatgina if operatori bilan birgalikda ishlatiladi.

**7.2- misol.** Berilgan sonni juft-toqligini tekshirish.

```

<?
If ($a%2=0)
{
echo "$a - juft son
";
}
else
{
echo "$a - toq son
";
}
?>

```

**7.3-misol.** Talaba imtihon topshirgan yoki topshirmaganligi tekshiriladi. Agar topshirgan bo'lsa, o'tgan yoki o'tmaganligi tekshiriladi.

```

<?
if ($natija=="")
{
echo "Talaba imtihon topshirmadi";
}
else
{
if ($natija<55) echo "Talaba imtihondan o'tmadi";
if ($natija>55) echo "Talaba imtihondan o'tdi";
}
?>

```

Demak, bu misolda birinchi if operatoridagi shart bajarilmagan holda ikkinchi shart uchun yana if operatori qo‘llanilgan.

### ***Elseif operatori***

If operatorining yana bir kengaytmasidan biri bu elseif operatoridir. Elseif – bu else va if ning kombinatsiyasidir.

#### **Sintaksisi:**

*If(shart) bajarish\_bloki*

*Elseif (shart1) bajarish\_bloki1*

...

*Else bajarish\_blokiN*

Bitta if blokda bir necha marta elseif operatorlari qatnashishi mumkin. Agar if sharti false qiymat qabul qilsa, keyingi elseif shartlar tekshiriladi. Elseif shartlarining birortasi true qiymat qabul qilsa, mos *bajarish\_bloki* bajariladi, aks holda keyingi qadamga o‘tiladi.

7.3 - misolni boshqacharoq shaklda elseif operatori yordamida yozishimiz ham mumkin.

```
<?
if ($natija=="")
echo "Talaba imtihon topshirmadi";
elseif ($natija<55)
echo "Talaba imtihondan o`tmadi";
 elseif ($natija>55)
echo "Talaba imtihondan o`tdi";
?>
```

Shartli o‘tish operatorlarida *shart* o‘rnida bir nechta shartni ham tekshirish mumkin. Bu shartlar VA (AND yoki &&) va YOKI (OR yoki ||) mantiqiy amallari yordamida bog‘lanadi. Quyidagi misolda VA mantiqiy amalini ishlatamiz.

#### **7.4- misol.** Berilgan o‘zgaruvchilar qiymatlarini solishtirish.

```
<?
$a=1;
$b=3;
$c=4;
$d=9;
if ($a!=$b and $c!=$d)
{
echo "O`zgaruvchilar qiymatlari teng emas!";
}
else
{
echo "O`zgaruvchilar qiymatlari teng!";
}
?>
Natija:O`zgaruvchilar qiymatlari teng emas!
```

Bu misolda ikkala shartda ham o‘zgaruvchi qiymatlari bir-biriga teng bo‘lmaganligi, ya’ni shart bajarilgani uchun keyingi qadam bajarildi.

#### **7.5 - misol.** Berilgan o‘zgaruvchilar qiymatlarini solishtirish.

```
<?
$a=1;
$b=3;
$c=9;
$d=9;
if ($a!=$b and $c!=$d)
{
echo "O`zgaruvchilar qiymatlari teng emas! ";
}
```

```

}
else
{
echo "O'zgaruvchilar qiymatlari teng!";
}
?>

```

Natija:O'zgaruvchilar qiymatlari teng!

Bu misolda and amali qatnashganligi uchun ikkala shart ham bajarilgandagina keyingi qadam bajariladi, aks holda else operatoridan keyingi qadam bajarilishi kerak. Misolni tekshiramiz, birinchi shart bajarildi, lekin ikkinchi shart bajarilmadi, shuning uchun else operatoridan keyingi qadam bajarildi.

Agar shartlar orasidagi and mantiqiy amal o'rniga or mantiqiy amalni ishlatsak natija o'zgaradi.

**7.6- misol.** Berilgan o'zgaruvchilar qiymatlarini solishtirish.

```

<?
$a=1;
$b=3;
$c=9;
$d=9;
if ($a!=$b or $c!=$d)
{
echo "O'zgaruvchilar qiymatlari teng emas!";
}
else
{
echo "O'zgaruvchilar qiymatlari teng!";
}
?>

```

Natija:O'zgaruvchilar qiymatlari teng emas!

Bu misolda esa YOKI (or yoki ||) amali qatnashganligi uchun shartlarni birortasi rost qiymat qabul qilsa ham keyingi qadam bajariladi.

Yuqoridagi misollarda o'zgaruvchi qiymatlarini taqqoslash uchun or va and mantiqiy amallari bilan birgalikda == va != amallaridan tashqari boshqa taqqoslash amallarini: <, >, <=, >= va h.k larni ham ishlatish mumkin.

**7.7 - misol.** Talabalarni baholash mezonlari sxemasini tuzib chiqamiz.

- 55 dan past ball to'plagan talabalar – o'zlashtira olmaganlar
- 55 dan yuqori 71 dan past ball to'plagan talabalar – 3 bahoga o'zlashtiruvchilar
- 71 dan yuqori 86 dan past ball to'plagan talabalar – 4 bahoga o'zlashtiruvchilar
- 86 dan yuqori ball to'plagan talabalar – 5 bahoga o'zlashtiruvchilar

Bu misolni PHP kodini if va elseif operatorlari va &&(yoki AND) amali yordamida yozamiz.

```

<?
if ($baho<55)
echo "O'zlashtira olmagan talabalar!";
elseif ($baho>=55 && $baho<71)
echo "3 bahoga o'zlashtirgan talabalar!";
elseif ($baho>=71 && $baho<86)
echo "4 bahoga o'zlashtirgan talabalar!";
elseif ($baho>=86)
echo "5 bahoga o'zlashtirgan talabalar!";

```



?>

E'tibor qarating! elseif operatorini else if shaklida ham yozish mumkin, ikkala holat ham to'g'ri hisoblanadi.

### ***Ichma-ich joylashgan IF-ELSE operatori***

Ichma-ich joylashgan shartlar – bu if shartli o'tish operatori ichida yana bir if shartli o'tish operatorining ishlatilishidir. Bu holda shart tekshirilgandan so'ng, bajarilishi kerak bo'lgan blok ichida biror bir amal emas, balki yana bir shart tekshiriladi.

### **7.8 – misol. Ichma-ich joylashgan shartlarning ishlatilishi**

```
<?
$a=1;
$b=3;
$c=9;
$d=9;
$talaba1="bakalavr";
$talaba2="magistr";
if ($a==$b or $c==$d)
{
echo "O'zgaruvchilar qiymatlari teng!";
if($talaba1!=$talaba2)
{
echo "talaba1 va talaba2 ikki xil yo'nalish talabasi!";
}
}
else
{
echo "O'zgaruvchilar qiymatlari teng emas!";
}
?>
```

Natija:O'zgaruvchilar qiymatlari teng!talaba1 va talaba2 ikki xil yo'nalish talabasi!

Bu misolni tahlil qilamiz. Avval o'zgaruvchi qiymatlari beriladi, so'ng If operatori yordamida shart tekshiriladi, shart bajarilsa, keyingi qadam bajariladi va shu bilan birga ikkinchi shart ham tekshiriladi, ikkinchi shart ham bajarilsa, keyingi qadamga o'tiladi.

### ***switch-case qayta ulovchi operatori***

Shartlarni tekshirish va tekshirish natijalariga ko'ra turli amallarni bajaruvchi operatorlardan biri switch operatoridir. O'zgaruvchi qanday shartga ega ekanligiga qarab, mos turli *bajarish\_bloklari* bajariladi.

switch operatori if va if...elseif operatoriga o'xshash, faqatgina undan farqli tomoni bu operatorida shart ikkitadan ortiq qiymat qabul qiladi. If operatorida *shart* true va false qiymatlarni qabul qiladi. switch operatorida esa *shart* oddiy tip (integer, string yoki double) bo'lgan ixtiyoriy turli qiymatlarni qabul qiladi.

#### **Sintaksisi:**

```
switch (shart yoki o'zgaruvchi) {
 case qiymat1: bajarish_bloki1
 break;
 case qiymat2: bajarish_bloki2
 break;
 case qiymat3: bajarish_bloki3
 break;
```

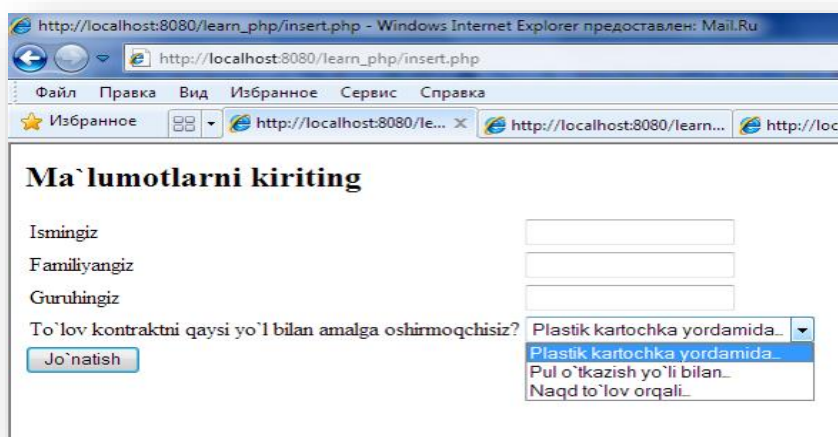
```

...
case qiymat4: bajarish_bloki4
break;
default: birorta ham holat yuzaga kelmagandagi bajarish_bloki
break;
}

```

Bu yerda *shart* case(*qiymat1*, *qiymat2*, *qiymat3*, ... ) dagi qiymatlar bilan solishtiriladi. Agar *shart* yoki *o'zgaruvchi* qiymat1, qiymat2, qiymat3, ... larning qaysidir biri bilan mos tushsa, qiymatdan keyingi ikki nuqtadan birinchi break operatorigacha (agar bo'lsa) yoki switch oxirigacha mos *bajarish\_bloki* bajariladi. Agar *shart* yoki *o'zgaruvchi* qiymat1, qiymat2, qiymat3, ... larning birortasi bilan mos tushmasa default kalit so'zidan keyingi *bajarish\_bloki* bajariladi.

**7.9 – misol.** Talabani nomi, familiyasi, guruhi, to'lov-kontraktini amalga oshirish yo'lini tanlash imkoni bo'lgan maydonlardan iborat forma yaratamiz. Talaba ushbu formani to'ldirib, ma'lumotlarni formani qayta ishlovchi sahifaga jo'natadi. Formani qayta ishlovchi sahifada tanlash operatorlarini ishlatishni ko'rib o'tamiz.



**7.1-rasm.** To'lov-kontraktini amalga oshirish yo'lini tanlash formasi

7.1 –rasmda berilgan formani HTML kodining qismi quyida berilgan.

```

<tr>
 <td>To'lov kontraktini qaysi yo'l bilan amalga oshirmoqchisiz?</td>
 <td><select name="usul">
 <option value = "plastik_kartochka">Plastik kartochka yordamida...
 <option value = "pul_utkazish">Pul o'tkazish yo'li bilan...
 <option value = "naqd">Naqd to'lov orqali...
 </select>
</td>
</tr>

```

yuqorida berilgan kodni qayta ishlash uchun **if-elseif dan** foydalanamiz.

```

<?
$usul="naqd";
if($usul=="plastik_kartochka")
echo "<p>Plastik kartochka yordamida...";
elseif($usul=="pul_utkazish")
echo "<p>Pul o'tkazish yo'li bilan...";
elseif ($usul=="naqd")
echo "<p>Naqd to'lov orqali...";
else
echo "<p>Talaba birorta ham yo'lni tanlamadi ";

```

```
?>
```

Natija:

Naqd to`lov orqali...

### switch-case operatori yordamida:

```
<?
```

```
$usul="naqd";
switch ($usul) {
case "plastik_kartochka":
echo "<p>Plastik kartochka yordamida...";
break;
case "pul_utkazish":
echo "<p>Pul o`tkazish yo`li bilan...";
break;
case "naqd":
echo "<p>Naqd to`lov orqali...";
break;
default:
echo "<p>Talaba birorta ham yo`lni tanlamadi ";
break;
}
?>
```

Natija: Naqd to`lov orqali...

Agar \$usul o`zgaruvchisi birorta ham qiymatga teng bo`lmasa, u holda natija o`zgaradi, ya`ni «Talaba birorta ham yo`lni tanlamadi» yozuvi ekranga chiqaziladi.

Misolning natijasi ikkala holat (if-else va switch-case)da ham bir xil bo`ladi. Faqatgina if-elseif operatorlari yordamida yozilgan vaqtda, har bir qatordagi *shart* tekshiriladi. switch operatori esa boshqacharoq tarzda ishlaydi. switch-case operatorlari ishga tushganda, *shart* {} blok oxirigacha tekshiriladi. Agar break operatori qo`llanilsa, unga duch kelganda, tekshirish to`xtatiladi.

### Takrorlash(sikl) operatorlari

PHP ning faqatgina biror bir shart asosida ishlovchi operatorlari emas, balki takrorlanish amallarini bajaruvchi sikllar ham mavjud. Ular while, do-while, foreach va for.

Takrorlash(sikl) operatori ikki qismdan iborat: tana va bosh qismdan iborat. Tana qismi takrorlashda bajariladigan operatorlarni o`z ichiga oladi. Bosh qism esa tanadagi operatorlarning takroriy bajarilishini ta`minlaydi.

#### **While sikli**

#### **Sintaksisi:**

```
while (shart) {bajarish_bloki}
```

yoki

```
while (shart) bajarish_bloki endwhile;
```

While – oddiy sikl bo`lib, bu yerda *bajarish\_bloki shart* false qiymat qabul qilgunga qadar bajariladi. While siklini if dan farqli tarafi shundaki, if operatorida *shart* tekshirilganda true qiymatni qabul qilsa, u holda keyingi qadam faqatgina bir marotaba bajarilar edi. While siklida esa har bir *shart* bajarilgan holat uchun *bajarish\_bloki* bajariladi, ya`ni bajarilish sikli takrorlanadi. *Shart* har safar sikl boshida tekshiriladi, shuning uchun agar *bajarish\_bloki* amalga oshirilayotgan vaqtda *shartga* o`zgartirish kiritilsa ham sikl iteratsiya oxirigacha, ya`ni *bajarish\_bloking* barcha buyruqlari bajarilguncha to`xtatilmaydi.

## 7.10 - misol. While operatori.

1 dan 5 gacha bo'lgan sonlarni chiqazing.

```
<?
$n=1;
while ($n<=5)
{ echo $n. "
";
$n++;
}
?>
```

Natija:

```
1
2
3
4
5
```

## 7.11 – misol. 1 dan 50 gacha bo'lgan sonlarning yig'indisini hisoblang.

```
<?
$sum=0;
$i=1;
while($i<=50)
{
$sum=$sum+$i;
$i++;
}
echo "1 dan 50 gacha bo'lgan sonlar yig'indisi - $sum";
?>
```

Natija:1 dan 50 gacha bo'lgan sonlar yig'indisi - 1275

### *do...while sikli*

do...while sikli while sikliga juda o'xshash, faqatgina bitta farqli tarafi shundaki, unda *shartning* rostligi sikl boshida emas, balki oxirida tekshiriladi. Shuning uchun do...while siklining *bajarish\_bloki* hech bo'lmaganda bir marotaba bajariladi.

### **Sintaksisi:**

do (*bajarish\_bloki*) while (*shart*);

## 7.12 – misol. do...while operatori

1 dan 10 gacha bo'lgan juft sonlarni chiqazing.

```
<?
$n=1;
do {
if ($n%2==0) echo $n. "
";
$n++;
}
while ($n<=10)
```

?>

Natija:

```
2
4
6
8
10
```

## **FOR sikli**

### **Sintaksisi:**

```
for (amal1; shart; amal2)
 {bajarish_bloki;}

```

*amal1* – faqatgina bir marotaba sikl boshida bajariladi. Odatda bunda hisoblagichning boshlang‘ich qiymati beriladi.

*shart* – shart har bir iteratsiyada tekshiriladi. Agar shart false qiymatni qabul qilsa, u holda iteratsiya to‘xtatiladi. Odatda shartda hisoblagich qiymati bilan berilgan qiymat solishtiriladi.

*amal2* – hisoblagich qiymatini o‘zgartirish uchun ishlatiladi va iteratsiya oxirida bajariladi.

*bajarish\_bloki* – har bir iteratsiyada bir marotaba bajariladi. Odatda bu yerda o‘zida kodlar blokini jamlagan sikl tanasi joylashadi.

*amal1*, *shart*, *amal2* larning har biri bo‘sh bo‘lishi mumkin. Agar *shart* bo‘sh bo‘lsa, bu siklning aniqlanmagan vaqtda bajarilishi kerakligini bildiradi. (bu holda php shartni hamma vaqt rost deb hisoblaydi). Bu sikl foydasizdek ko‘rinishi mumkin, aslida esa bu hollarda siklni break operatorini qo‘llash bilan ham to‘xtatish mumkin.

**7.13 - misol:** 1 dan 10 gacha bo‘lgan toq sonlarni chiqazing.

```
<?
for ($n=1; $n<=10; $n++)
{
 if ($n%2!=0) echo $n. "
";
}
?>
```

Natija:

```
1
3
5
7
9
```

Shu misolni  $n \leq 10$  shartni ishlatmasdan, break operatorini qo‘llab yozish mumkin.

```
<?
for ($n=1;;$n++)
{
 if ($n%2!=0) echo $n. "
";
 if ($n>=10) break;
}
?>
```

Natija:

```
1
3
5
7
9
```

**7.14 - misol:** 1 dan 50 gacha bo‘lgan sonlarning yig‘indisini hisoblang.

```
<?
for($i=1;$i<=50;$i++)
{
 $sum=$sum+$i;
}
echo "1 dan 50 gacha bo‘lgan sonlar yig‘indisi - $sum";
```

?>

Natija:1 dan 50 gacha bo'lgan sonlar yig'indisi - 1275

### ***Foreach***

Bu operator massivlar bilan ishlash uchun mo'ljallangan bo'lib, PHP ning 4 versiyasida kiritilgan. Turli tipdagi massivlar uchun foreach operatorining ikki xil ko'rinishi mavjud.

#### **Sintaksisi:**

***foreach*** (*\$array as \$value*) {*bajarish\_bloki*}

yoki

***foreach*** (*\$array as \$key=> \$value*) {*bajarish\_bloki*}

Birinchi navbatda *\$array* o'zgaruvchisiga berilgan massivning hamma elementlari bo'yicha sikl shakllanadi. Siklning har bir qadamida massivning joriy elementi qiymati *\$value* o'zgaruvchisiga yoziladi va massivning ichki hisoblagichi bir birlikka o'zgaradi(shuning uchun keyingi qadamda massivning keyingi elementi ko'rinadi). *bajarish\_bloki* ichida massivning joriy elementi qiymati *\$value* o'zgaruvchisi yordamida olinadi va *\$array* o'zgaruvchisida nechta element bo'lsa, *bajarish\_bloki* shuncha marta amalga oshiriladi.

***foreach*** massivning ikkinchi tur sintaksisida esa, yuqoridagiga qo'shimcha ravishda, siklning har bir qadamida massivning joriy elementi kaliti *\$key* o'zgaruvchisiga yoziladi va uni ham *bajarish\_bloki* ichida qo'llash mumkin.

foreach o'z ishini boshlaganda, massivning ichki ko'rsatkichi avtomatik tarzda birinchi elementga o'rnatiladi.

#### **7.15 - misol.** Foreach operatori

```
<?php
$names = array("Lola", "Karim", "Kamol");
foreach ($names as $val) {
echo "$val
";
}
?>
```

Natija:Lola

Karim

Kamol

Ikkinchi turi assotsiativ massivlar uchun ishlatiladi.

#### **7.16 - misol.** Foreach operatori

```
<?php
$names = array(
"Lola"=>7,
"Karim"=>9,
"Kamol"=>15);
foreach ($names as $i=>$tartib_nomer)
{
echo "$i ro'yxatda $tartib_nomer nomerida.
";
}
?>
```

Natija:

Lola ro'yxatda 7 nomerida.

Karim ro'yxatda 9 nomerida.

Kamol ro'yxatda 15 nomerida.

### **O'tish operatorlari**

Ayrim hollarda sikl ishini tezda to‘xtatish yoki biror fragmentdan chiqib ketish talab etiladi. Yoki siklning oxirgi instruksiyasidan keyin joylashgan dastur birinchi instruksiyaga boshqaruvni uzatishi kerak. Buning uchun break va continue operatorlaridan foydalaniladi.

### **Break(sikldan chiqish)**

Break operatori joriy siklning (for, foreach, while, do...while, switch) ishini to‘xtatadi. break operatori sonli parametr bilan birga qatnashgan hollarda, sonli parametrqa qarab boshqaruvchi kerakli struktura ishini to‘xtatadi (yoki sikldan chiqadi).

#### **7.17-misol. break operatori**

```
<?php
$i=1;
while ($i) {
 $n = rand(1, 10);
 // 1 dan 10 gacha sonlarni ichidan birini tanlab oladi
 echo "$i:$n";
 // iteratsiya tartibi va tanlangan sonlarni chiqaradi
 if ($n==5) break;
 /* Agar 5 soni tanlangan bo'lsa sikl ishini tugallaydi. Ushbu qatordan keyin baj
 arilishi kerak bo'lgan kodlar ishlamaydi */
 echo "Sikl ishlayapti
";
 $i++;
}
echo "
sikl iteratsiyalar soni $i";
?>
```

Natija:

```
1:10Sikl ishlayapti
2:7Sikl ishlayapti
3:3Sikl ishlayapti
4:4Sikl ishlayapti
5:6Sikl ishlayapti
6:6Sikl ishlayapti
7:8Sikl ishlayapti
8:10Sikl ishlayapti
9:4Sikl ishlayapti
10:4Sikl ishlayapti
11:7Sikl ishlayapti
12:2Sikl ishlayapti
13:2Sikl ishlayapti
14:2Sikl ishlayapti
15:5
sikl iteratsiyalar soni 15
```

rand funksiyasi ishlayotganligi sababli natija har gal turlicha bo‘lishi mumkin.

Agar break operatoridan keyin son ko‘rsatilsa, u holda shu songa teng miqdordagi ushbu operator qatnashgan sikllar ishi to‘xtatiladi. 6.17 - misol bu holatga misol bo‘la olmaydi, chunki unda ichma ich joylashgan sikllar mavjud emas. Skriptga o‘zgartirish kiritamiz:

```
<?php
$i=1;
while ($i) {
 $n = rand(1, 10);
```

```
// 1 dan 10 gacha sonlarni ichidan birini tanlab oladi
switch ($n){
case 5:
echo "
switch (n=$n) dan chiqish";
break 1;
// switch ishini to'xtatamiz (break qatnashgan birinchi siklni)
case 10:
echo "
switch dan chiqish va
while (n=$n)";
break 2;
// switch va while ishini to'xtatamiz (break qatnashgan ikkala siklni)
default:
echo "switch ishlayapti (n=$n), ";
}
echo " while ishlayapti -$i qadam
";
$i++;
}
echo "
 Sikl iteratsiyalar soni $i ";
?>
```

Natija:  
switch ishlayapti (n=4), while ishlayapti -1 qadam  
switch (n=5) dan chiqish while ishlayapti -2 qadam  
switch ishlayapti (n=9), while ishlayapti -3 qadam  
switch dan chiqish va while (n=10)  
Sikl iteratsiyalar soni 4

rand funksiyasi ishlayotganligi sababli natija har gal turlicha bo'lishi mumkin.

### ***Continue(siklni davom ettirish)***

Ayrim hollarda sikl ishini butunlay to'xtatmasdan, siklni yangi iteratsiyadan boshlashga to'g'ri keladi, bu holda continue operatorini qo'llab yangi iteratsiyani boshlash mumkin. continue operatori *bajarish\_bloki* ning istalgan siklidan keyingi instruktsiyalarni ishlashini to'xtatib, yangi doirada bajarishni davom ettirishni ta'minlaydi. Continue operatorini sonli parametr bilan qo'llash mumkin va ushbu parametrda nechta ichma-ich joylashgan boshqaruvchi konstruktsiyaning ishini tugatish kerakligi ko'rsatiladi. Masalan, ikkita ichma-ich joylashgan sikllar ichida Continue 2 buyrug'i berilsa, boshqaruv ikkala sikl ishlashini to'xtatadi.

7.17 - misoldagi break operatorini continuega almashtiramiz va sikl qadamlari sonini 4 ta bilan to'xtatamiz.

```
<?php
$i=1;
while ($i<4) {
$n = rand(1, 10);
// 1 dan 10 gacha sonlarni ichidan birini tanlab oladi
echo "$i:$n";
// iteratsiya generizatsiyalagan sonni chiqaramiz
if ($n==5) {
echo "
 Yangi iteratsiya
";
continue;
/* agar 5 soni generizatsiya qilingan bo'lsa, siklning yangi iteratsiyasini bosh
laymiz $i o'zgarmaydi */
}
echo "sikl ishlayapti
";
$i++;
```



```
}
echo "
".$i ."siklning iteratsiyalar soni";
?>
```

Natija:

1:4sikl ishlayapti

2:4sikl ishlayapti

3:5

Yangi iteratsiya

3:8sikl ishlayapti

4siklning iteratsiyalar soni

rand funksiyasi ishlayotganligi sababli natija har gal turlicha bo'lishi mumkin.

Continue operatori bajarilgandan so'ng sikl ishi to'xtamaydi. Bu yerda sikl hisoblagichi 5 sonini olishda o'zgarmaydi, chunki u continue operatoridan keyin joylashgan. Aslida biz continue yordamida 5 soni paydo bo'lgan holdan qochishga harakat qilamiz. Shuning uchun continue operatorini quyidagi almashtirish bilan yozib ko'ramiz:

```
<?php
$i=1;
while ($i<4) {
$n = rand(1, 10);
// 1 dan 10 gacha bo'lgan sonlarni tanlash
if ($n!=5) {
echo "$i:$n
";
// iteratsiya va generizatsiya qilingan sonni chiqaramiz
$i++;
}
}
?>
```

Natija:

1:2

2:1

3:7

rand funksiyasi ishlayotganligi sababli natija har gal turlicha bo'lishi mumkin.

PHP da continue operatorining qo'llanilishining bitta asosiy tarafi mavjud - u switch konstruksiyalarida xuddi break dagidek ishlaydi. Agar switch sikl ichida joylashgan bo'lsa va yangi iteratsiyani boshlash kerak bo'lsa, continue 2 ni qo'llash kerak.

## Qamrab olish operatorlari

Oddiy, lekin juda ko'p ishlatiladigan operatorlar: require(), include() larni ko'rib o'tamiz. require(), include() operatorlari yordamida PHP sahifaga faylni yuklab olinadi. Fayl o'zida matnlarni, HTML deskriptorlarni, PHP klasslarni, PHP funksiyalarni, PHP operatorlarni jamlagan bo'ladi.

WEB sayt yaratishda faylni yuklab olish nima uchun kerak? Sayt yaratilayotgan vaqtda saytni bloklarga ajratib dastur kodlarini yozish maqsadga muvofiq. Masalan, saytni yuqori qismi bloki, menyu qismi bloki, asosiy qismi bloki, o'ng tomoni bloki va shunga o'xshash. Bloklarga ajratib yozish, saytni biror bir qismiga o'zgartirish kiritishda qulay bo'lib (masalan, yangi menyu qo'shish yoki mavjud menyulardan

birortasini olib tashlash), u holda faqatgina o'sha blokka o'zgartirish kiritiladi xolos. Ana shu bloklar fayllarda saqlanadi va ushbu fayllarni require(), include() operatorlari yordamida yuklab olish mumkin. require(), include() operatorlari bir-biriga o'xshash holatda ishlaydi.

### ***include() operatori***

Include operatori ko'rsatilgan fayldagi barcha kodlarni ochish uchun ishlatiladi va dasturda bu operator necha marta ishtirok etsa, u shuncha marta bajariladi.

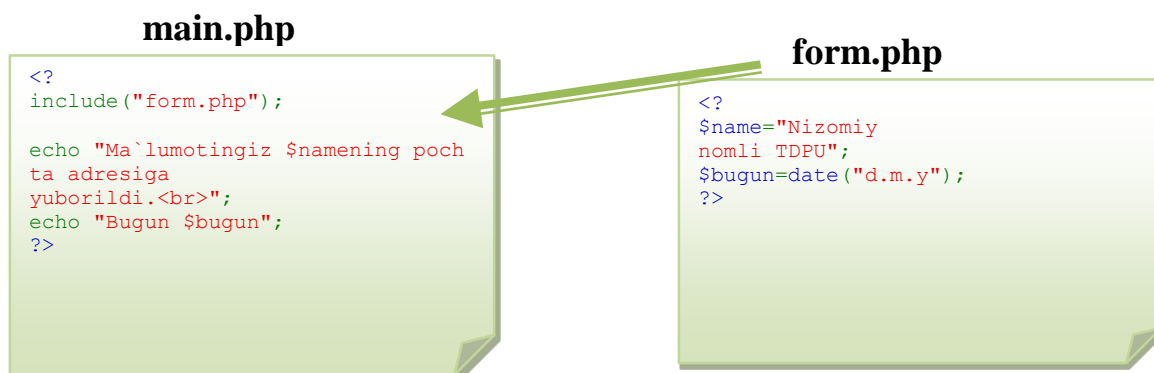
### **Sintaksisi:**

***include*** *'fayl\_nomi'*;yoki

***include*** *\$fayl\_nomi*;yoki

***include*** ("fayl\_nomi");

Masalan, form.php faylida funksiya yoki o'zgaruvchi qiymatlari joylashgan bo'lsin. Asosiy dastur kodini biror qismida ushbu funksiya yoki o'zgaruvchi qiymatlaridan foydalanish kerak bo'lsa include("form.php") ni qo'yamiz. Agar bir necha marta form.php dan foydalanish kerak bo'lsa, u holda asosiy dasturning kerakli qismida bir necha marta qo'llash mumkin. Sahifa bajarilayotgan vaqtda include() operatori include("form.php"); buyrug'i o'rniga form.php faylidagi mavjud kodlarni almashadi.



**7.18 – misol.** include() qamrab olish operatorining ishlatilishi form.php fayli

```
<?
$name="Nizomiy nomli TDPU";
$bugun=date("d.m.y");
?>
```

### **main.php fayli**

```
<?
include("form.php");
echo "Ma'lumotingiz $name ning pochta adresiga yuborildi.
";
echo "Bugun $bugun";
?>
```

Natija shuni ko'rsatadiki main.php ishga tushirilganda sahifa quyidagi ko'rinishga ega bo'ladi.

```
<?
$name="Nizomiy nomli TDPU";
$bugun=date("d.m.y");
echo "Ma'lumotingiz $name ning pochta adresiga yuborildi.
";
echo "Bugun $bugun";
```

?>

Natija:

Ma'lumotingiz Nizomiy nomli TDPU ning pochta adresiga yuborildi.

Bugun 26.01.11

### ***Fayl kengaytmasi va include() operatori***

PHP ning *include()* operatori murojaat etilayotgan faylning kengaytmasiga ahamiyat qaratmaydi. Faylning nomi qulay tarzda nomlanishi mumkin. *include()* operatori yordamida yuklanayotgan fayl PHP faylining bir qismi sifatida bo'ladi va qanday bo'lsa, shunday bajariladi. Qamrab olinayotgan(yuklanayotgan) fayl uchun ixtiyoriy kengaytmani qo'llash mumkin, shulardan biri *.inc*. Agar fayl *.inc* kengaytmasi yoki birorta boshqa standart bo'lmagan kengaytma bilan WEB hujjatlar katalogida joylashgan bo'lsa va foydalanuvchi bu faylni brauzer yordamida ishga tushirsa, u holda fayldagi kodlarni oddiy matn shaklida ko'rish mumkin, ya'ni fayl qayta ishlanmaydi. Shuning uchun murojaat etiladigan fayllarni biror bir katalogda saqlash tavsiya etiladi yoki standart kengaytmadan foydalanish kerak bo'ladi.

PHP operatorlari masalan *page.html* kabi nomlangan fayllarda ishtirok etsa bajarilmaydi. Aksincha, agar *include()* operatori yordamida *page.html* fayli yuklab olinsa, fayl ichida saqlanayotgan PHP operatorlari qayta ishlanadi. Qoidaga ko'ra, PHP faqatgina *.php* kabi kengaytmaga ega fayllardagina operatorlariga murojaat etib, boshqa fayllarni tahlil qiladi.

Demak *form.php* dagi dastur kodi *main.php* faylining bir qismiga ekvivalent. *form.php* faylini PHP kod ekanligini anglatuvchi teglarsiz oddiy matn sifatida yozish mumkinmi? Yo'q, chunki faylni qo'yish vaqtida PHP ni qayta ishlash rejimidan HTML rejimiga o'tadi. Shuning uchun yuklanayotgan fayl ichidagi PHP skripti ko'rinishida qayta ishlanadigan kod mos teglar bilan yozilishi kerak.

Yuklanayotgan fayllarni qidirish quyidagi tartibda amalga oshiriladi:

1. Faylni qidirish avval joriy ish katalogiga tegishli *include\_path* da olib boriladi.
2. Agar fayl topilmasa, u holda qidiruv joriy skript katalogiga tegishli *include\_path* da amalga oshiriladi.
3. PHP sozlovlar faylida aniqlangan *include\_path* parametrda yuklanadigan fayllarni qidirish mumkin bo'lgan kataloglar nomi beriladi.

Masalan, bizning misolda joriy ish katalogi bu - */www/* katalogidir. Asosiy fayl *main.php* fayliga *include/a.php* fayli va navbati bilan *b.php* fayli yuklanayotgan bo'lsa, parser birinchi o'rinda *b.php* faylini */www/* katalogidan qidiradi va agar bunday fayl topilmasa, u holda keyin */www/include/* katalogidan qidiradi.

*include()* yordamida yuklanayotgan fayl tarkibidagi o'zgaruvchilar *include()* joylashgan sohaga bog'liq ravishda ishlatiladi. Yuklanayotgan fayldagi ixtiyoriy o'zgaruvchi yuklanayotgan faylni o'zida ishlatilishi mumkin. Agar *include()* funksiya ichida yuklansa, u holda yuklangan fayl tarkibidagi kod funksiya ichida aniqlanganlar kabi bo'ladi.

*include()* operatori yordamida lokal fayllardan tashqari url – manzillari ko'rsatilgan tashqi fayllarni ham yuklash mumkin. Ushbu imkoniyat PHP sozlov faylidagi *url\_fopen\_wrappers* direktivasida nazoratga olinadi va odatda bu yoqilgan

---

bo‘ladi. Lekin windows uchun PHP ning 4.3.0 versiyasigacha bu imkoniyat mavjud bo‘lmagan.

`include()` bu maxsus konstruksiya, shuning uchun ichki shartli bloklar qo‘llanilishida uni figuralik qavs ichiga olib yoziladi.

```
<?
If ($value=true)
{
include("file1.inc");
}
Else
{
include("file2.inc");
}
?>
```

`include()` operatorini qo‘llashda ikki xil turdagi xatolik yuzaga kelishi mumkin – **yuklash xatoligi** (masalan, yuklash buyrug‘ini o‘zi noto‘g‘ri yozilgan bo‘lsa, ko‘rsatilgan faylni topish mumkin bo‘lmasa va h.) yoki **bajarilish xatoligi** (agar yuklanayotgan faylda xato mavjud bo‘lsa).

#### ***require() operatori***

`require()` va `include()` operatorlari juda o‘xshash bo‘lib, ishlash prinsipiga ko‘ra bir-biridan farq qiladi. `require()` operatori taxminan C++ dagi `# include` kabi ishlaydi. `include()` uchun aytilganlarning barchasi `require` uchun ham o‘rinlidir. `require` shuningdek dasturga bog‘lanishga va biror-bir faylni amalga oshirishga imkon yaratadi. `require` va `include` ning asosiy farqi uning xatolarga qanday reaksiya berishidir. `include` ogohlantiradi va script ishini davom ettiraveradi. `require` da esa xato skript ichida fatal xatoni chiqaradi va uni bajarilishini to‘xtatadi. Savol tug‘ilishi mumkin: u holda `require()` operatori nima uchun qo‘llaniladi, uning `include()` operatoridan qanday farqi bor? Masala shundaki, `require()` operatori `include()` operatoridan bir muncha tezroq ishlaydi.

`require()` operatori shartli operator ichida qo‘llanilsa, yaxshi samara bermaydi. Agar hech bo‘lmaganda ushbu operator qo‘llanilgan qator bajarilmasa, unda yuklangan fayldan bitta qator ham bajarilmaydi. Shuningdek sikllar ham `require()` ga ta’sir qilmaydi. Hatto yuklangan fayl sikl ob’yekti bo‘lsa ham, yuklash o‘z-o‘zidan bir marotabagina amalga oshiriladi.

PHP ning 4.0.2 versiyalarigacha `require()` ning qo‘llanilishi, interpretator albatta yuklanayotgan faylni o‘qishga harakat qilishini anglatgan.

`require()` ni ham `include()` kabi shartli bloklar ichida qo‘llaganda figurali qavs ichiga olib yozish kerak.

```
<?
If ($value=true)
{
require("file1.inc");
}
Else
{
require("file2.inc");
}
?>
```

---

require() operatori shartli bloklar ichida qo'llanilganda, har safar sahifa bajarilganda, barcha fayllar yuklab olinadi, lekin ulardan faqatgina bittasi \$value o'zgaruvchisi qiymatiga qarab bajariladi.

Agar require() operatorining o'rnida include() operatori qo'llanilsa, u holda \$value o'zgaruvchisi qiymatiga qarab faqat bitta fayl yuklab olinadi va bajariladi.

### **require() operatorining Web –sayt shablonlari uchun qo'llanilishi**

Web –sayt sahifalari bir xil ko'rinishdan iborat bo'lsa, u holda PHP dan foydalanib require() operatori yordamida sahifaga shablon va standart elementlarni qo'shish mumkin.



### **Savol va topshiriqlar:**

1. Razryadli operatorlar nima?
2. Mantiqiy operatorlar nima?
3. Tanlash operatorlari nima?
4. Sikl operatorlari?



## **4.9. PHPda funktsiyalar**

Funksiya ko'pgina dasturlash tillarida ishlatiladi. Funksiya dasturda batafsil aniqlangan topshiriqni alohida bajarish uchun yozilgan buyruqlar to'plami(kodlar)ni belgilash uchun ishlatiladi. Funksiya har gal topshiriqni bajarish uchun buyruqlar to'plamini o'qish va qayta ishlatish imkoniyatini yaratadi. Funktsiyalar ikki xil bo'ladi: **PHP da mavjud funktsiyalar(ichki funktsiyalari)** va **foydalanuvchi tomonidan aniqlangan funktsiyalar**.

**PHP ning ichki funktsiyalari** fayllar bilan ishlash, ma'lumotlar bazasini ishlatish, grafik tasvirlar yaratish va boshqa serverlarga bog'lanish uchun imkon beradi.

Ba'zi hollarda dastur tuzish vaqtida ichki funktsiyalar imkoniyatlaridan tashqari biror-bir narsani bajarishga to'g'ri keladi. Bu hollarda dasturchilarning imkoniyatlari chegaralanmagan, ya'ni ichki funktsiyalardan tashqari ular o'zlari funksiya yaratishlari mumkin va bu funktsiyalar **foydalanuvchi tomonidan aniqlangan funktsiyalar** deyiladi. Foydalanuvchi tomonidan aniqlangan funksiya o'zida nimani mujassamlashtirganini ko'rib o'tsak. Funksiya dasturlashda ham matematikaga o'xshab bir nechta parametrlarni ularning qiymatiga aks etishini ifodalaydi, ya'ni funksiya ishining natijasi bo'lgan har bir berilgan parametrqa qandaydir qiymat qaytaradi. Topshiriqni bajarish uchun yaratilgan kod biror-bir topshiriqni bajaruvchi maxsus, mantiqiy va mavjud funktsiyalar va amallar kombinatsiyasidan iborat. Bu kod funksiya e'lon qilinganda berilib, sahifani bir nechta qismida yoki bir nechta sahifalarida funktsiyaga murojaat etilganda bajariladi.

### **Foydalanuvchi tomonidan aniqlangan funktsiyalar**

Foydalanuvchi tomonidan aniqlangan funktsiyani e'lon qilish yangi funktsiyani yaratadi. Funksiya keyinchalik foydalanuvchi tomonidan aniqlangan funksiya

---

ekanligi haqida foydalanuvchiga xabar berish va dasturda PHP sintaksisini tahlil qilish uchun **function** kalit soʻzi yordamida eʼlon qilinadi. Funksiya **function** kalit soʻzi bilan boshlanib, funksiya nomini, talab qilinayotgan parametrlar va funksiyaga har gal murojaat etilganda bajariladigan kodlarni oʻz ichiga oladi.

Umumiy holda foydalanuvchi tomonidan aniqlangan funksiyalarni eʼlon qilish quyidagi koʻrinishda boʻladi.

**Sintaksisi:**

```
function funksiya_nomi(parametr1, parametr2, ..., parametrN)
{
 harakatlar_maydoni
 return qaytariladigan_funksiya_qiymati;
}
```

Bu yerda

*funksiya\_nomi* - funksiya nomi;

*harakatlar\_maydoni* - funksiya tanasi, yaʼni biror-bir topshiriqni bajaruvchi kodlar;

*parametr1, ..., parametrN* - lar esa almashuvchi maʼlumotlar.

Funksiya quyidagi talablarga javob berishi kerak:

1. funksiya nomi va parametr nomlari PHP ning nomlash qoidalariga mos tushishi kerak (krillcha belgilarni qoʻllamaslik maqsadga muvofiq). Funksiya nomlari registrga bogʻliq emas.

2. funksiya parametrlari oʻzgaruvchilar hisoblanadi. Shuning uchun ularning har biri oldidan \$ belgisi ishlatilishi kerak. Parametrlar roʻyxatida ...(uch nuqta) ishlatish mumkin emas.

3. *harakatlar\_maydoni* soʻzi oʻrnida biror-bir vazifani bajaruvchi PHP kod turishi kerak. (*parametr* larga bogʻliq boʻlishi shart emas).

PHP ning ichki funksiyalari barcha sahifalarda ishlay oladi. Foydalanuvchi tomonidan aniqlangan funksiyalar esa faqatgina ular eʼlon qilingan sahifalarda ishlaydi. Bundan kelib chiqadiki, tez-tez ishlatiladigan foydalanuvchi tomonidan aniqlangan funksiyalarni alohida faylda saqlash kerak. Va bu funksiyalarga qaysi sahifada murojaat etilsa, shu sahifani boshida require() yoki include() operatorlari yordamida funksiyalar joylashgan fayl yuklab olinadi.

Funksiyani ichki kodi figurali qavs ichida berilib, talab qilingan topshiriqni bajaradi.

Agar funksiya ichida PHP dan chiqib, HTML matnga oʻtish kerak boʻlsa, sahifani ixtiyoriy joyida PHP deskriptori quyidagicha yopilib, ochiladi:

```
<?
function my_funksiya()
{
?>
Mening funksiyam elon qilindi
<?
}
?>
```

## Registr va funksiya nomi

Funksiya nomini berilishi oʻzgaruvchi nomini berishdan farq qiladi. Oʻzgaruvchi nomi registrga sezgir shuning uchun \$Nomi va \$nomi – turli xil

---

o'zgaruvchilar, lekin `Nomi()` va `nomi()` - bitta funksiya. Funksiya nomi registrga ahamiyatsiz, shuning uchun `funksiya_nomi()`, `Funksiya_Nomi()` yoki `FUNKSIYA_NOMI()` murojaatlari bitta natijani beradi. Funksiya nomlarida harflarni ixtiyoriy (katta, kichik) holda ishlatish mumkin.

Funksiya nomi qisqa va tushunarli bo'lishi kerak. Agar funksiya sahifani yuqori qismi uchun yaratilayotgan bo'lsa, u holda `pageheader()` yoki `page_header()` kabi nomlanishi mumkin.

Funksiya nomiga quyidagicha chegaralar qo'yiladi:

- funksiya **PHP ning ichki funksiyalari** nomi bilan nomlana olmaydi;
- funksiya nomi faqatgina harflardan, raqamlardan va chiziqcha belgisidan iborat bo'la oladi.

- funksiya nomi raqam bilan boshlana olmaydi.

Ko'pgina dasturlash tillarida funksiya nomi qaytarilishiga ruxsat bor. Bu funksiyani qaytarilishi (peregruzka) deb ataladi. Faqatgina PHP qaytarilgan funksiyani qo'llamaydi, shuning uchun funksiya nomi **PHP ning ichki funksiyalari** nomlari va avval **foydalanuvchi tomonidan aniqlangan funksiya** nomlari bilan bir xil bo'lmasligi kerak.

Avval aytib o'tganimizdek, **PHP ning ichki funksiyalari** barcha PHP sahifalarda ishlashi mumkin, **foydalanuvchi tomonidan aniqlangan funksiya** esa faqatgina e'lon qilingan sahifalardagina ishlay oladi. Bu shuni anglatadiki, e'lon qilinadigan funksiya nomlari boshqa fayllarda takroran ishlatilishi mumkin, lekin bu chalg'ishga (adashishga) olib keladi. Shuning uchun bu holatni qo'llamagan ma'qul.

Funksiya nomlari quyidagicha bo'lishi mumkin:

`Name();`

`Name2();`

`Name_tree();`

`_namefour;`

Funksiya nomlari quyidagicha bo'lishi mumkin emas:

`5name()`

`Name_six;`

`Fopen()`

(`Fopen()` funksiya nomi **PHP ning ichki funksiyasi**)

## Funksiyaga murojaat etish

*Funksiyaga qanday murojaat etiladi?* Avval funksiya nomi beriladi va qavslar ichida uning parametrlari qiymatlari ro'yxati kiritiladi.

**Sintaksisi:**

`funksiya_nomi();`

bu kod biror-bir parametрни talab qilmaydigan `funksiya_nomi` nomli funksiyaga murojaat etadi. Funksiyaga bu tarzda murojaat etish funksiya qaytaradigan barcha qiymatlarni inkor etadi. Ko'pgina funksiyalar shu tarzda beriladi.

Misol uchun, **PHP ning ichki funksiyalaridan** biri `phpinfo()` funksiyasi. Bu funksiya o'rnatilgan PHP versiyasi va u haqida ma'lumot olish, PHP va serverning turli o'zgaruvchilari qiymatlarini ko'rsatish uchun ishlatiladi. Bu funksiya hech qanday parametr qabul qilmaydi va uning qaytaradigan qiymati ham inkor etiladi. Shuning uchun `phpinfo()` ning berilishi quyidagicha:

```
phpinfo();
```

**Foydalanuvchi tomonidan aniqlangan funksiyaga** murojaat etish **PHP ning ichki funksiyalariga** murojaat etish kabi bo'ladi.

**8.1 - misol.** Oddiy funksiyani e'lon qilamiz:

```
<?
function my_function() // my_function() funksiyasi e`lon qilindi
{
Echo "Mening funksiyam elon qilindi";
}
my_function(); // my_function() funksiyasiga murojaat etildi
?>
```

Natija: Mening funksiyam elon qilindi

### ***Funksiyaga qachon murojaat etiladi?***

PHP ning 4 versiyasigacha funksiya u aniqlangandan so'ng murojaat etilardi, ya'ni dasturlash qatorida

```
function funksiya_nomi() {...}
 blokidan so'ng.
```

Lekin PHP 4 da esa bunday talab yo'q. Hammasi interpretator olingan kodni qanday qayta ishlashiga bog'liq. Faqat shartli aniqlanadigan funksiyalar (shartli operator va boshqa funksiyalar ishida) bundan holi.

**8.2-misol.** Shartli o'tish operatori ichida funksiyaning ishlatilishi

```
<?
$make = true;
/* bu yerda Make_event()ga murojaat etish mumkin emas;
chunki u bu yerda mavjud emas,
lekin Save_info()ga murojaat etish mumkin */
Save_info("Vohid", "Fozilov", "Men Informatika fanini tanladim");
if ($make)
{
// Make_event() funksiyasining aniqlanishi
function Make_event()
{
 echo "<p> Tarix fanini o`rganishni xoxlayman";
}
}
// endi Make_event()ga murojaat etish mumkin
Make_event();
// Save_info funksiyasining aniqlanishi
function Save_info($first, $last, $message)
{
 echo "
$message
";
 echo "Ismi sharifi: ". $first . " ". $last . "
";
}
// Save info ga bu yerda ham murojaat etish mumkin
Save_info("Laziz", "Abdullayev", "Men Falsafa fanini tanladim");
?>
```

Natija:

Men Informatika fanini tanladim

Ismi sharifi: Vohid Fozilov



Tarix fanini o`rganishni xoxlayman  
Men Falsafa fanini tanladim  
Ismi sharifi: Laziz Abdullayev

Agar funksiya dasturda bir marta aniqlangan bo'lsa, uni ikkinchi marotaba qayta aniqlash yoki o'chirib tashlash mumkin emas. Funktsiyalar registrga bog'liq emasligiga qaramasdan ularga qanday nom bilan aniqlangan bo'lsa, o'sha nom bilan murojaat etish lozim.

### Parametrlar (yoki Argumentlar)

Ko'pgina funksiyalarda o'z topshiriqlarini bajarish uchun bir yoki undan ortiq *parametrlarni* uzatilishi talab etiladi. *Parametrlar* ma'lumotlarni, qiymatlarni funksiyaga yetkazish imkonini beradi. Har bir *parametr* o'zida o'zgaruvchini va konstantani mujassamlashtiradi. Ular () qavs ichida funksiya nomidan so'ng beriladi.

*funksiya\_nomi*("parametr1 uchun qiymat", "parametr2 uchun qiymat",...);

Bu yerda "parametr1 uchun qiymat", "parametr2 uchun qiymat",... lar o'rnida parametrlarning qiymatlari beriladi. **Masalan,**

*funksiya\_nomi*(2);

*funksiya\_nomi*(7.993);

*funksiya\_nomi*(\$variable);

Oxirgi berilgan funksiyada \$variable o'zgaruvchisi ixtiyoriy tipda, xatto massiv ham bo'lishi mumkin.

*Parametr* qiymatlari yoki qiymatlarni o'zida jamlagan o'zgaruvchi nomlari turli yo'l bilan uzatiladi. Funksiyaga ma'lumotlarni funksiya *parametrlari* yordamida uch xil yo'l bilan uzatish mumkin:

- 1) qiymati bo'yicha (odatga ko'ra ishlatiladi)
- 2) jo'natmasi (ssilka) bo'yicha
- 3) ko'rinmasligi bo'yicha

Funksiya *parametrlarini* oddiy yo'l bilan uzatish **qiymati bo'yicha** uzatish deb ataladi. *Parametrlarni* uzatishda uzatilayotgan qiymatni o'zida jamlagan yangi o'zgaruvchi yaratiladi. U mavjud (ichki) o'zgaruvchini nusxasi. Bu qiymat ixtiyoriy tarzda o'zgarishi mumkin, lekin mavjud (ichki) o'zgaruvchini qiymati funksiya tashqarisida o'zgarmay qoladi. *Parametr* funksiyaga qiymati bo'yicha uzatilayotganda parametrning funksiya ichida o'zgarishi uning funksiya tashqarisidagi qiymatiga ta'sir qilmaydi.

Qiymatni ketma-ket oshirib boruvchi funksiyani yaratamiz.

### 8.3 - misol. Parametrlarni qiymati bo'yicha uzatish

```
<?
Function increment($value, $a=1)
{
 $value=$value+$a;
}
$value=15;
increment($value);
Echo $value;
?>
```

Natija:15

**Natija** 15 chiqadi, ya'ni \$value o'zgaruvchini qiymati o'zgarmaydi.

Demak, bu kod biz talab etgan vazifani bajarmadi. Bu esa *harakatlar\_maydoni* bilan bog‘liq. Kod vazifasiga ko‘ra \$value o‘zgaruvchisini yaratadi, so‘ng dastur increment() funksiyasiga murojaat etadi. Funksiya ichida \$value o‘zgaruvchisi unga murojaat etishiga ko‘ra yaratiladi, so‘ng uning qiymatiga 1 ko‘shiladi va funksiya ichida \$value qiymati 16 ga teng bo‘ladi, lekin natijani ekranga chiqarishga buyruq berilmaganligi va funksiya tashqarisida \$value o‘zgaruvchisi global maydonda aniqlangan boshqa o‘zgaruvchi bo‘lganligi uchun o‘zgarmay qoladi, shuning uchun natija 15 ga teng bo‘ladi.

Bu muammoni hal qilish yo‘llaridan biri bu - \$value o‘zgaruvchisini funksiyada global sifatida e‘lon qilish kerak, lekin bu shuni anglatadiki, bu funksiyani ishlatishda uning qiymati oshirilishi kerak bo‘lgan o‘zgaruvchisini \$value deb nomlash talab etiladi. Bundan ham maqbul yo‘l sifatida **jo‘natma bo‘yicha** uzatishni aytishimiz mumkin. Funksiya *parametrlarini* o‘zgartirishga ruxsat berish uchun ularni **jo‘natma (ssilka) bo‘yicha** uzatish kerak.

**Jo‘natma bo‘yicha parametrlar** uzatilayotganda yangi qiymat yaratish bilan birga funksiya jo‘natmadan ichki o‘zgaruvchilarni qabul qiladi. Bu jo‘natma \$ dollar belgisi bilan boshlanuvchi o‘zgaruvchi nomi yoki biror bir ixtiyoriy o‘zgaruvchi. Farqi shundaki *parametr* olayotgan qiymatni ichki o‘zgaruvchiga uzatadi.

*Parametrlarni* jo‘natma yordamida uzatishda funksiya e‘lon qilinayotgan vaqtda *parametr* oldidan (&) ampersand belgisi qo‘yiladi. Funksiyaga murojaat etishda esa hech qanday o‘zgartirish kiritish shart emas. **8.2 - misolni** kamchiligini to‘g‘rilaymiz.

### 8.3 – misol. Parametrlarni jo‘natma bo‘yicha uzatish.

```
<?
Function increment (&$value, $a=1)
{
$value=$value+$a;
}
$b=15;
Echo $b.'
```

Funksiya oddiy yo‘l bilan uzatilgan bo‘lsa, unda *parametrlar* qiymatini aniqlash mumkin. Oddiy yo‘l bo‘yicha uzatilgan qiymat o‘zgaruvchi va klass vakili yoki boshqa funksiyadan chaqirilgan bo‘lmasdan konstanta bo‘lishi kerak.

*Parametrlarni ko‘rinmasligi bo‘yicha* uzatish.

Bizda berilgan *parametrga* bog‘liq holda o‘zgaradigan informatsion xabar yaratuvchi funksiya mavjud. Agar funksiya *parametri* qiymati berilmagan bo‘lsa, u holda qiymat o‘rnida “Dekanat” yozuvi qo‘llaniladi.

### 8.4 - misol. Parametrlarni ko‘rinmasligi bo‘yicha uzatish

```
<?php
function xabar($sign="Dekanat"){
// bu yerda sign «Dekanat» mazmuniga ega
echo "Keyingi majlis ertaga bo`ladi.";
echo "$sign .
";
}
```

```
xabar();
// funksiyani parametrsiz chaqiramiz. Vu holda yozuv -Dekanat
xabar("Hurmat bilan Afzalov");
// natija «Hurmat bilan Afzalov» bo`ladi
?>
```

Natija:

Keyingi majlis ertaga bo`ladi.Dekanat .

Keyingi majlis ertaga bo`ladi.Hurmat bilan Afzalov .

Bir nechta *parametrlar* bilan berilgan funktsiyani oddiy yo`l bilan e`lon qilinayotganda, qiymati bilan berilgan parametrlar hamma parametrlardan keyin yozilishi kerak. Aks holda, funksiyaga murojaat etilayotganda bu parametrlar tashlab yuboriladi va natija xato chiqadi.

**8.5 - misol.** Talaba tomonidan biror-bir maqola katalogga qo`shilishi kerak. Maqolani katalogga qo`shish uchun sahifada maqolaning nomi, muallifi va qisqa izohi kabi tavsiflari kiritiladi. Agar talaba maqolaning muallifini kiritmagan bo`lsa, u holda maqola muallifini Komilov Islom deb dasturning o`zi kiritilsin.

```
<?php
function Add_maqola($nomi, $izohi,
 $muallifi="Komilov Islom"){
 echo "Katalogga $nomi maqolasini qo`shamiz. ";
 echo "
Muallif: $muallifi";
 echo "
Qisqa izoh: ";
 echo "$izohi
";
}
Add_maqola("Informatika asoslari",
 "Bu maqola informatika bo`yicha ...",
 "Yuldashev Umid");
Add_maqola("Ma`lumotlar bazasi",
 "Bu maqola ma`lumotlar bazasi haqida ...");
?>
```

Natija:Katalogga Informatika asoslari maqolasini qo`shamiz.  
Muallif: Yuldashev Umid  
Qisqa izoh: Bu maqola informatika bo`yicha ...

---

Katalogga Ma`lumotlar bazasi maqolasini qo`shamiz.

Muallif: Komilov Islom

Qisqa izoh: Bu maqola ma`lumotlar bazasi haqida ...

Agar biz quyidagicha yozsak, xatolik yuzaga keladi, chunki qiymatga ega parametr oxirida emas, balki parametrlar ro`yxatini boshida yozilgan:

```
<?php
function Add_maqola($muallifi="Komilov Islom", $nomi, $izohi)
{
 echo "Katalogga $nomi maqolasini qo`shamiz. ";
 echo "
Muallif: $muallifi";
 echo "
Qisqa izoh: ";
 echo "$izohi
";
}
Add_maqola("Ma`lumotlar bazasi", "Bu maqola ma`lumotlar bazasi haqida ...");
?>
```

Natija:

**Warning:** Missing argument 3 for Add\_maqola(), called in  
D:\xampplite\htdocs\learn\_php\learn.php on line 9 and defined in

**D:\xampplite\htdocs\learn\_php\learn.php on line 2**

Katalogga Bu maqola ma`lumotlar bazasi haqida ... maqolasini qo`shamiz.

### ***O'zgaruvchi parametrlar ruyxati***

PHP4 da funksiyaga nechta *parametr* bo'yicha murojaat etishni oldindan bilmasdan turib funksiyani yaratish mumkin. Bunday funksiyani yozish uchun hech qanday maxsus sintaksislar talab etilmaydi. Hammasi **PHP ning ichki funksiyalari** yordamida bajariladi **func\_num\_args()**, **func\_get\_arg()**, **func\_get\_args()**.

**func\_num\_args()** funksiyasi foydalanuvchi tomonidan aniqlangan funksiyada beriladigan parametrlar sonini qaytaradi. Bu funksiya faqat foydalanuvchi tomonidan aniqlangan funksiya ichida qo'llanilishi mumkin. Agar **func\_num\_args()** funksiyasi foydalanuvchi tomonidan aniqlangan funksiyaning tashqarisida qo'llanilsa, unda interpreter bu haqida xabar beradi.

#### **8.6 - misol.** func\_num\_args() funksiyasining ishlatilishi

```
<?php
function DataCheck(){
 $n = func_num_args();
 echo "Funksiya parametrlari soni $n ta
";
}
DataCheck();
// «Funksiya parametrlari soni 0 ta » qatorini chiqaradi
DataCheck(1,2,3,6);
// «Funksiya parametrlari soni 3 ta » qatorini chiqaradi
?>
```

Natija:

Funksiya parametrlari soni 0 ta

Funksiya parametrlari soni 4 ta

**func\_get\_arg(butun parametr\_nomeri)** funksiyasi **foydalanuvchi tomonidan aniqlangan funksiya** parametrlari ruyxatidan raqamlar ketma - ketligi bo'yicha *parametr\_nomeri* ga mos berilgan parametrni qaytaradi. **Foydalanuvchi tomonidan aniqlangan funksiya** parametri noldan boshlab sanaladi. **func\_num\_args()** dagidek bu funksiya ham faqat **foydalanuvchi tomonidan aniqlangan funksiya ichida** qo'llanilishi mumkin.

*parametr\_nomeri* **foydalanuvchi tomonidan aniqlangan funksiyada** berilgan parametrlar sonidan katta bo'lishi mumkin emas. Aks holda **func\_get\_arg()** funksiyasi false ni qaytaradi va mos xabarni ishlab chiqadi.

**8.7 - misol.** Parametrlarda berilgan ma'lumotlar tipini tekshirish uchun funksiya yarataylik.

```
<?
Function DataCheck(){
 $check =true;
 $n = func_num_args();
 // funksiyaga berilgan parametrlar soni
 /* birinchi berilgan parametr butun son ekanligini tekshiramiz */
 if ($n>=1)
if (!is_int(func_get_arg(0))) $check = false;
 /* ikkinchi berilgan parametr satr ekanligini tekshiramiz*/
 if ($n>=2)
 if (!is_string(func_get_arg(1))) $check = false;
return $check;
}
if (DataCheck(a123,"text"))
 echo "Berilganlar shartlarni qoniqtiradi
";
```

```

else echo "Berilganlar shartlarni qoniqtirmaydi
";
if (DataCheck(324))
 echo " Berilganlar shartlarni qoniqtiradi
";
else echo "Berilganlar shartlarni qoniqtirmaydi";
?>

```

Natija:

Berilganlar shartlarni qoniqtirmaydi  
Berilganlar shartlarni qoniqtiradi

**func\_get\_args()** funksiyasi **foydalanuvchi tomonidan aniqlangan funksiyada** berilgan parametrlar ro'yxatidan tuzilgan massivni qaytaradi. Massivning har bir elementi funksiyada berilgan parametrغا mos tushadi. Agar funksiya **foydalanuvchi tomonidan aniqlangan funksiyada** qo'llanilmayotgan bo'lsa, u holda xabar ishlab chiqiladi.

8.7 - misolga ushbu funksiyani qo'llab o'zgartirish kiritamiz. Funksiyaga berilayotgan har bir toq parametr butun son ekanligini tekshiramiz.

```

<?
Function DataCheck(){
 $check =true;
 $n = func_num_args();
 // funksiyaga berilgan parametrlar soni
 $args = func_get_args();
 // funksiya parametri massivi
 for ($i=1;$i<=$n;$i++){
 $v = $args[$i];
 if ($i % 2 == 0){
 if (!is_int($v)) $check = false;
 // juft parametr butun sonligini tekshiramiz
 }
 }
 return $check;
}
if (DataCheck("text", 324))
 echo "Berilgan juft parametr juft son
";
else echo "Berilganlar shartlarni qoniqtirmaydi";
?>

```

Natija:Berilganlar shartlarni qoniqtirmaydi

Ko'rib turganimizdek **func\_num\_args()**, **func\_get\_arg()**, **func\_get\_args()** funksiyalar kombinatsiyasi **foydalanuvchi tomonidan aniqlangan funksiyalar** o'zgaruvchi parametrlar ro'yxatiga ega bo'lishi uchun qo'llaniladi. Bu funksiyalar faqat PHP 4 da berilgan. PHP 3 da shunga o'xshash natijaga erishish uchun massiv funksiyasi qo'llaniladi.

**8.8 - misol.** Funksiyadagi har bir juft parametr butun son ekanligini tekshirish uchun skriptni qo'yidagicha yozish mumkin:

```

<?
Function DataCheck($params){
 $check =true;
 $n = count($params);
 // funksiyaga berilgan parametrlar soni
 for ($i=0;$i<$n;$i++){
 $v = $params[$i];
 if ($i % 2 !== 0){
 // juft parametr butun son ekanligini tekshiramiz
 }
 if (!is_int($v)) $check = false;
 }
}

```

```
return $check;
}
if (DataCheck("text", 324))
 echo "Berilgan har bir juft parametr butun son
";
else echo "berilganlar shartlarni qoniqtirmaydi";
?>
```

Natija: Berilgan har bir juft parametr butun son

## Harakatlar maydoni

Oldingi mavzularda o‘zgaruvchilarni biror bir fayl ichida e’lon qilib, sahifaga require() yoki include() operatorlari yordamida yuklab olish haqida gapirib o‘tgan edik, funksiyada esa o‘zgaruvchilarni funksiyani o‘zida e’lon qilinadi.

O‘zgaruvchilarning xarakat diapazoni o‘zgaruvchilarni ko‘rinarli va belgilanganligiga qarab boshqaradi. Turli dasturlash tillarida o‘zgaruvchilarni harakat diapazoni o‘rnatilgan turlicha qoidalarga bo‘ysunadi.

PHP da esa quyidagicha qoida mavjud:

- funksiya ichida aniqlangan o‘zgaruvchilar ular e’lon qilingan operatoridan boshlab funksiyani yakunlovchi qavsgacha bo‘lgan maydonda harakatlanadi. Bu maydon funksiya maydoni, o‘zgaruvchilar esa – **lokal o‘zgaruvchilar** deb ataladi.

- funksiya tashqarisida aniqlangan o‘zgaruvchilar ular e’lon qilingan operatoridan boshlab, funksiya ichida emas, balki fayl oxirigacha bo‘lgan maydonda harakatlanadi. Bu maydon global maydon, o‘zgaruvchilar esa – **global o‘zgaruvchilar** deyiladi.

- require() va include() operatorlari yordamida yuklab olingan fayllarda e’lon qilingan o‘zgaruvchilar funksiya harakat maydonida qabul qilinmaydi. Agar operatorlar funksiya ichida ishlatilsa, funksiya maydonida qabul qilinadi. Agar operatorlar funksiya tashqarisida ishlatilsa, global maydonda qabul qilinadi.

- funksiya ichida aniqlangan yoki ishlatilayotgan o‘zgaruvchilarni global maydonda harakatlana olishi uchun global kalit so‘zi yordamida belgilash(ko‘rsatish) mumkin, ya’ni o‘zgaruvchilarni funksiya ichida qo‘llash uchun bu o‘zgaruvchilar global o‘zgaruvchi sifatida e’lon qilinishi kerak. Buning uchun funksiya tanasida global kalit so‘zidan keyin ularning nomlari keltirilishi kerak.

global \$var1, \$var2, \$var3, ...,

- o‘zgaruvchilar unset(\$variable\_name) funksiyasi yordamida o‘chiriladi. Agar o‘zgaruvchi o‘chirilgan bo‘lsa, harakat maydonida boshqa ishlatilmaydi.

## Funksiya ichida o‘zgaruvchilarni ishlatilishi

fn() funksiyasi ichida \$var o‘zgaruvchisi e’lon qilindi. Bu o‘zgaruvchi funksiya ichida e’lon qilingani uchun e’lon qilingan joydan boshlab funksiya yakunigacha funksiya harakatlanish maydonida ishlay oladi. Funksiyadan tashqarida esa \$var o‘zgaruvchisiga murojaat etildi, ya’ni yangi \$var o‘zgaruvchisi yaratildi. Bu yangi o‘zgaruvchi global harakatlanish maydonida fayl oxirigacha ishlay oladi. Agar bu o‘zgaruvchini echo operatori yordamida ekranga chiqazsak, hech qanday qiymat chiqmaydi. Chunki bu o‘zgaruvchi funksiya tashqarisida qiymatga ega emas.

```
<?
Function fn()
{
$var="mazmun";
}
```

```
Echo $var;
```

```
?>
```

Quyidagi misol bunga teskari. O‘zgaruvchini funksiya tashqarisida e‘lon qilamiz, so‘ngra funksiya ichida uni ishlatamiz.

### 8.9 - misol. O‘zgaruvchilarni funksiya ichida va tashqarisida ishlatilishi

```
<?
Function fn()
{
Echo "funksiya ichida, \$var=".$var."
";
$var="mazmun2";
Echo "funksiya ichida, \$var=".$var."
";
}
$var="mazmun1";
Fn();
Echo "funksiya tashqarisida, \$var=".$var."
";
?>
```

Natija:

funksiya ichida, \$var=

funksiya ichida, \$var=mazmun2

funksiya tashqarisida, \$var=mazmun1

funksiya bu holda ham bajarilmadi, chunki o‘zgaruvchi funksiya ichiga chaqirilmadi, birinchi bajarilishi lozim bo‘lgan operator \$var=”mazmun1”;

bu yerda global maydonda harakatlana oladigan va qiymati mazmun1 ga teng \$var o‘zgaruvchisi yaratiladi.

Keyingi bajarilayotgan operator – funksiyaga murojaat etish. Funksiya ichidagi satrlar ketma-ket bajariladi. Funksiyani birinchi satri \$var o‘zgaruvchisiga murojaat etadi. Bu satr bajarilganda yaratilgan \$var o‘zgaruvchisini ko‘ra olmaydi, shuning uchun funksiya maydonida yangi o‘zgaruvchi yaratadi va uni ekranga chiqazadi. Natijada 1 satr ekranga chiqadi.

Funksiya ichidagi keyingi satr \$var o‘zgaruvchisini «mazmun2» ga tenglashtiradi. Harakat funksiya ichida bo‘layotganligi uchun bu satr global emas, balki lokal o‘zgaruvchi \$var qiymatini o‘zgartiradi. Ikkinchi satr esa bu o‘zgarishni ekranga chiqazadi.

Shundan so‘ng funksiya ishini tugatadi va sahifani oxirgi satri bajariladi. Echo operatori yordamida global o‘zgaruvchini o‘zgarmaganligini ko‘rish mumkin.

### **Global o‘zgaruvchilar**

Agar funksiya ichida yaratilgan o‘zgaruvchi global bo‘lishi kerak bo‘lsa, uning oldidan **global** kalit so‘zi ishlatiladi.

### 8.10 - misol. Global o‘zgaruvchilarni ishlatilishi

```
<?
Function fn()
{
global $var;
$var="contents";
Echo "funksiya ichida, \$var=".$var."
";
}
Fn();
Echo "funksiya tashqarisida, \$var=".$var."
";
?>
```

Natija:

funksiya ichida, \$var=contents

funksiya tashqarisida, \$var=contents

---

Bu misolda, \$var global o'zgaruvchi sifatida e'lon qilingan bo'lib, o'zgaruvchi maydonda global \$var satri bajarilgan joydan boshlab aniqlangan. Shuning uchun o'zgaruvchi funksiya tashqarisida ham ishlatilishi mumkin.

Funksiyani unga murojaat etilayotgan joydan yuqorida yoki pastda ham e'lon qilish mumkin. (E'tibor bering, funksiya harakatlanish maydoni o'zgaruvchi harakatlanish maydonidan tubdan farq qiladi!)

### **Statik o'zgaruvchilar**

O'zgaruvchilarni funksiya ichida, funksiyadan chiqqandan keyin ham ular qiymatini saqlab qolinadigan qilib qo'llash uchun ularni statik o'zgaruvchilar sifatida e'lon qilish kerak. Agar dastur bajarilishi funksiya qiymati chegaralaridan chiqsa, u holda statik o'zgaruvchilar faqat funksiya ichida ko'rinadi. Bunday o'zgaruvchilarni e'lon qilish uchun static kalit so'zi qo'llaniladi:

```
static $var1, $var2;
```

Statik o'zgaruvchiga istalgan qiymat biriktirilishi mumkin, lekin jo'natmani unga biriktirib bo'lmaydi.

### **8.11 - misol. Statik o'zgaruvchilarni ishlatilishi**

```
<?
function Test_s(){
static $a = 1;
// Ifoda va jo`natmani biriktirish mumkin emas
$a = $a*2;
echo $a;
}
Test_s(); // 2 ni chiqaradi
echo $a; // $a faqatgina funksiya ichida qiymatga ega bo`lganligi uchun
// hech narsani chiqarmaydi
Test_s(); // funksiya ichida $a=2, shuning uchun
// funksiya ishi natijasi 4 soni bo`ladi
?>
Natija:24
```

### **Qaytariluvchi mazmunlar**

Yuqorida misol tariqasida keltirilgan funksiyalarning hammasi qandaydir harakatlarni bajaradi va natijani ekranga chiqazadi. Ba'zi hollarda istalgan funksiya ishining natijasi sifatida qandaydir qiymatni ekranga chiqazish emas, balki faqatgina natijani qaytarish kerak bo'ladi. Bu **return** buyrug'i yordamida amalga oshiriladi. Qaytariladigan qiymat har-xil toifada bo'lishi mumkin, shu qatorda ro'yxat ham, ob'yekt ham. Interpretator funksiya tanasida **return** buyrug'ini ko'rgach, uni bajarilishini darrov to'xtatadi va qaysi qatorda funksiyaga murojaat etilgan bo'lsa o'sha qatorga o'tadi.

### **8.12 – misol. Qaytariluvchi mazmunlar**

```
<?
function kvadrat($a)
{
$nat=$a*$a;
return $nat;
}
$y=kvadrat($a);
$c=$y/10+25;
echo $c;
?>
Natija:25
```



## Qaytariluvchi jo‘natmalar

O‘z ishining natijasi sifatida funksiya qandaydir o‘zgaruvchi(global, statik yoki parametrlardan biriga)ga jo‘natmani ham qaytarishi mumkin. Qaytariluvchi jo‘natmalar funksiyani qaysi o‘zgaruvchiga jo‘natma biriktirilganini aniqlash kerak bo‘lgan hollarda juda ham qo‘l keladi. Funksiyadan jo‘natmani olish uchun uni e‘lon qilinishida va har safar funksiyaga murojaat etilayotganda nomi oldidan ampersand belgisi (&) ni qo‘yish kerak. Odatda funksiya jo‘natma bilan berilgan bo‘lsa, jo‘natmani qandaydir global o‘zgaruvchiga (yoki uning qismi – jo‘natmani global massiv elementiga), statik o‘zgaruvchiga (yoki uning qismiga) yoki parametrlardan biriga qaytaradi.

### 8.13 - misol. Qaytariluvchi jo‘natmalar

```
<?
$a = 'juft son';
$b = 'toq son';
function &ref($par){
global $a, $b;
 if ($par % 2 == 0) return $a;
 else return $b;
}
$c=4;
$var =&ref($c);
echo $c. " - ". $var. "
";
// 4 - juft son chiqaradi
$c = 11;
$var =&ref($c);
echo $c. " - ". $var. "
";
// 11 - toq son chiqaradi
?>
Natija:
4 - juft son
11 - toq son
```

## O‘zgaruvchili funksiyalar

PHP o‘zgaruvchili funksiyalar konsepsiyasini qo‘llaydi. Bu, agar o‘zgaruvchi qavslar bilan tugagan bo‘lsa, u holda PHP xuddi shunday nomli funksiyani qidirishini va uni bajarishga harakat qilishini bildiradi.

### 8.14 – misol. O‘zgaruvchili funksiyalarning ishlatilishi

```
<?
/* ikki oddiy funksiyani yaratamiz:
Add_sign - qatorga yozuv qo`shadi va
Show_matn - matn qatorini chiqaradi */
function Add_sign($string, $sign="Hurmat bilan Komil")
{
 echo $string ." ".$sign;
}
function Show_matn()
{
 echo "Xabarni pochta orqali jo`natish
";
}
$func = "Show_matn";
// Show_matn funksiyasi nomiga teng qiymatli o`zgaruvchini yaratamiz
$func();
// Bu Show_matn funksiyasiga murojaat etadi
$func = "Add_sign";
// Add_sign funksiyasi nomiga teng qiymatli o`zgaruvchini yaratamiz
$func("Hammaga salom
");
// Bu "Hammaga salom" parametrli
```

```
// Add_sign funksiyasiga murojaat etadi
?>
Natija:
Xabarni pochta orqali jo`natish
Hammaga salom
Hurmat bilan Komil
```

Bu misolda Show\_matn funksiyasi faqat satrni chiqarish uchun xizmat qiladi. Maxsus echo() funksiyasi mavjudligi tufayli nima uchun alohida funksiya yaratdik degan savol tug'iladi. Gap shundaki echo(), print(), unset(), unclude() kabi funksiyalarni o'zgaruvchi sifatida qo'llab bo'lmaydi. Shuning uchun biz echo "TEXT" ; ni quyidagidek yozsak, u holda interpretator quyidagi xatoni chiqazadi

```
<?
$func = "echo";
$func("TEXT");
?>
Natija:
Fatal error: Call to undefined function echo() in D:\xampplite\htdocs\learn_php\learn.php on
line 3
```

Shuning uchun yuqoridagi sanab o'tilgan funksiyalarni o'zgaruvchi sifatida qo'llash uchun yuqoridagi misoldagidek yangi funksiya yaratib olish kerak.

### Ichki funksiyalar

PHP – Web sahifalarni yaratishda ishlatiladigan kengaytirilgan funksiyalar kutubxonasiga ega texnologiya. PHP ning 100 dan ortiq ichki (standart) funksiyalari mavjud. Bu funksiyalarni ko'pchiligi qandaydir ma'lumotlarni qabul qilib va natija sifatida biror-bir ma'lumot chiqazadi.

Funksiyani e'lon qilishni oddiy formasi sifatida PHP ning ichki funksiyalaridan biri *date()* funksiyasini ko'ramiz.

#### Sintaksisi:

*date(parametr);*

E'tibor qarating, funksiyaga berilayotgan ma'lumot qavs ichiga olib yozilmoqda. Bu qiymat funksiya *argumenti* yoki *parametri* deb ataladi. *Parametrlar* kiruvchi qiymatlar hisoblanib, funksiya ekranga *parametrlarga* bog'liq aniq bir natijani chiqazadi.

Bu yerda *parametr* o'rnida quyidagilar ko'rsatilishi mumkin:

- a — "am" yoki "pm"
- A — "AM" yoki "PM"
- d — oy kuni, raqamli, 2 ta raqamdan (birinchi o'rinda nol turadi)
- D — hafta kuni, matnli, 3 harfdan iborat; "Fri" va h.
- F — oy, matnli, to'liq; "January" va h.
- h — soat, raqamli, 12 soatli formatda
- H — soat, raqamli, 24 soatli formatda
- i — minut, raqamli
- j — oy kuni, raqamli, boshida keluvchi nolsiz
- l ('L' katta harfli) — hafta kuni, matnli, to'liq; "Friday" va h.
- m — oy, raqamli
- M — oy, matnli, 3 harfdan iborat; "Jan" va h.

- s — sekund, raqamli
- S — inglizcha tartibdagi suffiks, matnli, 2 belgidan iborat; "th", "nd" va h.
- U — sekundda, asr boshidan boshlab
- Y — yil, raqamli, 4 ta raqamdan iborat
- w — hafta kuni, raqamli, 0 yakshanbani bildiradi
- y — yil, raqamli, 2 ta raqamdan iborat
- z — yil kuni, raqamli; "299" va h.

### 8.15 - misol. date() funksiyasining ishlatilishi

```
<?
$d=date(d);
$m=date(m);
$y=date(y);
Echo "Bugun: $d.$m.$y";
?>
Natija:Bugun: 27.01.11
```

**isset() funksiyasi** o'zgaruvchi mavjud yoki mavjud emasligini tekshiradi, agar o'zgaruvchi mavjud bo'lsa, funksiya true(1) qiymatini, aks holda false(0) ni qaytaradi.

#### Sintaksisi:

**isset(o'zgaruvchi);**

### 8.16 - misol. Isset() funksiyasining ishlatilishi

```
<?
$c=34;
If(isset($c))
{
Echo "O`zgaruvchi mavjud va uning qiymati $c ga teng";
}
Else
{ Echo "O`zgaruvchi mavjud emas";
}
?>
Natija:O`zgaruvchi mavjud va uning qiymati 34 ga teng
```

**Unset() funksiyasi** o'zgaruvchilarni o'chirish uchun ishlatiladi.

#### Sintaksisi:

**unset(o'zgaruvchi);**

### 8.17 - misol. Unset() funksiyasining ishlatilishi

```
<?
$k=34;
Echo "funksiyaga murojaat etishdan avval k=". $k."
";
unset($k);
Echo "funksiyaga murojaat etgandan so`ng k=". $k."
";
?>
Natija:
funksiyaga murojaat etishdan avval k=34
funksiyaga murojaat etgandan so`ng k=
```

**mail() funksiyasi** xabarni elektron pochta orqali berilgan manzilga jo'natadi.

#### Sintaksisi:

*boolean mail (string oluvchi, string mavzu, string xabar [, string qo'shimcha\_sarlavxa])*

«mavzu» parametrida, xabar mavzusi beriladi. «xabar» parametri esa xabar

---

matnidan tarkib topadi, shart bo'lmagan parametr - «*qo'shimcha\_sarlavha*» da xabar bilan jo'natiladigan qo'shimcha ma'lumot belgilanadi (masalan, HTML formatlash atributlari). **Misol uchun,**

```
<?
$email = "yourname@yourserver.com";
$mavzu = "Bu mavzu";
$xabar = "Bu xabar";
$sarlavha = "From: somebody@somesite.com";
mail ($email, $mavzu, $xabar, $sarlavha);
?>
```

Biz PHP ning ichki funksiyalarini ba'zilar bilan, ya'ni echo(), print(), include(), require(), date(), isset(), unset(), mail() kabilar bilan tanishdik. Aslida yuqorida sanab o'tilgan funksiyalarning ichidan date() dan tashqari hammasi til konstruksiyalari hisoblanadi. Ular PHPning yadrosi hisoblanadi va hech qanday sozlash va modullarni talab etmaydi. Lekin shunday funksiyalar borki ular bilan ishlash uchun har-xil kutubxonalarini o'rnatish va mos modulni ulash kerak. Masalan, ma'lumotlar bazasi bilan ishlovchi MySQL funksiyasi bilan ishlash uchun PHP ni shu kengaytmada komplekslash kerak. Oxirgi vaqtlarda tarqalgan kengaytmalar va mos ravishda ularning funksiyalari ular bilan ishlashda interpretatorning hech qanday qo'shimcha sozlashlarisiz ishlashda PHPning boshlang'ich tarkibiga kiritilgan.

Shunga o'xshash PHP ning massivlar bilan, fayllar bilan, ma'lumotlar bazasi bilan ishlovchi ichki funksiyalari mavjud. Biz bu funksiyalar bilan keyingi bo'limlarda tanishamiz.



### Savol va topshiriqlar:

1. PHPda funktsiya nima?
2. PHPda rekursiya nima?
3. O'zgaruvchilarning ko'rinish sohasi nima?
4. Funktsiya parametri nima?
5. Funktsiya argumenti nima?



## 4.10. PHPda massivlar

Oldingi ma'ruzalarda massiv tipidagi ma'lumotlar haqida gapirib o'tganmiz. Massivni ikki xil usul bilan yaratish mumkin:

### **Array()**

1. **array() funksiyasi** yordamida

```
$array_name= array ([value1, value2,...]);
```

Assotsiativ massivlarda

```
$array_name = array («key1»=>«value1»,
«key2»=>«value2»);
```

2. **to'g'ridan-to'g'ri qiymat berish yo'li** bilan

```
$array_name []=value1;
Assotsiativ massivlarda
$array_name[«key1»] = value1;
```

Bizdan ma'lumotlar bazasidan o'chirilishi kerak bo'lgan fanlar ro'yxatini saqlash talab etilgan bo'lsin. Ularni massiv ko'rinishida saqlash mumkin: bu yerda kalit - fan identifikatori (uni unikal nomeri), qiymat esa – fan nomi. Bu massivni quyidagicha yaratish mumkin:

```
<?
$del_items = array("10"=>"Fizika",
 "12"=>"Infomatika");
$del_items["13"] = "Php da dasturlash";
// massivga element qo'shamiz
>
```

### ***Ko'p o'lchovli massivlar***

Endi jadvaldagi ma'lumotlarni barchasini massiv shaklida aniqlaymiz va quyidagicha PHP kodini yozamiz:

```
<?
$talabalar=array(

array("1","Islomov", "Ikrom", 150, 1, "Andijon", "3/12/1992", "TDPU"),
 array("3","Fozilov", "Fozil", 200, 3, "Qashqadaryo", "1/12/1990", "T
DPU"),
 array("6","Salimov", "Vohid", 150, 4, "Namangan", "7/06/1989", "NamD
U"),
 array("10","Karimov", "Botir", 100, 2, "Jizzax", "8/12/1991", "TDPU"
)
);
>
```

Bundan ko'rinib turibdiki, talabalar massivi to'rtta massivdan tashkil topgan, ya'ni ko'p o'lchovli massiv. Bir o'lchovli massivga murojaat etishda massiv nomi va uning indeksidan foydalangan edik. Ikki o'lchovli massivda esa, har bir elementning ikkita indeksi bo'ladi – satri va ustuni. Bu massivning tarkibini ko'rish quyidagicha amalga oshiriladi:

```
echo "|".$talabalar[0][0]."|".$talabalar[0][1]."|".$talabalar[0][2]."|".$talabalar[0][3].
"|".$talabalar[0][4]."|".$talabalar[0][5]."|".$talabalar[0][6]."|".$talabalar[0][7]."|
;
echo "|".$talabalar[1][0]."|".$talabalar[1][1]."|".$talabalar[1][2]."|".$talabalar[1][3].
"|".$talabalar[1][4]."|".$talabalar[1][5]."|".$talabalar[1][6]."|".$talabalar[1][7]."|
;
echo "|".$talabalar[2][0]."|".$talabalar[2][1]."|".$talabalar[2][2]."|".$talabalar[2][3].
"|".$talabalar[2][4]."|".$talabalar[2][5]."|".$talabalar[2][6]."|".$talabalar[2][7]."|
;
```

Yoki ushbu natijani olish uchun sikldan foydalanish ham mumkin:

```
<?
$talabalar=array(
 array("1","Islomov", "Ikrom", 150, 1, "Andijon", "3/12/1992", "TDPU"),
 array("3","Fozilov", "Fozil", 200, 3, "Qashqadaryo", "1/12/1990", "TDPU"),
 array("6","Salimov", "Vohid", 150, 4, "Namangan", "7/06/1989", "NamDU"),
 array("10","Karimov", "Botir", 100, 2, "Jizzax", "8/12/1991", "TDPU")
);
for($row=0;$row<4;$row++)
{
 for($column=0; $column<8; $column++)
 {
 echo "|".$talabalar[$row][$column];
```

```

 }
 echo "|
";
}
?>

```

Natija:

```

|1|Islomov|Ikrom|150|1|Andijon|3/12/1992|TDPU|
|3|Fozilov|Fozil|200|3|Qashqadaryo|1/12/1990|TDPU|
|6|Salimov|Vohid|150|4|Namangan|7/06/1989|NamDU|
|10|Karimov|Botir|100|2|Jizzax|8/12/1991|TDPU|

```

Jadvaldagi ma'lumotlarning massivini ustun nomlarini ishlatib ham hosil qilish mumkin. Bunday massiv assotsiativ massiv bo'ladi.

```

$stalabalar=array(
 array(stud_id=>"1",surname=>"Islomov", name=>"Ikrom", stipend=>150, kurs=>1,
 city=>"Andijon", birthday=>"3/12/1992", univ_id=>"TDPU"),
 array(stud_id=>"3",surname=>"Fozilov", name=>"Fozil", stipend=>200, kurs=>3,
 city=>"Qashqadaryo", birthday=>"1/12/1990", univ_id=>"TDPU"),
 array(stud_id=>"6",surname=>"Salimov", name=>"Vohid", stipend=>150, kurs=>4,
 city=>"Namangan", birthday=>"7/06/1989", univ_id=>"NamDU"),
 array(stud_id=>"10",surname=>"Karimov", name=>"Botir", stipend=>100, kurs=>2,
 city=>"Jizzax", birthday=>"8/12/1991", univ_id=>"TDPU")
);

```

Bunday massivning tarkibini ko'rish uchun ham sikllardan foydalanish mumkin:

```

<?
$stalabalar=array(
 array(stud_id=>"1",surname=>"Islomov", name=>"Ikrom", stipend=>150, kurs=>1,
 city=>"Andijon", birthday=>"3/12/1992", univ_id=>"TDPU"),
 array(stud_id=>"3",surname=>"Fozilov", name=>"Fozil", stipend=>200, kurs=>3,
 city=>"Qashqadaryo", birthday=>"1/12/1990", univ_id=>"TDPU"),
 array(stud_id=>"6",surname=>"Salimov", name=>"Vohid", stipend=>150, kurs=>4,
 city=>"Namangan", birthday=>"7/06/1989", univ_id=>"NamDU"),
 array(stud_id=>"10",surname=>"Karimov", name=>"Botir", stipend=>100, kurs=>2,
 city=>"Jizzax", birthday=>"8/12/1991", univ_id=>"TDPU")
);
for($row=0;$row<4;$row++)
{
 echo "|".$stalabalar[$row]["stud_id"]."|".$stalabalar[$row]["surname"]."|".
 $stalabalar[$row]["name"]."|".$stalabalar[$row]["stipend"]."|".$stalabalar[$row]["kurs"]."|".
 $stalabalar[$row]["city"]."|".$stalabalar[$row]["birthday"]."|".
 $stalabalar[$row]["univ_id"]."|
";
}
?>

```

Natija: |1|Islomov|Ikrom|150|1|Andijon|3/12/1992|TDPU|  
3	Fozilov	Fozil	200	3	Qashqadaryo	1/12/1990	TDPU
6	Salimov	Vohid	150	4	Namangan	7/06/1989	NamDU
10	Karimov	Botir	100	2	Jizzax	8/12/1991	TDPU

## Massivlar ustida amallar

Massiv – bu ma'lumot tipi, ma'lumotlarni bu tipi ustida amallar bajarish mumkin. Massivlarni qo'shish va taqqoslash mumkin.

Massivlarni qo'shish uchun «+» operatoridan foydalanamiz. Umuman olganda bu amal massivlarni birlashtirish amali. Bizda ikkita \$a va \$b massivlari mavjud, ularni qo'shish(birlashtirish) natijasi \$a va \$b massivlardan tashkil topgan \$c massivga teng bo'ladi. Agar mos kalitlar uchrashsa, u holda natijada birinchi massiv(ya'ni, \$a)ning elementlari olinadi.

Massivlarni teng yoki teng emas, ekvivalent yoki ekvivalent emasligini tekshirish yordamida taqqoslash mumkin.

Barcha massiv elementlarining kalit/qiyamat juftliklari mos tushsa bu massivlar teng deyiladi.

---

Solishtirilayotgan massivlarning kalit/qiymat juftliklari bir xil tartibda yozilsa va teng bo'lsa, u holda bu massivlar ekvivalent massivlar deyiladi.

PHP da teng qiymati - «= =» belgisi bilan, ekvivalent esa - «= = =» belgisi bilan belgilanadi.

### 10.2 - misol. Massivlarni taqqoslash

```
<?
$a = array("i"=>"Informatika", "m"=>"Matematika");
$b = array("m"=>"Matematika", "i"=>"Informatika");
if ($a == $b) echo "Massivlar teng va";
else echo " Massivlar teng emas va";
if ($a === $b) echo " ekvivalent";
else echo " ekvivalent emas";
// " Massivlar teng va ekvivalent emas "
// natijasini olamiz
?>
Natija:Massivlar teng va ekvivalent emas
```

## Massivlar bilan ishlovchi funksiyalar

### *each()* funksiyasi

Bu funksiya massivning joriy elementini qaytaradi (chiqazadi) va keyingi elementini joriy deb belgilaydi.

#### Sintaksisi:

*each*(massiv);

*each()* funksiyasiga while sikli ichida murojaat etilganda, u navbatma-navbat massivning har bir elementini qaytaradi va massiv oxiriga yetgach o'z ishini to'xtatadi.

```
<?
$stalabalar=array(stud_id=>"1",surname=>"Islomov", name=>"Ikrom", stipend=>150, kurs=>1, city=>"Andijon", birthday=>"3/12/1992", univ_id=>"TDPU");
while($element = each($stalabalar))
{echo $element["key"];
echo " - ";
echo $element["value"] ;
echo "
";
}
?>
Natija: stud_id - 1
surname - Islomov
name - Ikrom
stipend - 150
kurs - 1
city - Andijon
birthday - 3/12/1992
univ_id - TDPU
```

Bu misolda \$element o'zgaruvchisi ham massiv elementlarini qabul qilib olayotganligi uchun massiv bo'ladi. *each()* funksiyasiga murojaat etganda, u sakkizta qiymatdan va sakkizta indeksdan iborat kataklardan tashkil topgan massivni beradi. key va 0 katagi joriy elementning kaliti, value va 1 katagi – joriy elementning qiymatlaridan tarkib topgan. Kataklardan qaysi birini tanlash ahamiyatsiz, ushbu misolda raqamlangan kataklarni emas, nomli kataklarni ishlatdik, ya'ni assotsiativ massivdan foydalandik.

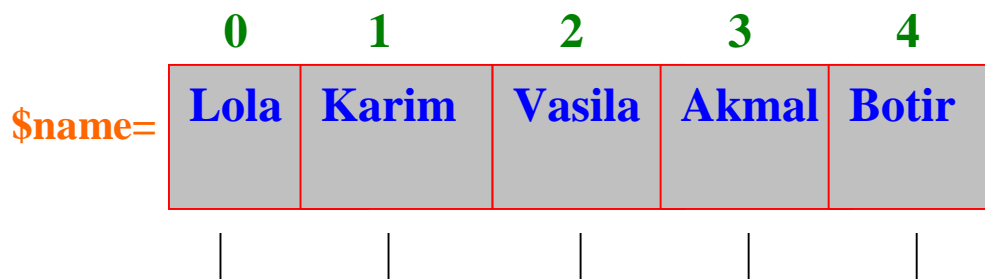
### *List()* funksiyasi

List() funksiyasi array() ga o'xshash bo'lib, uning asosiy vazifasi – bir vaqtning o'zida massiv ichidagi bir nechta qiymatni bir nechta o'zgaruvchiga tenglashtiradi. Bu funksiya berilgan massivni bir nechta qiymatlarga ajratadi.

**Sintaksisi:**

`void list(o'zgaruvchi1 [,o'zgaruvchi2 , ...])`

**list()** - ma'lumotlar bazasidan yoki fayldan ma'lumotlarni o'zgaruvchiga o'qib olish uchun juda qulay funksiya.



`list($talaba1, $talaba2, $talaba3, $talaba4, $talaba5)=$name`

Bu yerda \$talaba1= Lola, \$talaba2=Karim, \$talaba3= Vasila, \$talaba4= Akmal, \$talaba5= Botir

**10.3 - misol. list() funksiyasini qo'llash.** 10.1 – jadvaldan foydalanamiz.

```
<?
$talabalar=array("1", "Islomov", "Ikrom", "150", "1", "Andijon", "3/12/1992", "TDPU");
list($stud_id, $surname, $name, $stipend, $kurs, $city, $birthday, $univ_id)=$talabalar;
echo $stud_id.
;
echo $surname.
;
echo $name.
;
echo $stipend.
;
echo $kurs.
;
echo $city.
;
echo $birthday.
;
echo $univ_id;
?>
```

Natija:  
1  
Islomov  
Ikrom  
150  
1  
Andijon  
3/12/1992  
TDPU

**range() funksiyasi**

**range()** funksiyasi quyi va yuqori chegarasi aniq bo'lgan butun sonlar intervalini oson va tez yaratish imkonini beradi. **range()** funksiyasi ko'rsatilgan intervaldagi barcha butun sonlardan tashkil topgan massivni chiqazadi.

**Sintaksisi:**

`array range(int quyi_chegara, int yuqori_chegara).`

**10.5 – misol. range() funksiyasini qo'llash**

```
<?
$lot= range(0,9);
// $lot=array(0,1,2,3,4,5,6,7,8,9).
?>
```



---

**range()** funksiyasida 0 dan 9 gacha bo'lgan interval ko'rsatilganligi uchun \$slot massivi ko'rsatilgan intervaldagi butun sonlardan iborat bo'ladi.

### **Massiv elementlarini hisoblash: sizeof(),count(),array\_count\_values()**

#### **sizeof() funksiyasi**

Massivlarni ustida yana bir muhim amalni bajarish mumkin - massiv elementlari sonini aniqlash. **sizeof()** funksiyasi - massiv elementlari sonini hisoblaydi. Yana bir shunga o'xshash funksiya **sizeof()** ning kengaytirilgan formasi - bu **count()** funksiyasidir.

#### **count() funksiyasi**

**count()** funksiyasi **sizeof()** ga o'xshash bo'lib, massivdagi qiymatlarning sonini hisoblaydi. Umuman olganda, bu funksiya o'zgaruvchining elementlari sonini hisoblaydi.

Agar bu funksiyani massiv o'zgaruvchidan boshqa biror bir o'zgaruvchi uchun qo'llanilsa, natija 1 chiqadi. NULL tipidagi o'zgaruvchi esa bundan mustasno, ya'ni **count(NULL)** 0 ga teng.

Bu funksiya ko'p o'lchovli massiv elementlari sonini hisoblash uchun qo'llanilganda, qo'shimcha parametr **COUNT\_RECURSIVE** ishlatiladi.

#### **Sintaksisi:**

**count(o'zgaruvchi, [COUNT\_RECURSIVE]);**

#### **10.6 - misol. count() funksiyasini qo'llash**

```
<?
$del_items = array("langs" => array("10"=>"Python", "12"=>"Lisp"),
 "other"=>"Infomatika");
echo count($del_items) . "
";
// 2 chiqadi
echo count($del_items,COUNT_RECURSIVE);
// 4 chiqadi
?>
```

Natija:

2  
4

#### **array\_count\_values() funksiyasi**

#### **Sintaksisi:**

**array\_count\_values(o'zgaruvchi);**

**array\_count\_values(\$ar)** funksiyasi \$ar massivida unikal qiymat necha marta uchraganligi sonini chiqazadi.

#### **10.7 - misol. array\_count\_values() funksiyasini qo'llash**

```
<?
$ar=array(4,5,1,2,3,1,2,2,1,2,1,1);
$nat=array_count_values($ar);
echo '<table>';
echo '<tr><td>qiyamatlar</td><td>soni</td></tr>';
for($i=1;$i<=5;$i++){
echo '<tr><td>'. $i. '</td><td>'. $nat[$i]. '</td></tr>';
}
echo '</table>';
?>
```

Natija:

qiyamatlar soni

---

1	5
2	4
3	1
4	1
5	1

Natija shuni ko'rsatadiki, \$ar massivida 1 - 5 marta, 2 - 4 marta, 3,4,5 qiymatlari 1 marta uchragan.

**Massiv elementlari bilan ishlash:** *In\_array()*, *array\_search()*, *array\_keys()*, *array\_values()*, *array\_unique()*

***In\_array()* funksiyasi**

**in\_array()** - massiv elementlarini qidiruvchi funksiya. Berilgan massivda qidirilayotgan qiymat mavjudligini tekshiradi.

**Sintaksisi:**

**in\_array(«qidirilayotgan\_qiymat», «massiv», [«tip chegarasi»]);**

Agar uchinchi parametr true sifatida berilgan bo'lsa, u holda massivdan faqatgina qiymati bo'yicha emas, balki tipi bo'yicha ham mos elementni qidiradi. Agar qidirilayotgan qiymat - satr bo'lsa, u holda taqqoslanayotganda registrga ahamiyat beradi.

*In\_array()* funksiyasida qidirilayotgan qiymat massiv shaklida ham bo'lishi mumkin. Faqatgina bu imkoniyat PHP 4.2.0. versiyasidan boshlab qo'shilgan.

***array\_search()* funksiyasi**

Ushbu funksiya ham **in\_array()** funksiyasi kabi massiv elementlarini qidiradi. **array\_search()** funksiyasining **in\_array()** funksiyasidan farqli tarafi agar element topilgan bo'lsa, kalitini chiqazadi, aks holda false ni qaytaradi.

**Sintaksisi:**

**array\_search(«qidirilayotgan\_qiymat», «massiv», [«tip chegarasi»]);**

Satrlarni solishtirish registrga sezgir, agar uchinchi parametr ko'rsatilgan bo'lsa, u holda qiymat tipini ham solishtiradi. PHP ning 4.2.0 versiyasigacha, agar qidirilayotgan qiymat topilmasa, funksiya xato yoki bo'sh NULL qiymatni qaytaradi.

Bu funksiya *in\_array()* funksiyasiga nisbatan funksionalroq, chunki bu funksiya yordamida qidirilayotgan element nafaqat massivda mavjudligini, balki u massivni qayerida joylashgani haqida ham bilishimiz mumkin. Agar qidirilayotgan element massivda bir nechta bo'lsa, nima qilamiz? Bu holatda **array\_search()** funksiyasi birinchi topilgan element kalitini chiqazadi. Barcha element kalitlarini olish uchun esa **array\_keys()** funksiyasidan foydalaniladi.

***array\_keys* funksiyasi**

---

*array\_keys()* funksiyasi massivning barcha kalitlarini chiqazadi. Qo‘shimcha parametr sifatida berilgan aniq qiymat yordamida element kalitlari ro‘yxatini olish mumkin.

**Sintaksisi:**

*array\_keys(«massiv», [«qidirish uchun qiymat»])*

*array\_keys()* funksiyasi massivning sonli kalitlari kabi satrli kalitlari qiymatlarini ham chiqazadi. Bunda qiymatlardan hosil bo‘lgan yangi massiv sonli indeksli massiv ko‘rinishida bo‘ladi.

*array\_keys()*, funksiyasi avvalgi ikkita funksiya kabi registrga sezgir, masalan, FIZIKA elementini massivdan topmaydi. *array\_keys()* funksiyasi PHP4 versiyasidan boshlab qo‘shilgan.

Massiv elementlarining kalitlarini olish uchun ishlatiladigan funksiyalarni ko‘rib chiqdik, endi massiv elementlarining qiymatlarini olish uchun ishlatiladigan funksiyalar bilan tanishamiz.

*array\_values()* funksiyasi

**Sintaksisi:**

*array\_values(massiv);*

Bu funksiya massivda berilgan barcha element qiymatlarini yangi massivga butun sonlar bilan indekslab yozadi, shuning uchun massiv elementlarining barcha kalitlari yo‘qoladi va faqatgina qiymatlari qoladi.

*array\_unique* funksiyasi

**Sintaksisi:**

*array\_unique(massiv);*

Bu funksiya massivdan takrorlanuvchi qiymatlarni o‘chirib, yangi massiv hosil qiladi. Natijada, bir nechta bir xil qiymat va ularning kalitlari o‘rniga bitta qiymat qoladi. Uni kaliti qanday bo‘ladi? Yangi massivga kalit bir xil elementlarning bir nechta kalitlari ichidan qanday tanlab olinadi? Massivning barcha elementlari satrlarga akslantiriladi va saralanadi. Shundan so‘ng saralovchi har bir qiymatning birinchi kalitini eslab qoladi, boshqa kalitlar esa inkor etiladi.

## Massivlarni saralash

Turli xil masalalarni yechishda tez-tez massiv ko‘rinishida saqlanayotgan ma’lumotlarni saralash zarur bo‘lib turadi. Agar bu masalani Si da yechmoqchi

---

bo'lsak, u holda o'ntagacha kod satrlari yozish kerak bo'ladi, PHP da esa bu bitta oddiy buyruq bilan amalga oshiriladi.

### 10.2 - jadval. Saralash funksiyalari.

Funksiya	Saralash	Teskari tartib	<kalit/qiymat>juftligi saqlash
sort	Qiymat	Yo'q	Yo'q
rsort	Qiymat	Ha	Yo'q
asort	Qiymat	Yo'q	Ha
arsort	Qiymat	Ha	Ha
ksort	Qiymat	Yo'q	Ha
krsort	Qiymat	Ha	Ha
usort	Qiymat	?	Yo'q
uasort	Qiymat	?	Ha
uksort	Kalit	?	Ha

#### sort funksiyasining ishlatilishi

`sort()` funksiyasi massivni saralab, uning elementi *qiymatlarini* o'sish tartibi (kichigidan kattasiga qarab) bo'yicha tartiblaydi. Bu funksiya massivning barcha mavjud kalitlarini o'chirib, ularni elementlarning yangi tartibiga mos keluvchi sonli indeksga almashtiradi. Agar funksiya ishni muvaffaqiyatli yakunlasa `true` ni, aks xolda – `false` ni qaytaradi.

#### Sintaksisi:

`sort (massiv [, bayroq]);`

Qo'shimcha parametr bayroq sifatida quyidagi konstantalardan birortasi ishlatilishi mumkin:

- `SORT_REGULAR` – massiv elementlarini oddiy ravishda taqqoslaydi;
- `SORT_NUMERIC` – massiv elementlarini son kabi taqqoslaydi;
- `SORT_STRING` – massiv elementlarini satr kabi taqqoslaydi.

#### asort, rsort, arsort funksiyalari

Bu funksiyalar sintaksisi ham `sort()` funksiyasi kabi bo'lib, vazifalari bir muncha farq qiladi.

Massiv elementlarini kalitlari saqlangan holda saralash uchun `asort()` funksiyasi ishlatiladi.

#### Sintaksisi:

`asort(massiv [, bayroq]);`

---

Agar massivni teskari tartibda, ya'ni katta qiymatdan boshlab kichigiga qarab saralash kerak bo'lsa, u holda *rsort()* funksiyasi ishlatiladi.

**Sintaksisi:**

*rsort(massiv [, bayroq]);*

Shu holda, ya'ni teskari tartibda saralangan holda yana kalitlarini ham saqlash kerak bo'lsa, *arsort()* funksiyasidan foydalaniladi.

**Sintaksisi:**

*arsort(massiv [, bayroq]);*

Mos holda bayroq qiymatlari ham *sort()* funksiyasi kabi bo'ladi: *SORT\_REGULAR*, *SORT\_NUMERIC*, *SORT\_STRING*.

*SORT\_NUMERIC* bayrog'i faqatgina PHP4 da qo'shilgan.

***Massivni kaliti bo'yicha saralash: ksort(), krsort()***

Ba'zan massivni *kalit qiymatlari* bo'yicha saralash zarurati tug'iladi. Misol uchun, agar bizda yuqoridagi misolda berilgani kabi kitoblar haqidagi ma'lumotlar massivi bor bo'lsa, u holda kitoblarni muallifi bo'yicha saralashimiz mumkin. Buning uchun PHP da bir qancha satr kodlarni yozish shart emas, chunki PHP ning bu vazifalarni bajaruvchi funksiyalari mavjud.

*ksort()* funksiyasi - massiv kalitlarini kalitning mos qiymatlari saqlangan holda, o'sish tartibi (to'g'ri tartib) bo'yicha saralaydi.

*krsort()* funksiyasi - massiv kalitlarini kalitning mos qiymatlari saqlangan holda, kamayish tartibi (teskari tartib) bo'yicha saralash uchun ishlatiladi. Bu funksiyalarni sintaksisi ham *sort()* funksiyasi sintaksisiga o'xshash.

**Ko'p o'lchovli massivlarni saralash**

Massivlarni saralashda, bitta o'zgartirish kiritish yordamida ularning tartibini alifbo yoki raqamli tartibga o'tkazish mumkin. PHP da ikkita son yoki ikkita matnli satrni taqqoslash imkoniyati mavjud, biroq ko'p o'lchovli massivlarning elementlari ham massivdan iborat. PHP ning ikkita massivni taqqoslash imkoniyati mavjud emas, shuning uchun ularni taqqoslashda metodlar yaratish zarur.

***Massivlarni foydalanuvchi tomonidan aniqlangan funksiya yordamida saralash: usort(), uksort(), uasort()***

---

Massiv qiymatlarini saralashni oddiy ikkita yo‘li(kamayish yoki o‘shish tartibi) dan tashqari PHP da foydalanuvchi tomonidan berilgan biror bir mezon(kriteriya) yordamida ham saralash mumkin. Bu mezonlar funksiya yordamida berilib, funksiya nomi maxsus *usort()* yoki *uksort()* saralash funksiyalarida parametr sifatida ko‘rsatiladi. Ushbu maxsus funksiyalar foydalanuvchi tomonidan aniqlangan funksiyalar yordamida mos holda *usort()* massiv elementlari *qiymatlarini*, *uksort()* esa – *massiv kaliti qiymatlarini saralaydi*. Bu ikkala funksiya ham agar saralash ijobiy yakun topsa, true ni, aks holda false ni qaytaradi.

### Sintaksisi:

*usort(massiv, saralovchi\_funksiya)*  
*uksort(massiv, saralovchi\_funksiya)*

Albatta, massivlarni foydalanuvchi tomonidan aniqlangan funksiya yordamida saralab bo‘lmaydi. Bu funksiya massiv elementlarini taqqoslash imkonini beruvchi aniqlangan kriteriyalarni qoniqtirishi kerak. Saralovchi funksiya qanday yaratilgan bo‘lishi kerak? - degan savol tug‘iladi. Birinchidan, ikkita parametr ga ega bo‘lishi kerak. Ularga interpretator *usort()* funksiyasi uchun *element qiymati* yoki *uksort()* funksiyasi uchun *massiv kaliti* juftligini berishi mumkin. Ikkinchidan, saralovchi funksiya quyidagilardan birortasini qaytarishi kerak:

- noldan kichik butun sonni, agar birinchi parametr ikkinchisidan kichik bo‘lsa;
- nolga teng sonni, agar ikkala parametr teng bo‘lsa;
- noldan katta sonni, agar birinchi parametr ikkinchisidan katta bo‘lsa.

Boshqa saralash funksiyalari kabi *usort()* funksiyasining ham o‘xshashi mavjud, kalit qiymati o‘zgarmaydigan *uasort()* funksiyasi.

Quyida avval ko‘rilgan massiv berilgan. Bu massivda talabalar haqidagi ma’lumotlar, ularning kodi, familiyasi, ismi, stipendiyasi, kursi, manzili, tug‘ilgan sanasi, universiteti berilgan.

**10.16 – misol.** *Ko‘p o‘lchovli massivni foydalanuvchi tomonidan aniqlangan funksiya yordamida saralash*

```
<?
$talabalar=array(
 array("1","Islomov", "Ikrom", 150, 1, "Andijon", "3/12/1992", "TDPU"),
 array("3","Fozilov", "Fozil", 200, 3, "Qashqadaryo", "1/12/1990", "TDPU"),
 array("6","Salimov", "Vohid", 150, 4, "Namangan", "7/06/1989", "NamDU"),
 array("10","Karimov", "Botir", 100, 2, "Jizzax", "8/12/1991", "TDPU")
); ?>
```

Bu ko‘p o‘lchovli massivni saralash, uning qiymatlari qanday tartibda saralanadi? Bizga ma’lumki, massiv tarkibini ikkita tartibda saralash mumkin. Satrli ma’lumotlar(familiyasi, ismi, manzili, universiteti)ni alifbo tartibida, sonlilari(kodi, stipendiyasi, kursi)ni raqamli tartibda saralanadi. Buning uchun *usort()* funksiyasi va

---

elementlarni qanday taqqoslash kerakligini ko'rsatuvchi foydalanuvchi tomonidan aniqlangan funksiya ishlatiladi.

Quyidagi kod bu massivning uchinchi ustuni(ismi) qiymatlarini alifbo tartibida saralaydi.

```
function ftaf($x, $y)
{
 if ($x[2] == $u[2])
 return 0;
 else if ($x[2] < $u[2])
 return -1;
 else
 return 1;
}
usort($talabalar, ftaf);
```

Foydalanuvchi tomonidan aniqlangan funksiyalar qanday yaratilishini avvalgi mavzularda ko'rib o'tganmiz. Bu misolda foydalanuvchi tomonidan aniqlangan funksiya - **ftaf()**. **ftaf()** funksiyasi parametr sifatida ikkita qiymatni qabul qiladi: **x va y**. **ftaf()** funksiyasining vazifasi ushbu ikkita qiymatni qabul qilib, ularning tartibini aniqlash.

Qaralayotgan misolda **x va y** parametrlari har biri alohida talabaning ismini bildiruvchi asosiy massiv ichidagi massivlar. **x** massivining **name(ismi)** elementiga murojaat etish uchun **x[2]** deb yozish kerak. Chunki massiv noldan boshlab indekslanganligi uchun **name(ismi)** – ikkinchi element. Massivdagi **name(ismi)** elementini taqqoslash uchun funksiyaga uzatiladigan **\$x[2]** va **\$y[2]** o'zgaruvchilari ishlatiladi.

Funksiya o'z ishini tugatgandan so'ng javob sifatida qiymat qaytaradi. Funksiya qiymatlarini qaytarish uchun **return** kalit so'zi ishlatiladi. Masalan, **return 1;** satri funksiyaga murojaat etganda, 1 kodini qaytaradi. **ftaf()** funksiyasi **x va y** ni taqqoslash uchun **usort()** funksiyasi bilan ishlatiladi. Agar **x** ning qiymati **y** ning qiymatiga teng bo'lsa, 0 ni, kichik bo'lsa, manfiy sonni, katta bo'lsa, musbat sonni qaytaradi. **x va y** ning qiymatiga bog'liq ravishda funksiya 0,1 yoki -1 ni qaytaradi.

Kod satrining oxirida **usort()** funksiyasiga saralash kerak bo'lgan massiv(**\$talabalar**) va foydalanuvchi funksiyasi(**ftaf()**) bilan murojaat etiladi.

Agar massivni boshqa tartibda saralash lozim bo'lsa, boshqa funksiya yaratiladi. Masalan, talabalarining kurslari bo'yicha saralash lozim bo'lsa, beshinchi ustun qaraladi va saralash uchun quyidagi funksiya yaratiladi.

```
function ftaf($x, $y)
{
 if ($x[4] == $u[4])
 return 0;
 else if ($x[4] < $u[4])
 return -1;
 else
 return 1;
}
usort($talabalar, ftaf);
```

---

`usort($talabalar, ftaf);` funksiyasiga murojaat etgandan, so‘ng massiv kurslarini o‘shish tartibi bo‘yicha tartiblaydi.

`usort()` funksiyasidagi *u* belgisi “user”(«foydalanuvchi») degan ma‘noni anglatadi, shuning uchun bu funksiya foydalanuvchi funksiyasini talab etadi.

`uksort()` va `uasort()` funksiyalari `ksort()` va `asort()` larga o‘xshash, faqatgina bular ham foydalanuvchi funksiyasini talab etadi.

`asort()`ga o‘xshab `uasort()` funksiyasi ham assotsiativ massivlarni qiymatlarini saralaydi. Qiymatlar oddiy sonlar va satrlardan iborat bo‘lsa, `asort()` funksiyasi ishlatiladi. Agar qiymatlar massiv singari murakkab ob‘yektlardan tashkil topgan bo‘lsa, u holda taqqoslash uchun funksiyani aniqlash va `uasort()` funksiyasini ishlatish kerak.

`sort()`, `asort()` va `ksort()` funksiyalarining ularga mos nomida "g" belgisi ishtirok etgan teskari tartibda saralovchi o‘xshashi mavjud. Saralash uchun foydalanuvchi tomonidan aniqlangan funksiyalarning teskari versiyasi yo‘q, biroq ko‘p o‘lchovli massivlarni teskari tartibda saralash mumkin. Modomiki, taqqoslash funksiyalari dasturchilar tomonidan yaratilgan ekan, demak qarama-qarshi qiymatni qaytaruvchi taqqoslash funksiyalarini ham yaratish mumkin. Saralash teskari tartibda bajarilishi uchun, agar *x* ning qiymati *u* dan kichik bo‘lsa 1 va agar *x* ning qiymati *u* dan katta bo‘lsa -1 bo‘lishi kerak.

## Qism massivni belgilash

### *array\_slice()* funksiyasi

Massiv bu – elementlar to‘plami. Ushbu to‘plamdan birorta to‘plamosti ajratib olish uchun PHP da *array\_slice()* funksiyasidan foydalaniladi. Ajratib olingan massiv qism massiv deyiladi.

#### Sintaksisi:

*array\_slice(massiv, element\_nomeri[, uzunligi])*

Bu funksiya *element\_nomeri* parametrda berilgan nomerdagi massiv elementidan boshlab, uzunligi *uzunligi* parametriga teng bo‘lgan uzunlikdagi *qism massivni* berilgan massivdan ajratib oladi. Musbat *element\_nomeri* element nomerining tartibini mos ravishda massivning boshidan boshlab ko‘rsatadi, manfiy *element\_nomeri* esa massivning oxiridan boshlab hisoblaydi.

### 10.21 – misol. *array\_slice()* funksiyasining qo‘llanilishi

```
<?php
$arr = array(1,2,3,4,5);
$sub_arr = array_slice($arr,2);
print_r($sub_arr);
/*
Array ([0] => 3 [1] => 4 [2] => 5),
3, 4, 5 elementlaridan tashkil topgan qism massivini chiqazadi */
```



```
$sub_arr = array_slice($arr,-2);
print_r($sub_arr);
// Array ([0] => 4 [1] => 5) chiqazadi,
// 4, 5 elementlaridan tashkil topgan qism massivini chiqazadi
?>
Natija:Array ([0] => 3 [1] => 4 [2] => 5) Array ([0] => 4 [1] => 5)
```

Agar *array\_slice()* ni ishlatishda *uzunligi* parametri berilsa, u holda belgilangan qism massiv elementlari soni bu parametrdan berilgan songa teng bo'ladi. *uzunligi* parametri manfiy son sifatida ham ko'rsatilishi mumkin. Bu holda interpretator massiv oxiridan boshlab, *uzunligi* parametri moduliga teng sondagi elementni o'chiradi.

### Massiv elementlari yig'indisini hisoblash

Massiv elementlari qiymatlarining yig'indisini hisoblash uchun *array\_sum()* funksiyasidan foydalaniladi. Bu funksiyaning *parametri* sifatida esa element qiymatlari yig'indisi hisoblanishi kerak bo'lgan *massiv* ko'rsatiladi.

Sintaksisi:

```
mixed array_sum(array massiv);
```

Bu funksiyaning ishlatilishiga bir muncha qiyinroq misol ko'ramiz. Bu misolda avvalroq ko'rib o'tgan *array\_slice()* funksiyasini ham ishlatamiz.

### Massiv elementlarini joylashgan tartibini o'zgartirish

Ba'zi masalalarda massiv elementlari tartibini turli usullar bilan o'zgartirish talab etiladi. **Shuffle()** funksiyasi massiv elementlarini tasodifiy tartibda joylashtiradi. **array\_reverse()** funksiyasi massiv elementlari teskari tartibda joylashgan massiv nusxasini qaytaradi.

#### *shuffle()* funksiyasining ishlatilishi

Talabalar portalining asosiy sahifasiga a'lochi (eng ko'p stipendiya oluvchi) talabalarining rasmlarini joylashtiraylik. Bunday talabalar ko'pchilikni tashkil etishi mumkin, lekin ularning hammasini emas, tasodifiy to'rtta a'lochi talabani rasmi asosiy sahifada chiqib tursin. Chunki, har doim bir xil rasmni chiqib turishi portal foydalanuvchilarini zeriktirib qo'yishi mumkin. Agar barcha rasmlar massivda saqlansa, bu masalani hal etish oson.

### Massivlar ustida boshqa amallarni bajarish

Biz massivlarni qayta ishlovchi funksiyalar bilan tanishib chiqdik. Boshqa massivlar bilan ishlashda foydali funksiyalarni ham ko'rib o'tamiz.

**Massiv ichida ko'chirish: each(), current(), reset(), end(), next(), pos() va prev()**

Avval aytib o'tganimizdek, har bir massiv ichida massivning joriy elementini ko'rsatib turuvchi ichki ko'rsatkichi bo'ladi. Biz ushbu ko'rsatkichni *each()* funksiyani ishlatganimizda ko'rib o'tdik, biroq uni bevosita ishlatish va qo'llash mumkin.

Yangi massivni yaratishda joriy ko'rsatkich massivning birinchi elementini ko'rsatish uchun o'rnatiladi. **current(\$array\_name)** funksiyasiga murojaat etganda, birinchi elementni chiqazadi.

**next()** yoki **each()** funksiyasiga murojaat etganda, ko'rsatkich bitta element oldinga ko'chadi. **each(\$array\_name)** funksiyasiga murojaat etganda, ko'rsatkich ko'chmasdan avval, joriy elementni qaytaradi. **Next()** funksiyasi esa biroz boshqacha, **next(\$array\_name)** funksiyasiga murojaat etganda, ko'rsatkichni ko'chirib, so'ng yangi joriy elementni qaytaradi.

**reset()** funksiyasi ko'rsatkichni massivning birinchi elementiga qaytaradi. Shunga o'xshash, **end(\$array\_name)** funksiyasiga murojaat etganda, ko'rsatkich massivning oxiriga ko'chadi. **reset()** va **end()** funksiyalari mos ravishda massivning birinchi va oxirgi elementlarini qaytaradi.

Massiv ichida ko'chirishni teskari yo'nalishda bajarish uchun **end()** va **prev()** funksiyalaridan foydalaniladi. **prev()** funksiyasi **next()** funksiyasiga nisbatan teskari funksiY. U joriy ko'rsatkichni bitta element orqaga ko'chiradi, so'ng yangi joriy elementni qaytaradi.

**Masalan**, quyidagi kod massiv elementlarini teskari tartibda chiqazadi:

### 10.30 – misol.

```
<?
// $array massivi quyidagicha elon qilingan
$array = array (1, 2, 3) ;
$value = end ($array) ;
while ($value)
{
echo "$value
";
$value = prev($array);
}
?>
```

Natija:

```
3
2
1
```

**each()**, **current()**, **reset()**, **end()**, **next()**, **pos()** i **prev()** funksiyalari ishlatilganda massivda ixtiyoriy tartibda o'zgartirish uchun kod yaratish mumkin.

### **Massivni skalyar o'zgaruvchiga almashtirish: extract()**

Kalit-qiyamat juftliklaridan tashkil topgan assotsiativ massivni **extract()** funksiyasi yordamida skalyar o'zgaruvchilar to'plamiga almashtirish mumkin.

### **Sintaksisi:**

**extract (array var\_array [, int extraot\_type] [, string prefix] );**

**extract()** funksiyasi skalyar o'zgaruvchilarni yaratishda ularning nomlarini massiv kalitlari nomi bilan nomlaydi. O'zgaruvchilar massivdagi mos qiymatlarga tenglanadilar.

### 10.31- misol.

```
<?
$array = array("key1" => "value1", "key2" => "value2", "key3" => "value3") ;
extract($array);
echo "$key1 $key2 $key3";
```

```
?>
```

Natija: value1 value2 value3

Massiv kalitlari key1, key2 va key3 dan iborat uchta elementdan tashkil topgan. **extract()** funksiyasi ishlatilganda uchta \$key1, \$key2 va \$key3 skalyar o'zgaruvchilar

yaratildi. Natijada ko‘rib turganingizdek, \$key1, \$key2 va \$key3 o‘zgaruvchilarning qiymatlari mos ravishda "value1", "value2" "value3" lar bo‘ladi.

**extract()** funksiyasi ikkita zarur bo‘lmagan parametrga ega: *extract\_type* va *prefix*. *extract\_type* parametri **extract()** funksiyasiga kalit nomi bilan bir xil o‘zgaruvchilar mavjud bo‘lganda yuzaga keladigan konfliktlarni qayta ishlash usulini beradi. Odatga ko‘ra mavjud o‘zgaruvchining qiymati yangisiga almashtiriladi. *extract\_type* parametrining to‘rtta mumkin bo‘lgan qiymatlari jadvalda keltirilgan. Jadval. **extract()** funksiyasining *extract\_type* parametrini mumkin bo‘lgan qiymatlari

Qiymati	Vazifasi
EXTR_OVERWRITE	Konflikt yuzaga kelganda mavjud o‘zgaruvchini qayta yozadi.
EXTR_SKIP	Konflikt yuzaga kelganda elementni tashlab o‘tadi(qoldirib ketadi).
EXTR_PREFIX_SAME	Konflikt yuzaga kelganda \$prefix_key o‘zgaruvchisini yaratadi. Bu holda funksiyada prefix parametri ko‘rsatilishi kerak.
EXTR_PREFIX_ALL	Barcha o‘zgaruvchilarga prefix old qo‘shimchasini beradi. Bu holda funksiyaga bu parametr berilishi kerak.

Odatga ko‘ra ko‘p ishlatiladigan qiymatlar: (EXTR\_OVERWRITE) va (EXTR\_PREFIX\_ALL). Qolgan ikkita qiymat avvaldan konflikt yuzaga keladigan joyi ma‘lum bo‘lsa va kalit tushirib qoldirilgan va old qo‘shimcha ishlatilishi talab etilayotgan hollarda ishlatiladi.

Quyida EXTR\_PREFIX\_ALL ning parametr sifatida ishlatilishiga misol ko‘rsatilgan. Ko‘rib turganingizdek, yaratilgan o‘zgaruvchi *old\_qo‘shimcha\_kalit\_nomi* nomini oladi.



### Savol va topshiriqlar:

1. PHPda qanday massivlar bor?
2. Ko‘p o‘lchovli massivlar nima?
3. Massivlarga misol keltiring?



## 4.11. PHPda fayllar bilan ishlash

Ma‘lumotlarni saqlashning ikkita asosiy usuli mavjud: ikki o‘lchamli faylda va ma‘lumotlar bazasida.

Ikki o‘lchamli fayl turli formatlarda bo‘lishi mumkin, biroq ikki o‘lchamli (flat) fayl deganda oddiy matnli faylni tushunamiz. Quyida biz talabalar haqidagi ma‘lumotlarni matnli faylga yozishni ko‘rib o‘tamiz. Bunda har bir talabaning ma‘lumotlari bitta satrga yoziladi.

Bu usul oddiy va chegaralangan. Agar katta hajmdagi ma‘lumotlar bilan ishlashga to‘g‘ri kelganda, ma‘lumotlar bazasi bilan ishlagan ma‘qul. Biroq, ikki

---

o'Ichamli fayl keng ko'lamda qo'llaniladi, shuning uchun fayllar bilan ishlash texnologiyalarini o'rganish zarur.

Ma'lumotlarni faylga yozish uch bosqichda amalga oshiriladi:

1. Faylni ochish. Agar fayl mavjud bo'lmasa, uni yaratish talab etiladi.
2. Ma'lumotlarni faylga yozish.
3. Faylni yopish.

Xuddi shunga o'xshash, ma'lumotlarni fayldan o'qish ham shu uch bosqichga bog'liq holda bajariladi:

1. Faylni ochish. Agar faylni ochish mumkin bo'lmasa(masalan, fayl mavjud bo'lmasa), bu holat tekshiriladi va bu holatdan to'g'ri chiqib ketish yo'li qaraladi.
2. Fayldan ma'lumotlarni o'qish.
3. Faylni yopish.

Zarur bo'lganda, fayldan ma'lumotlarni bir o'qishda faylning qancha qismini o'qish kerakligini tanlash mumkin. Keyinroq barcha mavjud imkoniyatlarini ko'rib o'tamiz.

### **Faylni ochish va faylga bog'lanish**

PHP da faylni ochish uchun `fopen()` funksiyasi ishlatiladi. Faylni ochishda fayl ustida qanday amal olib borish mo'ljallanayotganligi ko'rsatiladi. Bu ruxsat tipi deyiladi.

Server operatsion tizimi ochilgan fayl bilan qanday amal bajarish kerakligini bilishi kerak. Undan quyidagilarni bilishi talab etiladi: balki fayl ochiq bo'lishi va ayni vaqtda agar sahifa egasining ushbu sahifa bilan ishlashga huquqi yetarli bo'lsa fayl ochiq va boshqa bir sahifa bilan qayta ishlanayotgan bo'lishi mumkin. Mantiqan olganda, ruxsat tiplari operatsion tizimga boshqa foydalanuvchilar yoki ssenariylardan tushayotgan so'rovlarni qayta ishlash usulini aniqlash mexanizmini, shuningdek biror fayl bilan ishlash uchun ruxsat va huquqlarning mavjudligini tekshirish imkoniyatini beradi.

Faylni ochish quyidagi uchta qoidaga asoslanadi:

1. Faylni faqatgina o'qish uchun, faqatgina yozish uchun yoki o'qish va yozish uchun ochiladi.
2. Faylga yozish vaqtida ixtiyoriy fayl tarkibi qayta yoziladi yoki yangi ma'lumot faylning oxiriga yoziladi.
3. Tizimdagi faylga yozish vaqtida ikkilik va matnli fayllar bir-biridan farqlanadi shuning uchun fayl tipi ko'rsatilishi talab etiladi.

Umuman olganda, PHP ning fayllar yaratish uchun belgilangan funksiyasi mavjud emas. Ko'pgina funksiyalari serverdagi fayllar sistemasidagi mavjud fayllar bilan ishlaydi. Lekin, vaqtinchalik fayllarni yaratish uchun bir nechta funksiyalari mavjud.

Eng oddiy faylni yaratish uchun lokal va masofaviy fayllarni ochuvchi funksiyadan foydalaniladi. Bu funksiya `fopen()` deb nomlanadi. `fopen()` berilgan faylni dasturni boshqaruvchi potok bilan bog'laydi.

### ***fopen()* funksiyasi**

#### **Sintaksisi:**

*resource fopen ( fayl\_nomi, ruxsat\_tipi  
[, use\_include\_path(yo'lni\_qo'shish)])*

Bu funksiya natijasi sifatida ochilgan faylga ko'rsatkichni (resurs tipi) qaytaradi. Funksiya parametrlari sifatida ochilishi kerak bo'lgan *fayl\_nomi*, faylga *ruxsat\_tipi*(bu yerda fayl bilan qanday amal bajarilmoqchiligi ko'rsatiladi) va *include\_path* (ko'rsatilgan faylni qayerdan qidirishni ko'rsatiladi) beriladi. Bu uchta parametr haqida batafsilroq ma'lumot beramiz.

*fayl\_nomi* parametri faylning to'g'ri lokal nomi yoki faylning tarmoqdagi URL-manzilidan tarkib topgan satrdan iborat bo'ladi. Agar *fayl\_nomi* kirish protokolini ko'rsatish bilan boshlansa(masalan, *http://...* Yoki *ftp://...*), u holda interpretator buni URL manzili nomi deb hisoblaydi va qayta ishlovchini ko'rsatilgan URL protokolidan qidiradi. Agar qayta ishlovchi topilgan bo'lsa, u holda PHP xuddi oddiy fayl(*allow\_url\_fopen* direktivasi) kabi URL ob'yekti bilan ishlashga ruxsat berilganligini tekshiradi. Agar *allow\_url\_fopen=off* bo'lsa, u holda **fopen()** xatolikni chaqiradi va ogohlantirishni chiqazadi. Agar *fayl\_nomi* protokol bilan boshlanmasa, u holda *fayl\_nomi* lokal nom sifatida ko'rsatilgan bo'ladi. Lokal faylni ochish uchun, PHP bu faylga mos ruxsat huquqiga ega bo'lishi kerak.

*fayl\_nomi* bir necha xil ko'rinishlarda berilishi mumkin:

- agar parametr lokal fayl nomidan iborat bo'lsa, **fopen()** funksiyasi bu faylni ochib, ko'rsatkichni qaytaradi;

- agar parametr *php://stdin*, *php://stdout* yoki *php://stderr* ko'rinishida berilgan bo'lsa, mos kirish/chiqish standart potoklari ochiladi;

- agar parametr *http://* old qo'shimchasi bilan boshlansa, u holda funksiya serverga HTTP bog'lanishni ochadi va ko'rsatkichni ko'rsatilgan faylga qaytaradi;

- agar parametr *ftp://* old qo'shimchasi bilan boshlansa, u holda funksiya serverga FTP bog'lanishni ochadi va ko'rsatkichni(manipulyatorni) ko'rsatilgan faylga qaytaradi. Bu holatda asosiy e'tiborni ushbu ikkita holga qaratish lozim: agar server FTP ning passiv rejimini qo'llamasa, u holda **fopen()** ga murojaat etish muvaffaqiyatsiz yakunlanadi. Xususan, FTP fayllari yoki o'qish uchun yoki yozish uchun ochiladi.

*ruxsat\_tipi* parametri quyidagi qiymatlardan birortasini qabul qilishi mumkin.(12.1 - jadval)

### 9.1 - jadval. Faylning ruxsat tiplari

Ruxsat tipi	Nomlanishi
r	O'qish rejimi - faylni faqat o'qish uchun ochadi; ko'rsatkich faylga uning boshidan boshlab o'rnatiladi.
r+	O'qish rejimi - fayl o'qish va yozish uchun ochiladi; ko'rsatkich faylga uning boshidan boshlab o'rnatiladi.
w	Yozish rejimi - fayl faqat yozish uchun ochiladi; ko'rsatkich faylni boshidan boshlab o'rnatiladi. Agar fayl mavjud bo'lsa, fayl nol uzunlikkacha qisqartiriladi, ya'ni tarkibidagilarning barchasi o'chiriladi. Agar fayl mavjud bo'lmasa, uni ochish vaqtida faylni yaratadi.
w+	Yozish rejimi - fayl o'qish va yozish uchun ochiladi; ko'rsatkich faylni boshidan boshlab o'rnatiladi. Agar fayl mavjud bo'lsa, tarkibidagilarning

	barchasi o‘chiriladi. Agar fayl mavjud bo‘lmasa, uni ochishga harakat qiladi va natijada faylni yaratadi.
a	Qo‘shish rejimi - fayl faqat qo‘shish(yozish) uchun ochiladi; ko‘rsatkich fayl tarkibini oxiriga o‘rnatiladi. Agar fayl mavjud bo‘lmasa, uni ochishga harakat qiladi va natijada faylni yaratadi.
a+	Qo‘shish rejimi - faylni yozish va o‘qish uchun ochadi; ko‘rsatkich fayl tarkibini oxiriga o‘rnatiladi. Agar fayl mavjud bo‘lmasa, uni ochishga harakat qiladi va natijada faylni yaratadi.
x	Faqatgina yozish uchun fayl yaratadi va ochadi; ko‘rsatkichni faylning boshiga o‘rnatadi. Agar fayl mavjud bo‘lsa, u holda fopen() false ni chiqazadi va ogohlantirish beradi. Agar fayl mavjud bo‘lmasa, u holda uni yaratishga urinib ko‘radi. Ruksatning bu tipi PHP 4.3.2 versiyasidan boshlab qo‘llaniladi va faqatgina lokal fayllar bilan ishlaydi.
x+	Yozish va o‘qish uchun fayl yaratadi va ochadi; ko‘rsatkichni faylni boshiga o‘rnatadi. Agar fayl mavjud bo‘lsa, u holda fopen() funksiyasi false ni qaytaradi va ogohlantirish beradi. Agar fayl mavjud bo‘lmasa, u holda uni yaratishga urinib ko‘radi. Ruksatning bu tipi PHP 4.3.2 versiyasidan boshlab qo‘llaniladi va faqatgina lokal fayllar bilan ishlaydi.

Quyida faylni ochish funksiyasi fopen() ga misol keltirilgan. Fayl ochilmagan holatda, xatolik haqida xabar chiqazish uchun die() chaqiruvi fopen() funksiyasi bilan birgalikda ishlatiladi.

```
<?
$file = "user.txt"; // birorta fayl
$fh = fopen($file, "a+") or die("File ($file) doesnot exist!");
?>
```

Quyidagi fragment PHP saytiga ulanishni ochadi (<http://www.php.net>):

```
<?
$site = "http://www.php.net";
// Server HTTP bog‘lanishi barchaga ochiq
$sh = fopen($site, "r");
?>
```

use\_include\_path parametrda 1 yoki TRUE qiymatlari o‘rnatilgan bo‘lsa, interpretator **fopen()** da ko‘rsatilgan faylni include\_pathdan qidiradi. include\_path – bu PHP ni sozlash faylida fayllarni yoqish uchun joylashtirilgan, direktoriyalar ro‘yxatida berilgan direktiva. **fopen()** funksiyasidan tashqari bu direktiva include() va require() funksiyalarida ham ishlatiladi.

Fayl yaratishdan oldin qaysi operatsion tizimda ishlayapsiz va bu faylni qaysi operatsion tizim(OT)da o‘qish mumkin, shuni bilish kerak. Gap shundaki, turli operatsion tizimlarda satr tugallanishi turlicha belgilanadi. Unix OTida satr tugallanishi \n, Windows tipidagi OT larda esa satr tugallanishi \r\n bilan belgilanadi. Satrlarni Unix tipidagi tizimlardan Windows tizimiga o‘tkazish uchun satr tugallanishi belgisini maxsus bayroq t sifatida belgilash taklif etiladi. Buning aksini bajaruvchi b bayrog‘i ham mavjud bo‘lib, binar fayllarda tez-tez ishlatiladi. Bu bayroqlarni *fayl* uchun tanlangan *ruksat tipi* belgisi oxirisida yozib ishlatish mumkin. Masalan, o‘qish uchun ochilayotgan faylda, satr oxiridagi barcha belgilarni \r\n ga

---

qayta kodlashtirish uchun, r o‘rniga rt ni ishlatish kerak. Agar binar fayllarni ochishda b bayrog‘i ishlatilmasa, u holda mavjud fayllarni o‘zgartirish bilan bog‘lik xatolar yuzaga keladi. Dasturni turli platformalarga ko‘chirish masalalarida, **fopen()** yordamida fayllarni ochishda doimo b bayrog‘ini ishlatish tavsiya etiladi.

Agar **fopen()** yordamida faylni ochish yoki yaratish amalga oshmasa, u holda PHP da xatolik yuzaga keladi, fopen funksiyasi esa o‘zining ishi natijasi sifatida false qiymatni qaytaradi. Bunday ko‘rinishdagi xatolikni @ belgisi yordamida yashirish mumkin.

**Masalan**, quyidagi satr, agar faylni ochish amalga oshmasa ham, xatolik haqidagi ogohlantirishni chiqazmaydi:

```
$h = @fopen("dir/another_file.txt","w+");
```

Bu holda **fopen()** funksiyasi faqatgina bo‘sh faylni yaratadi va unga yozish uchun ruxsatni ta‘minlaydi. Bu faylga ma‘lumotlarni qanday yozish mumkin? Mavjud fayldan ma‘lumotlarni qanday o‘qish mumkin?

Bu savollarga javob berishdan avval **fopen()** yordamida ochilgan bog‘lanishni yopishni ko‘rib o‘tamiz. Fayl bilan ishlab bo‘lgandan so‘ng har doim **fclose()** funksiyasi yordamida bog‘lanishni yopish kerak.

## Fayl bilan bog‘lanishni yopish

Fayldagi ma‘lumotni o‘qish, yozish yoki boshqa biror-bir zarur amallarni bajarish uchun bu fayllar bilan **fopen()** yordamida bog‘lanish o‘rnatiladi. Bu amallar bajarib bo‘lingandan so‘ng, albatta bog‘lanishni yopish zarur. Buning uchun **fclose()** funksiyasi ishlatiladi.

*fclose() funksiyasi*

**Sintaksisi:**

*fclose (faylga ko‘rsatkich)*

Bu funktsiya natijasi sifatida bog‘lanishni yopish muvaffaqiyatli amalga oshirilgan bo‘lsa, TRUE qiymatini, aks holda FALSE ni qaytaradi. **fclose()** funksiyasi **fopen()** yoki **fsockopen()** funksiyalari yordamida ochilgan fayllarni muvaffaqiyatli yopadi. Bu funktsiyaning parametri sifatida **fopen()** funksiyasi yordamida ochilgan fayl ko‘rsatkichini ko‘rsatish kerak.

### 9.2 - misol. fclose() funksiyasining ishlatilishi

```
<?php
$h = fopen("my_file.html", "w");
fclose($h);
?>
```

Agar fayl bilan bog‘lanish yopilmasa, skriptni bajarishda hech qanday xatolik yuzaga kelmaydi. Lekin bu serverda jiddiy oqibatlariga olib kelishi mumkin. Masalan, xakker ochiq bog‘lanishdan foydalangan holda faylga virus yozishi mumkin, serverda bu ortiqcha yo‘qotishni yuzaga keltiradi. Shuning uchun fayl bilan zarur amallarni bajargandan so‘ng bog‘lanishni yopishni maslahat beramiz.

Ochiq fayllar bilan ikkita asosiy amal bajarish mumkin – o‘qish va yozish. Undan avval esa fayl mavjudligini tekshirib olish zarur.

## Fayl mavjudligini tekshirish

Biz amallar bajarmoqchi bo‘lgan fayllar mavjud bo‘lmasa, yoki o‘qish va yozish uchun ruxsat berilmagan bo‘lsa, u holda amallar bajaruvchi funksiyalarni xech

---

qaysi biri ishlaymadi va PHP xatolik haqida xabar chiqazadi. Shuning uchun fayl mavjudligini quyidagi funksiyalar yordamida tekshirib olish kerak: `file_exists()`, `is_writable()`, `is_readable()`.

### ***Fayl mavjudligini tekshirish: `file_exists()` funksiyasi***

#### **Sintaksisi:**

`bool file_exists (fayl yoki katalog nomi)`

`file_exists()` funksiyasi argument sifatida berilgan nomdagi fayl yoki katalog mavjudligini tekshiradi. Agar katalog yoki fayl server fayllar sistemasida mavjud bo'lsa `TRUE`, aks holda `FALSE` ni qaytaradi. Bu funksiyalar ishining natijasi keshlanadi (kompyuter xotirasidagi jildda saqlab qo'yadi). Ushbu keshlarni tozalash tavsiya etiladi, buning uchun dastur kodiga `clearstatcache()` buyrug'ini qo'shib, muvozanat holatidagi kesh tozalanadi. Lokal bo'lmagan fayllar uchun `file_exists()` funksiyasini ishlatib bo'lmaydi.

#### **Misol. `fclose()` funksiyasining ishlatilishi**

```
<?
$file = "user.txt";
if (file_exists($file))
{
 $fh = fopen($file, "r");
 // bu yerda fayllar bilan amallar bajariladi
 fclose($fh);
}
else
{print "File $file does not exist!";}

?>
```

### ***Faylga yozish mumkinligini tekshirish: `is_writable()` funksiyasi***

`bool is_writable (string fayl)`

Fayl mavjudligini tekshirishdan tashqari, bu faylga ma'lumot yozish mumkinligini ham tekshirish mumkin, buning uchun `is_writable()` funksiyasi yoki uning psevdonimi `!is_writable()` funksiyasidan foydalaniladi. Yozish imkoniyatining mavjudligi fayl uchun qanday tekshirilsa, katalog uchun ham shunday tekshiriladi.

#### **Sintaksisi:**

`bool is_writable (fayl yoki katalog nomi)`

bu funksiya agar fayl mavjud bo'lsa va unga yozish mumkin bo'lsa `TRUE` qiymatini qaytaradi. Faylga ruxsat serverda ishlayotgan foydalanuvchining (ko'pincha bu foydalanuvchi nobody yoki www) foydalanuvchi nomi orqali amalga oshiriladi. `is_writable()` funksiyasi ishining natijasi keshlanadi.

### ***Faylni o'qish mumkinligini tekshirish: `is_readable()` funksiyasi***

Fayldagi ma'lumotni o'qish mumkinligini tekshirish uchun `is_readable()` funksiyasidan foydalaniladi. O'qish imkoniyatining mavjudligi fayl uchun qanday tekshirilsa, katalog uchun ham shunday tekshiriladi.

#### **Sintaksisi:**

`bool is_readable (fayl yoki katalog nomi)`

**Misol.** `my_file.txt` fayli mavjudligi va uni o'qish uchun ruxsat borligini



**is\_readable()** funksiyasi yordamida tekshirish.

```
<?php
$filename = 'c:/users/files/my_file.txt';
if (is_readable($filename)) {
 print "$filename fayl mavjud va o`qish mumkin";
} else {
 print "$filename fayl mavjud emas yoki o`qish mumkin emas";
}
?>
```

## Faylga ma'lumot yozish

### *fwrite()* funksiyasi

fopen() funksiyasi yordamida ruxsat ochilgan faylga ma'lumot yozish uchun fwrite() funksiyasi ishlatiladi.

#### Sintaksisi:

int fwrite (faylga ko'rsatkich, o'zgaruvchi [, uzunligi])

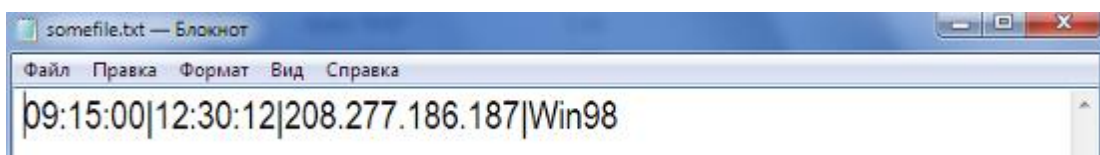
Bu funksiya o'zgaruvchi tarkibidagi satrlarni faylga ko'rsatkich ko'rsatgan faylga yozadi. Agar qo'shimcha parametr - uzunligi berilgan bo'lsa, u holda yozuv yozilayotgan belgilar soni ushbu parametr qiymatiga teng bo'lganda yoki satr oxiriga yetganda tugatiladi.

fwrite() funksiyasi o'zining ishi natijasi sifatida yozuv bayti sonini yoki xatolik yuzaga kelsa false ni chiqazadi.

#### Misol. fwrite() funksiyasining qo'llanilishi

```
<?
// foydalanuvchi sayti trafigi haqidagi ma'lumot
$data = "09:15:00|12:30:12|208.277.186.187|Win98";
$filename = "somefile.txt";
// faylni ochib ko'rsatkichni fayl oxiriga o'rnatish
$fh = fopen($filename, "a+");
// agar fayl mavjud va unga yozish mumkinligini tekshirish
if (is_writable($filename)) :
// $data tarkibini faylga yozish
$success = fwrite($fh, $data);
// faylni yopish
fclose($fh); else :
print "Could not open $filename for writing";
endif;
?>
```

## Natija



skript ishga tushganda so'ng somefile.txt fayl tarkibida:

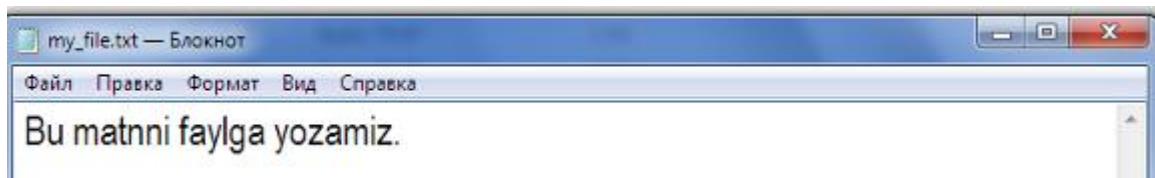
**9.1 – rasm.** Dastur kodi ishga tushgandan so'ng hosil bo'lgan somefile.txt fayli

**Misol. fwrite()** funksiyasining qo'llanilishi.

Bizning ishchi katalogimizda my\_file.txt fayli mavjud emas. Uni yaratamiz va unga matn satrlarini yozamiz:

```
<?php
$faylnomi="my_file.txt";
$h = fopen($faylnomi,"w");
$text = "Bu matnni faylga yozamiz.";
if (fwrite($h,$text))
 echo "Yozuv muvaffaqiyatli yakunlandi";
else
 echo "Ma`lumotlarni yozishda xatolik yuzaga keldi";
fclose($h);
?>
```

**Natija:**Yozuv muvaffaqiyatli yakunlandi



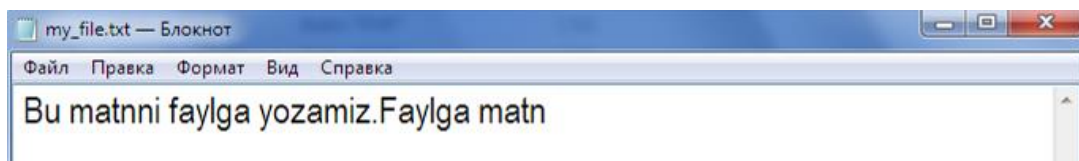
**9.2 – rasm.** Dastur kodi ishga tushgandan so‘ng hosil bo‘lgan my\_file.txt fayli

Bu skript ishi natijasi sifatida agar yozish muvaffaqiyatli amalga oshirilsa, brauzerda my\_file.txt faylini ochganimizda «Bu matnni faylga yozamiz.» xabarini ko‘rishimiz mumkin. Agar ushbu fayl yuqoridagi skriptni bajarguncha mavjud bo‘lgan bo‘lsa, u holda undagi barcha ma‘lumotlar o‘chiriladi va ma‘lumot qayta yoziladi.

Agar biz quyidagicha skript yozsak:

```
<?php
$faylnomi="my_file.txt";
$h = fopen("$faylnomi ","a");
$add_text = "Faylga matn qo`shamiz.";
if(fwrite($h,$add_text,11))
 echo "Matn qo`shish muvaffaqiyatli amalga oshirildi
";
else echo "Ma`lumotlarni qo`shishda xatolik yuzaga keldi
";
fclose($h);
?>
```

**Natija:**Matn qo`shish muvaffaqiyatli amalga oshirildi



**9.3 – rasm.** my\_file.txt faylining keyingi ko‘rinishi

U holda, avvaldan mavjud bo‘lgan my\_file.txt faylidagi satr davomiga \$add\_text o‘zgaruvchisi tarkibidagi satrning 11 ta belgisi qo‘shiladi, ya’ni «Faylga matn»

**fwrite()**(file write - faylga yozish) funksiyasining u kabi ishlaydigan **fputs()**(file put string – faylga satrlarni yozish) o‘xshashi mavjud. Bu funktsiyani **fwrite()** funksiyasi o‘rnida ishlatish mumkin va uning sintaksisi **fwrite()** kabidir. **fputs()** funksiyasining **sintaksisi**:

int **fputs** (faylga ko‘rsatkich, o‘zgaruvchi [, uzunligi])

Ma‘lumotlarni faylga yozishda jadvalning har bir qatori faylning alohida satrlariga yoziladi. Satrlarni bir-biridan ajratish uchun yangi satr belgisi(«\n») ishlatiladi.

---

Ma'lumotlar maydonlari jadvalda berilgan tartibda yoziladi, bu maydonlarni ajratish uchun esa tabulyatsiya belgisi («\t») ishlatiladi. Maydonlarni ajratish uchun o'qish oson bo'lgan ixtiyoriy belgini ishlatish mumkin.

### **Fayldan ma'lumotlarni o'qish**

Fayldagi ma'lumotlarni o'qish fayllar bilan ishlashning eng asosiy amallaridan biridir. Quyida fayldan o'qishni amalga oshiruvchi bir qancha funksiyalar berilgan.

Agar mavjud fayldagi ma'lumotni o'qimoqchi bo'lsak, ma'lumot yozishdagi kabi fopen() funksiyasini ishlatish yetarli emas. Bu ochiq faylga ko'rsatkichni qaytaradi xolos, lekin fayldagi birorta satrni o'qimaydi. Shuning uchun, fayldan ma'lumotlarni o'qish uchun, maxsus funksiyalar (*file*, *readfile*, *file\_get\_contents*, *fread*, *fgets* va h.) dan birortasidan foydalaniladi.

#### ***Fayl oxirini aniqlash: feof() funksiyasi***

PHP da fayl pozitsiyasi ko'rsatkichi fayl oxiriga kelganligini ko'rishni tekshiruvchi maxsus funksiya mavjud. Bu bulev funksiya - feof(). feof() funksiyasi bitta parametr *faylga ko'rsatkich*ni qabul qiladi.

#### **Sintaksisi:**

*string feof(faylga ko'rsatkich)*

Bu funksiya nomini eslab qolish oson, uning to'liq nomi File End Of File (Fayl: fayl tugadi).

#### ***Satrlarni berilgan uzunligiga ko'ra o'qish: fread() funksiyasi***

Bu funksiyadan fayldagi ma'lumotlarni o'qishda foydalaniladi. Uni binar fayldagi ma'lumotlarni o'qish uchun ham ishlatish mumkin.

#### **Sintaksisi:**

*string fread (faylga ko'rsatkich, uzunligi)*

bu funksiya *faylga ko'rsatkich* ko'rsatgan fayldan *uzunligi* parametrda ko'rsatilgan uzunlikka(baytga) teng ma'lumotni o'qiydi. *faylga ko'rsatkich* parametri aniq mavjud bo'lgan, masalan fopen() yordamida ochilgan fayl bilan bog'langan resurs tipidagi o'zgaruvchi bo'lishi kerak. *faylga ko'rsatkich* ochilgan va o'qish uchun ruxsat berilgan faylga yuboriladi. Ma'lumotlarni o'qish fayl oxirigacha yoki *uzunligi* parametrda ko'rsatilgan baytga teng satrni o'qib bo'lguncha davom etadi.

**fread()** funksiyasining natijasi sifatida fayldagi satrdan o'qilgan ma'lumotlar chiqaziladi.

Bu funksiyada *uzunligi* parametri albatta ko'rsatilishi kerak. Shunday ekan, agar biz fayldagi barcha satrlarni o'qimoqchi bo'lsak, u holda uning uzunligini bilishimiz kerak. PHP ko'rsatilgan fayl uzunligini hisoblay olish imkoniyatiga ega. Buning uchun **filesize()** funksiyasidan foydalaniladi.

#### ***Fayl o'lchamini(uzunligini) aniqlash: filesize() funksiyasi***

Fayl o'lchami **filesize()** funksiyasi yordamida aniqlanadi. Bu funksiya fayl o'lchamini baytlarda chiqazadi.

#### **Sintaksisi:**

*filesize(fayl nomi)*

Bu funksiya **fread()** funksiyasi bilan birgalikda faylni bir vaqtda barchasini (yoki uning aniq bir qismini) o'qish uchun ishlatiladi.

Xatolik yuzaga kelsa, bu funksiya false ni qaytaradi. Afsuski, uni faqatgina *lokal fayllarni hajmi(uzunligi)*ni olishgagina ishlatish mumkin.

**Misol. fread()** funksiyasining qo'llanilishi.

my\_file.txt fayli tarkibini o'qiymiz

```
<?php
$h = fopen("my_file.txt","r+") or die("Can't openfile!");
// faylni yozish va o'qish uchun ochamiz
$content = fread($h,
 filesize("my_file.txt"));
// fayl tarkibidagi satrlarni o'qiymiz
echo filesize("my_file.txt").'
';
// faylni uzunligini chiqazamiz
fclose($h); // fayl bilan bog'lanishni yopamiz
echo $content;
// brauzer ekraniga fayl tarkibini chiqazish
?>
```

**Natija:**36

Bu matnni faylga yozamiz.Faylga matn

Windows kabi tizimlarda mavjud binar fayl(masalan rasm)ni o'qish uchun fayl tarkibining oxirida rb yoki uni o'xshashi b bayrog'i yordamida ochish tavsiya etiladi.

**filesize()** funksiyasi o'z ishining natijasini keshlab qo'yadi(kompyuter xotirasidagi jildda saqlab qo'yadi). Agar my\_file.txt fayli tarkibi o'zgartirilib, so'ng yuqoridagi skript qayta ishga tushirilsa, uning ishini natijasi o'zgarmaydi. Skriptni bu fayldagi ma'lumotlarni o'qish uchun boshqa funksiya(masalan, fgetss)dan foydalanib yozsak ham, natija **fread()** dagi kabi bo'ladi.

**Satrima-satr o'qish: fgets(), fgetss(),fgetcsv() funksiyalari**

**fgets() funksiyasi**

**fgets()** funksiyasi yordamida fayldagi matn satri o'qiladi. Bu funksiyani sintaksisi **fread()** kabi bo'lib, undan farqli tarafi o'qiladigan satr uzunligini ko'rsatish shart emas:

*string fgets ( faylga ko'rsatkich [, uzunligi])*

**fgets()** funksiyasi o'z ishi natijasi sifatida *faylga ko'rsatkich* ko'rsatgan fayldan satr uzunligi (*uzunligi-1*) baytga teng satrni chiqazadi.

**fread()** va **fgets()** funksiyalarning ikkalasi ham ko'rsatkich agar fayl oxiriga kelsa, fayldan ma'lumotlarni o'qishni to'xtatadi.

**Masalan,** my\_file.txt faylidagi barcha satrlarni quyidagicha o'qish mumkin:

```
<?php
$h = fopen("my_file.txt","r");
while (!feof ($h)) {
 $content = fgets($h);
 echo $content,"
";
}
fclose($h);
?>
```

**Natija:**Bu matnni faylga yozamiz.Faylga matn

Esga olamiz, PHP da bitta belgi – bu bir bayt. Agar o'qiladigan satr uzunligi ko'rsatilmasa (berilayotgan imkoniyat PHP 4.2.0 versiyasidan boshlab kiritilgan), u holda 1 Kbayt (1024 bayt) matn yoki 1024 belgi o'qiladi.

PHP 4.3 versiyasidan boshlab, agar *uzunligi* parametri berilmasa, satrni hammasi o‘qiladi.

Xatolik yuzaga kelsa **fgets()** funksiyasi false ni qaytaradi. PHP ning 4.3 versiyasidan boshlab ikkilik fayllar uchun bu funksiya xavfsiz hisoblanadi.

Quyidagi shartlardan birortasi bajarilsa, o‘qish to‘xtatiladi:

- fayldan (*uzunligi-1*) bayt yoki belgi o‘qilgan bo‘lsa;
- fayldan keyingi(yangi) satrga o‘tish belgisi o‘qilsa;
- fayldan fayl tugaganligi belgisi(EOF) o‘qilsa;

Agar fayldan har bir satrni o‘qish talab etilsa, u holda ikkinchi parametr sifatida satrdagi baytlar soniga teng qiymatni berish kerak. Fayldan satrma-satr o‘qish va chiqazish uchun misol keltiramiz:

### ***fgetss() funksiyasi***

**fgets()** funksiyasining yana bir o‘xshashi bu – **fgetss()** funksiyasidir. Bu funksiya ham ko‘rsatilgan fayldan satrlarni o‘qish uchun imkon yaratadi va shu bilan birga faylda uchraydigan barcha html va PHP– teglarni o‘chiradi.

#### **Sintaksisi:**

*string fgetss(faylga ko‘rsatkich, uzunligi [, ruxsat berilgan teglar])*

E’tibor bering, bu yerda uzunligi parametri albatta bo‘lishi kerak.

### ***fgetcsv() funksiyasi***

**fgetcsv()** — funksiyasi **fgets()** funksiyasining yana bir varianti. U quyidagicha **sintaksisga** ega:

*array fgetcsv(int faylga ko‘rsatkich, int uzunligi, string [ajratuvchi]);*

bu funksiya fayldagi satrlarni *ajratuvchi* sifatida berilgan belgi yordamida ajratish uchun ishlatiladi. Ajratuvchi sifatida ishlatiluvchi belgilar elektron jadvallar va b. larda qo‘llanilgan tabulyatsiya belgisi, vergul va h. lar bo‘lishi mumkin.

Bu funksiyaga **fgets()** funksiyasiga murojaat etgan kabi murojaat etiladi, faqatgina unga maydonlarni ajratish uchun ishlatilgan ajratuvchi belgi beriladi. *Uzunligi* parametri fayldan o‘qilayotgan eng uzun satrdan katta uzunlikka ega bo‘lishi kerak.

### ***Belgini o‘qish: fgetc() funksiyasi***

Agar fayldan ma’lumotlarni satrma-satr o‘qish mumkin bo‘lsa, u holda belgima-belgi o‘qish ham mumkin. Buning uchun **fgetc()** funksiyasidan foydalaniladi.

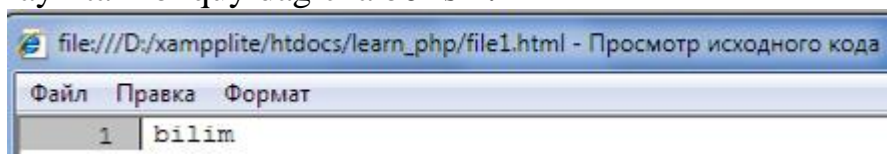
#### **Sintaksisi:**

*string fgetc ( faylga ko‘rsatkich)*

Bu funksiya *faylga ko‘rsatkich* da ko‘rsatilgan fayldagi belgilarni chiqazadi va agar satr oxiri uchrasa FALSE qiymatini qaytaradi.

**Misol.** Bu misolda fayl belgilarini **fgetc()** funksiyasi yordamida bittalab o‘qish ko‘rsatilgan:

file1.html fayli tarkibi quyidagicha bo‘lsin.



### **9.8 – rasm. file1.html faylining ko‘rinishi**

<?php

```

$ h = fopen("file1.html", "r");
while (!feof ($h)) {
 $content = fgetc($h);
 echo $content. "
";
}
fclose($h);
?>
Natija:
b
i
l
i
m

```

Fayl tarkibini o'qish uchun **fopen()** funksiyasi yordamida faylni ochib, bog'lanish o'rnatish shart emas. PHP da buni bitta fayl nomini ishlatib bajara oladigan funksiyalari mavjud. Bu funksiyalar `readfile()`, `file()` va `file_get_contents()`.

## Fayllar bilan ishlovchi funksiyalar

### *Readfile() funksiyasi*

#### Sintaksisi:

*int readfile ( fayl nomi [, use\_include\_path])*

**readfile()** funksiyasi *fayl nomi* parametrda berilgan faylni o'qiydi va uni tarkibini, belgilar baytini ekranga chiqazadi. Agar qo'shimcha parametr `use_include_path` TRUE qiymatiga teng bo'lsa, u holda berilgan nomdagi faylni `include_path` ga kiruvchi kataloglardan qidiradi. Dasturda bu funksiya faylning o'qilgan belgilari va baytini chiqazadi, aks holda xatolik yuzaga kelganda – FALSE ni qaytaradi. Xatolik haqidagi xabarni bu funksiyada @ amali yordamida yashirish mumkin.

**Misol.** `readfile()` funksiyasining qo'llanilishi.

Quyidagi skript, `file.txt` fayli mavjud bo'lsa, uni tarkibini va hajmini chiqazadi. Aks holda xatolik yuzaga kelganda «Error in readfile» satrini chiqazadi.

```

<?php
$ n = @readfile ("somefile.txt");
/* fayl tarkibini ekranga chiqazadi va uning hajmini $n o`zgaruvchisiga yozib
qo`yadi*/
if (! $n) echo "Error in readfile";
/* agar readfile() funksiyasi xatolik bilan bajarilsa,
u holda $n=false va xato haqida xabarni chiqazamiz */
else echo $n;
// agar xato bo`lmasa, o`qilgan
// belgilar sonini chiqazamiz
?>
Natija:Fayllar bilan ishlashni o`rganamiz. 36

```

**readfile()** funksiyasi yordamida fayl nomi sifatida berilgan, URL-manzili ko'rsatilgan, uzoq masofadagi fayl tarkibini o'qish mumkin (agar bu opsiya serverni sozlashda o'chirilmagan bo'lsa).

Fayl tarkibini darhol ekranga chiqazish har doim ham qulay emas. Gohida, keyinchalik bu fayl bilan biror-bir amalni bajarish imkoni bo'lishligi uchun fayldagi ma'lumotni o'zgaruvchiga yozish kerak bo'ladi. Buning uchun **file()** yoki **file\_get\_contents()** funksiyalarini ishlatish mumkin.

**Fayldagi ma'lumotlarni massivga o'tkazish: file() funksiyasi**

*file()* funksiyasi fayldagi ma'lumotlarni massiv tipidagi o'zgaruvchiga o'tkazib o'qiydi. Bu yerda massivning har bir elementi faylning bitta satriga mos tushadi. Bu funksiyaning sintaksisi **readfile()** funksiyasi kabi bo'lib, farqli tarafi u natijasi sifatida massivni chiqazadi.

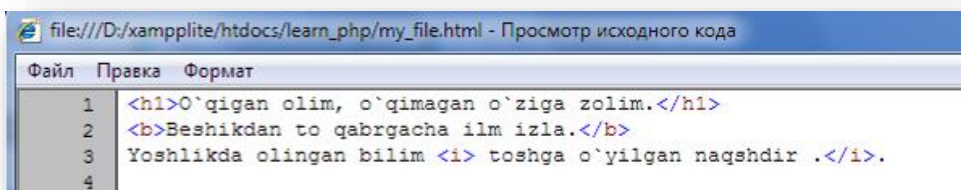
**Sintaksisi:**

*array file ( fayl nomi [, use\_include\_path])*

Bu funksiya qanday massivni chiqazadi? Berilgan massivning har bir elementi biz o'qiyotgan ma'lumotlar fayli(uning nomi fayl nomi parametrda beriladi)ning satri hisoblanadi. *file()* funksiyasida xatolik yuzaga kelsa, boshqalari kabi false ni qaytaradi. Qo'shimcha parametr *use\_include\_path* berilgan fayl *include\_path* katalogida bor yoki yo'qligini aniqlash uchun ishlatiladi. Agar server tomonidan ta'qiqlanmagan bo'lsa, bu funksiya yordamida uzoq masofadagi fayllarni ham ochish mumkin. PHP 4.3 versiyasidan boshlab bu funksiya orqali binar fayllar bilan ishlash xavfsiz hisoblanadi.

**Misol.** *file()* funksiyasining qo'llanilishi

*my\_file.html* fayli tarkibi quyidagicha bo'lsin.



**9.9 – rasm.** *my\_file.html* faylining ko'rinishi

Uning tarkibini **file()** funksiyasi yordamida o'qiydiz:

```
<?
$file_array = file("my_file.html");
while (list($line_num, $line) = each($file_array))
{
 print "Satr $line_num: ". htmlspecialchars($line). "
\n";
}
?>
Natija:
Satr 0: <h1>O`qigan olim, o`qimagan o`ziga zolim.</h1>
Satr 1: Beshikdan to qabrgacha ilm izla.
Satr 2: Yoshlikda olingan bilim <i> toshga o`yilgan naqshdir .</i>.
Satr 3:
```

**Fayl ko'rsatkichi o'rnini o'zgartirish: rewind(), fseek() va ftell()**

Fayl ichida fayl ko'rsatkichining o'rnini(pozitsiyasini) aniqlash va uni rewind(), fseek() va ftell() funksiyalari yordamida o'zgartirish.

**rewind()** funksiyasi fayl ko'rsatkichini faylning boshlanishiga o'rnatadi.

**Sintaksisi:**

*int rewind( faylga ko'rsatkich);*

**ftell()** funksiyasi esa ko'rsatkich o'rnini fayl boshidan boshlab baytlarda aniqlaydi.

**Sintaksisi:**

*int ftell( faylga ko'rsatkich);*

**fseek()** funksiyasi fayl ko'rsatkichini fayl ichidagi ba'zi nuqtalarga o'rnatish uchun ishlatiladi.

---

**Sintaksisi:**

*int fseek(int faylga ko'rsatkich, int offset);*

**fseek()** funksiyasiga murojaat etgandan so'ng, *faylga ko'rsatkich* fayl boshidan boshlab *offset* bayt siljigan nuqtaga o'rnatiladi.

*rewind()* funksiyasi *fseek()* funksiyasiga nol miqdorda siljish uchun murojaat etilganda (*offset=0* da) ekvivalent bo'ladi.

**Masalan**, *fseek()* funksiyasini fayldagi yozuvlarni o'rtasini topishda yoki binar qidiruvlarni bajarishda ishlatish mumkin.

**Faylni satrlarga o'qib olish: file\_get\_contents() funksiyasi**

PHP ning 4.3 versiyasidan boshlab fayl tarkibini satrlarga o'qib olish imkoniyati paydo bo'ldi. Bu **file\_get\_contents()** funksiyasi yordamida bajariladi. Bu funksiya ham parametr sifatida *fayl nomini* oladi va *use\_include\_path* ko'rsatkichiga ko'ra uni *include\_path* katalogidan qidiradi.

**Sintaksisi:**

*string file\_get\_contents (fayl nomi [, use\_include\_path])*

bu funksiya **file()** funksiyasiga juda o'xshash, faqatgina u fayl tarkibini satr sifatida chiqazadi. Bundan tashqari u binar ma'lumotlarga ishlov berish uchun xavfsiz va agar server sozlashlarida ta'qiqlanmagan bo'lsa, masofada joylashgan fayldagi ma'lumotlarni o'qiy oladi.

**Faylni o'chirish: unlink() funksiyasi**

Fayl ustida bajariladigan amallardan biri – bu faylni o'chirish amali. PHP da faylni o'chirish uchun *unlink()* funksiyasidan foydalaniladi.

**Sintaksisi:**

*bool unlink ( fayl nomi)*

bu funksiya *fayl nomida* ko'rsatilgan nomga ega bo'lgan faylni o'chiradi va agar funksiya ishi muvaffaqiyatli amalga oshirilgan bo'lsa TRUE ni, aks holda xatolik yuzaga kelsa FALSE ni qaytaradi. Faylni o'chirish uchun ham, unga mos ruxsat huquqiga ega bo'lish kerak (masalan, o'qish uchun berilgan ruxsat, faylni o'chirishga ruxsat bermaydi).

**Misol. Unlink() funksiyasining qo'llanilishi**

```
<?php
$filename = 'd:/xampplite/htdocs/learn_php/my_file.html';
if (unlink($filename))
{echo $filename.' fayli o`chirildi'; }
else
{echo $filename.' fayli o`chirilmadi';}
// d:/xampplite/htdocs/learn_php/my_file.html
// nomli faylni o`chirish
?>
Natija:d:/xampplite/htdocs/learn_php/my_file.html fayli o`chirildi
```

**Tashqi dasturlarni yurgizish: yexec() funksiyasi**

PHP ssenariylari serverda joylashgan dasturlarni ham bajarishi mumkin. Bunday imkoniyatlar tizimni web-brauzer orqali boshqarishda tez-tez ishlatiladi hamda tizim haqida ma'lumot olish uchun qulay.

**yexec()** funksiyasi berilgan dasturni ishga tushiradi va u chiqargan ma'lumotlarni oxirgi satrini qaytaradi.

**Sintaksisi:**

*string exec (string buyruq [, string massiv [, int qaytarish]])*



**yexes()** funksiyasi ishi natijasi sifatida hech narsa chiqarmagan holda, faqatgina buyruqni bajaradi. Buyruq natijasida olingan barcha ma'lumotlar shart bo'lmagan parametr - *massiv*da saqlanadi. Bundan tashqari, agar berilgan *massiv* parametrda *qaytarish* o'zgaruvchisi ham berilsa, buyruq bajarilishi so'ngida qaytarish kodlari ushbu o'zgaruvchiga tenglanadi.

**Fayllar tizimi bilan ishlash: *basename()* funksiyasi**

Serverdagi fayllarni ko'rish va turli amallarni bajarish uchun PHP ning maxsus funksiyasi mavjud. Server fayllari atributlari(turgan joyi, egasi va vakolatlari) haqidagi ma'lumotlar tez-tez kerak bo'ladi. **basename()** funksiyasi berilgan to'liq nomdan fayl nomini ajratib oladi.

**Sintaksisi:**

*string basename(string to'liq nomi)*

faylning tayanch nomini to'liq nomdan ajratib olish uchun quyidagicha yoziladi:

```
<?
$path = "/usr/local/phppower/htdocs/index.php";
$file = basename ($path); // $file = "index.php"
?>
```

Bu funksiya to'liq nomdan unga ko'rsatilgan yo'lni o'chiradi va fayl nomini o'zini qoldiradi.

**Faylni muhofazalash(blokirovkalash)**

Faraz qilaylik, bir vaqtning o'zida ikkita foydalanuvchi web-forma yordamida ma'lumotlarni kiritmoqchi. Birinchi foydalanuvchi *fopen()* funksiyasiga murojaat etib, faylga yozishni boshladi, ikkinchi foydalanuvchi ham *fopen()* funksiyasiga murojaat etdi va faylga yozish uchun urinmoqda. Natijada fayl tarkibi qanday bo'ladi? Avval birinchi foydalanuvchi ma'lumoti yoziladi, keyin ikkinchisikimi yoki aksincha? Yoki bu ikki ma'lumot ixtiyoriy tarzda galma-gal yoziladimi?

Bu savollarga javob berish ishlatilayotgan operatsion tizimga bog'liq va ularga aniq javob berib bo'lmaydi. Shunga o'xshash muammolardan qochish uchun fayllarni muhofazalash ishlatiladi. PHP da muhofazalash *flock()* funksiyasi yordamida amalga oshiriladi. Bu funksiyaga faylni ochgandan so'ng, lekin faylga ma'lumotlarni yozish va ma'lumotlarni o'qishdan oldin murojaat etiladi.

**Sintaksisi:**

*bool flock (int faylga ko'rsatkich, int blok\_amali);*

bu funksiyaga parametr sifatida *faylga ko'rsatkich* va son(*blok\_amali*) beriladi. Bu son talab qilinayotgan muhofazalash amalining turi. Agar muhofazalash muvaffaqiyatli amalga ohsa, funksiya true qiymatini, aks holda false ni qaytaradi.

*blok\_amali* parametrining qiymatlari (muhofazalash amalining turi) 9.3 - jadvalda berilgan.

**9.3 - jadval. *flock()* funksiyasining *blok\_amali* parametrining qiymatlari.**

| <b>blok_amali parametrining qiymati</b> | <b>Vazifasi</b>                                                                                                        |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| \                                       | O'qishni muhofazalash. Bu shuni anglatadiki, fayl boshqa o'quvchi ilovalar bilan birga(sovmestno) ishlatilishi mumkin. |

|    |                                                                                                                                                          |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2  | Yozishni muhofazalash. Bu yakka (monopoliya) rejimi. Fayl birga ishlash uchun ruxsatga ega emas.                                                         |
| 3  | Mavjud muhofazadan chiqarish.                                                                                                                            |
| +4 | <i>blok_amali</i> parametrining joriy qiymatiga 4 qo'shilsa, joriy muhofazalash bajarilayotgan vaqtda boshqa muhofazalashga urinishlarni bartaraf etadi. |

Agar flock() funksiyasini ishlatishga qaror qilingan bo'lsa, uni berilgan fayl ishlatilayotgan tegishli barcha sahifalarga kiritish kerak, aks holda u o'z mazmunini yo'qotadi.

**Misol.** Muhofazalash uchun misol ko'ramiz, learn.php faylini quyidagicha o'zgartiramiz:

```
<?
$qqoshish="Yuqorida maqollar berilgan.";
$fh = fopen("my_file.txt", "a", 1);
flock($fh, 2); // yozish uchun faylni muhofazalash
fwrite($fh, $qqoshish);
flock($fh, 3); // yozish uchun muhofazalashni olib tashlash
fclose($fh);
?>
```

Hamda insert.php sahifasida ham faylni blokirovkalashni qo'shamiz:

```
<?php
$fh = fopen("my_file.txt", "r");
flock($fh, 1); // fayldan o`qish uchun faylni muhofazalash
flock($fh, 3); // muhofazalashni olib tashlash
fclose($fh);
?>
```

Endi kod ishonchli, biroq hali ham mukammal emas. Agar ikkita sahifa bir vaqtning o'zida bitta faylni muhofazalashga urinsa, nima bo'ladi? Jarayonlar muhofazalashni o'rnatishda baxslashib qolib, konflikt yuzaga kelishi mumkin va bunga qaysidir biri muvaffaq bo'ladi, biroq o'z navbatida yana yangi muammo tug'iladi. Shuning uchun bunday masalalarni yechishda MBBT ni ishlatiladi, bu kabi masalalar MBBTda nisbatan muvaffaqiyatli yechiladi.



### Savol va topshiriqlar:

1. PHPda fayllarni ochish?
2. PHPda fayllardan nusxa olish, nomini o'zgartirish va fayllarni o'chirish?
3. PHPda papkalar bilan ishlashni tushuntiring?
4. PUT va POST usullari?

---

## X U L O S A

Xulosa qilib aytganda, jamiyat barcha jabhalarida ro'y berayotgan hamma yangiliklar, kashfiyotlar va rivojlanishning to'xtovsiz o'sishi zamirida ta'lim tizimida erishilayotgan yutuqlar yotadi. Buni boshqacha qilib ta'lim tizimida erishilayotgan yutuqlar samaralari deb ham qarash mumkin. Hammamizga ma'lum faqat ta'lim tizimidagina jamiyatning barcha jabhalari uchun zarur yuksak bilimli, iqtidorli kadrlar tayyorlanadi. Ta'lim tizimida samaradorlikka erishish uchun esa professor-o'qituvchilardan o'z ustida tinimsiz ishlash, fan-texnika yangiliklaridan o'z vaqtida xabardor bo'lish va ularni o'z mehnat faoliyatiga tatbiq qilish, chet tillarini bilish, axborot texnologiyalaridan oqilona foydalana olishi zarur bo'ladi.

Axborot texnologiyalari va uning eng so'nggi yutuqlaridan xabardor bo'lish va ularni o'z sohalariga tatbiq qilib samarali mehnat faoliyatini tashkil qilish muhim ahamiyat kasb etadi. Yuqorida aytilganidek, ta'lim tizimida o'quv materiallarining vizuallashtirishi bilan ta'lim samaradorligi ortib boradi. Bunda esa elektron o'quv resurslaridan keng foydalaniladi. Turli fanlar bo'yicha elektron o'quv resurslarini yaratish va joriy qilish orqali ta'lim tizimida o'qitish samaradorligiga erishiladi. elektron o'quv resurslarini yaratish uchun esa avvalo boshlang'ich kompyuter savodxonligiga ega bo'lish, shu bilan birga maxsus dasturiy vositalardan foydalanish va bir qancha dasturlash tillarini bilish talab qilinadi. Boshlang'ich kompyuter savodxonligi informatika va axborot texnologiyalari fanlarini o'qitish orqali o'rgatilsa, elektron o'quv resurslarini yaratish esa Web dizayn kabi fanlarni o'qitish orqali o'rgatiladi. Shu maqsadda ushbu o'quv qo'llanmada ham elektron o'quv resurslari uchun turli illyustratsiyalar tayyorlashga mo'ljallangan grafik muharrirlar, web saytlarni yaratishga mo'ljallangan dasturiy vositalar, animatsiyalar tayyorlashga mo'ljallangan dasturlarni o'rgatishga oid ma'lumotlar keltirilgan.

---

## IZOHLI LUG‘ATI:

### A

**Animatsiya** - bir necha tasvir yoki kadrlarni ko‘rsatish orqali yaratiladigan harakat taqlidi. Animatsiya turli xil vizual effektlar (harakatlanadigan kartinkalar, chizmalar, jadvallar va boshqalar) qo‘llanilishiga asoslangan kompyuterning dinamik grafikasi, ekranda harakat illyuziyasini hosil qiladigan dinamik tasvirlar sintezidir.

**Axborot butunligi** - axborot va uni tashuvchining holati. Butun axborot va uning alohida tarkibiy qismlari bo‘linmasligini ta‘minlash hamda ularni ruxsatsiz qasddan yo‘q qilish, buzib talqin qilish, sizib chiqib ketish, o‘g‘irlash, qalbakilashtirish va almashtirib ko‘yishni, oldindan bartaraf qilishni nazarda tutiladi.

**Axborot mazmuni** - ma'lum ob'ekt yoki hodisa to‘g‘risida jami elementlar, tomonlar, ular o‘rtasidagi aloqa va munosabatlarni belgilovchi aniq ma'lumotlar.

**Axborot tizimi** – axborotni uzatish va qabul qilish tizimi. Axborot manbai, uzatkich, aloqa kanallari, axborotni qabul qiluvchilardan tarkib topgan bo‘ladi.

### B

**Banner** - veb-sahifadagi reklama xarakteridagi tasvir yoki matn bloki. U reklama beruvchining veb-saytiga yoki mahsulot yoxud xizmat turi atroflicha bayon qilingan sahifalarga giper ishoratdan iborat.

**Bilimlar bazasi (BB)** - aniq bir predmet sohasi bo‘yicha dalillar va qoidalar shaklida rasmiylashtirilgan bilimlar to‘plami. Inson tomonidan aniq predmet sohasi bo‘yicha yig‘ilgan bilimlarni kompyuterda ifodalash uchun mo‘ljallangan semantik (ma'noli) model. Biror-bir predmet sohasiga oid tushuncha, qoida va dalillarning tizimlashgan majmuini saqlash uchun bitta fayl yoki maxsus tashkil qilingan fayllar to‘plami. Bilimlar bazasi (BB) sun'iy tafakkur (intellekt) masalalarini echishda keng qo‘llaniladi.

**Bosh sahifa** - asosiy sahifa. Ingliz tilidan to‘g‘ri tarjima qilinganligidan qat'iy nazar, bu muayyan insonning uy (shaxsiy) sahifasi emas, balki veb-saytning boshlang‘ich sahifasidir. Odatda murojaatlar aynan veb-saytning bosh sahifasiga qilinadi, shuning uchun Ushbu sahifaga tashrif buyuruvchilar soni xohlagan boshqa sahifaga qaraganda ko‘proq. Bosh sahifa (veb-sayt yuzi) bo‘yicha foydalanuvchi qaerda ekanligi va saytning boshqa sahifalarida nimalarni ko‘rish mumkinligi haqida tasavvur oladi (ba'zan bosh sahifa birinchi va yagona bo‘ladi).

### V

**Vektorli grafika** - bu grafika vositasida shakllangan tasvir sodda grafik ob'yektlar to‘plamidan tuzilgan bo‘lib, uning tipik elementiga mos keladi. Vektorli tasvirning asosiy elementi chiziq bo‘lib hisoblanadi.

**Video ma'ruza** – interfaol bo‘lmagan video material ko‘rinishida taqdim etilgan o‘quv materialini ko‘rinishi.

### W

**Web** – инглиз тилидан олинган бўлиб, ўргимчак ини, тола, тўқима, ўргимчак ини, тармоқ каби маъноларни билдиради.

**www** (world wide web) – жахон ўргимчак уяси деган маънони билдиради.

---

**Veb-resurslar katalogi (ingl. Web directory)** - tavsiflar bilan birga berilgan internet-resurslarga tizimlashtirilgan va rubrikator asosida tartibga solingan giperishoratlarning termasi. Kataloglar ixtisoslashgan (soha bo'yicha) va umumiy hamda hududiy, milliy va global turlarga bo'linadi.

**Veb-bog'lama** - veb-serverda umumiy katalogda saqlangan, bir-biri bilan bog'liq bo'lgan veb-sahifalar, rasmlar, hujjatlar, multimedia va boshqa fayllar to'plami.

**Veb-dizayn** (ingliz tilidan olingan bo'lib, web-design – veb sahifani loyihalash ma'nosini bildiradi) – bu veb-sahifani jihozlanishidir. Web-dizayn sayt uchun huddi poligrafiya dizayni va qog'ozli nashrlar dastgohlari singari muhim rol o'ynaydi. Veb-dizayn deganda nafaqat veb sayt uchun grafikli elementlarni yaratish, balki uning strukturasi loyihalash, unda harakatlanish vositalari, ya'ni butun saytni yaratish tushuniladi.

**Veb-interfeys** - foydalanuvchiga veb-brauzer orqali turli dasturlar bilan o'zaro ishlash imkonini beruvchi interfeys (masalan, o'z buyurtmasini boshqarish, Internet do'konida yoki tarmoq printerini sozlash). Veb-interfeyslarining qulayligi shundaki, ular bitta ofisda joylashmagan xodimlarga birgalikda ish yuritish imkonini beradi.

**Veb-master** – veb-sahifalarni loyihalash, yaratish va bezash bilan shug'ullanuvchi shaxs. Veb-master Internet texnologiyalari bo'yicha bilimlar majmuasi va rassom-bezaklovchi tajribasiga (kompozitsiya, dizayn) ega bo'lishi lozim. Saytning tashqi ko'rinishi va ishi uchun javobgar kompaniya xodimi. Veb-master deganda turli-tuman majburiyatlar doirasi tushuniladi – kichik oddiy sayt uchun sahifalashtiruvchidan tortib dizayner va tizim ma'murigacha. Internet foydalanuvchilari uchun veb-master bu sayt va kompaniya domeniga bog'liq barcha masalalar bo'yicha aloqada bo'ladigan shaxsdir.

**Veb-ranglar** - rasmlarda ranglarni aniq solishtirish va aks ettirish uchun ishlatilishi mumkin bo'lgan 256 rangdan 216tasini o'z ichiga olgan ranglar jadvali qolgan 40 rang ishlatilmaydi, chunki ular kompyuterlarda rang uzatish sifatining sozlanganligiga ko'ra turlicha aks ettirilishi mumkin. Veb standart palitra odatda xavfsiz ranglar palitrasi deyiladi.

**Veb-sayt** - inglizcha “site” (tarjimasi “joy”) so'zining o'zbekcha talaffuzi. Umumjahon o'rgimchak to'ri ma'lum axborot topish mumkin bo'lgan va noyob URL bilan belgilangan virtual joy. Mazkur URL veb-saytning bosh sahifasi manzilini ko'rsatadi. O'z navbatida, bosh sahifada veb-saytning boshqa sahifalari yoki boshqa saytlarga murojaatlar bo'ladi. Veb-sayt sahifalari HTML, ASP, PHP, JSP, grafik va boshqa fayllardan tashkil topgan bo'lishi mumkin. Veb-saytni ochish uchun brauzer dasturidan foydalaniladi. Veb-sayt shaxsiy, tijorat, axborot va boshqa bo'lishi mumkin.

**Veb-sayt statistikasi** - saytlar egalariga saytda tashrifchilar soni, qaysi bo'limlar eng ko'p mashhur bo'lgan va boshqa narsalarni bilish imkonini beradi.

**Veb-sayt tuzilmasi** - veb bog'lamasini sahifalari orasidagi aloqalar to'plami.

**Veb-sahifa** - Internet manzili (URL) bilan bir xil ma'noda belgilanuvchi mantiqiy birlik. U veb-saytning tarkibiy qismidir. Veb saytlardan iborat bo'lsa, saytlar esa o'z navbatida sahifalardan iborat deyish mumkin. Fizik nuqtai nazardan u HTML fayldir. Matn, tasvirlar, JAVA appletlari va boshqa elementlardan iborat bo'lishi mumkin.

---

Sahifa statik yoki dinamik shakllantirilgan bo'lishi mumkin. Freymlardan foydalangan holda har bir freym alohida sahifa hisoblanadi.

**Veb-sahifa sarlavhasi** - veb-sahifani aynanlashtiruvchi tavsiflovchi matn. Ochiq sahifa nomi veb-brauzeri oynasining sarlavha qatorida aks o'tiriladi.

**Veb-server** – Internet yoki Intranetga ulangan umumfoydalanishdagi axborot serveri. Unda hujjatlar va fayllar – audio, video, grafik va matn fayllari – saqlanib, ular foydalanuvchilarga HTTP vositalari orqali taqdim etiladi. Web-server nomi u umumjahon tarmog'ining qismi bo'lgani uchun kelib chiqqan. Maxsus dasturiy ta'minotga ega bo'lgan, bir yoki bir necha veb-sayt fayllarini saqlash va ularga ishlov berishi mumkin. Bir necha veb-sayt bitta kompyuterda ishlasa, veb-server deganda veb-sayt ishlovchi virtual makon (dasturiy ta'minot va kompyuterdagi joy) tushuniladi.

**Veb-shablon** - mundarijani veb-sahifa dizaynidan ajratish uchun va veb-hujjatlarni ko'plab ishlab chiqish uchun qo'llaniladigan moslamadir. Veb-shablondan istagan inson yoki tashkilot o'zini veb-saytini tashkil veb-server qilish uchun foydalanishi mumkin. Shablon sotib olingandan yoki saqlab olingandan so'ng, foydalanuvchi barcha shablonning asosiy ma'lumotlarini o'zining tashkiloti yoki mahsuloti ma'lumotlari bilan to'ldiradi.

**Veb-hujjat** - odatda, maxsus HTML (Hypertext Markup Language) tilidagi hujjat. Veb-hujjat Umumjahon tarmog'i asosini tashkil qiladi. Ular gipermatndan iborat bo'lib, foydalanuvchiga ajratib ko'rsatilgan so'z yoki jumlagaga qaratib, ma'lumotlarni o'qish, hujjatning boshqa qismiga yoki ayni hujjat bilan giperishorat yordamida bog'langan boshqa veb-hujjatga o'tish imkonini beradi. Veb-hujjat, shuningdek, matn, tasvir, tovush, videolarni mujassamlovchi gipermuhit ma'lumotni ham o'z ichiga olishi mumkin. Veb-hujjatni ochish, ularni o'qish yoki aks ettirish Internet brauzerlari yordamida amalga oshiriladi. Veb-hujjat tushunchasi “veb-sahifalar” va “veb-saytlar” tushunchalari bilan chambarchas bog'liq. Odatda veb-sahifasi atamasi veb-hujjat atamasining sinonimini bildiradi, **veb-sayti** atamasi esa yagona mavzu ostida birlashtirilgan yoki bitta tashkilot, muallif yoki foydalanuvchiga tegishli bo'lgan sahifalar majmuasiga tegishlidir.

## G

**Gipermatn** - matnni kompyuterda ifodalash shakli. Unda ajratilgan tushunchalar, ob'ektlar va bo'limlar orasidagi ma'noli bog'lanishlar avtomatik tarzda qo'llab-quvvatlanadi. Displayning ekraniga gipermatnni chiqaradigan va ma'noli aloqalar bo'yicha o'tishlarni amalga oshiradigan axborot dasturi. Gipermatn klaviatura yoki sichqoncha yordamida, matnning rang bilan ajratilgan qismi – murojaatni shu zahotiy oq ekranga chiqaradi. Bular mazkur so'z yoki jumlagaga ta'rif va izohlar, adabiyotlar ro'yxatiga murojaatlar va bundan keyingi o'qishga oid tavsiyalar bo'lishi mumkin. Gipermatnning ikki guruhini ajratishadi. Uning muallifi tomonidan ko'zda tutilmagan ob'ektlarni unga qo'shish mumkin bo'lsa, u ochiq gipermatn deb ataladi. Dinamik gipermatn turi uchun, uni kattalashtirish amalini qo'llash odatiy holdir. Gipermatn, global ulanish xizmatida veb-sahifalarini yozishda keng ishlatiladi. Zamonaviy dasturiy vositalarning so'rov (Help) tizimlari gipermatn ko'rinishida

---

yaratilmoqda. Gipermatnlar ta'lim tizimlarida, izohli lug'atlarda va masofaviy o'qitishda keng ishlatilmoqda.

**Gipermatnli belgilash tili** - markerlash tili. Internetning global ulanish xizmatida hujjatlarning yozma shaklini belgilaydi. HTML tili, matn muharriri yordamida tayyorlangan matnga kiritiladigan buyruqlar majmuasidan iborat bo'lib, veb-sahifalarni yaratishda ishlatiladi. HTML abzatslarni formatlash, sarlavha bilan ishlash, ramzlarni formatlash, axborot bloklarini ifodalash, dastlabki tayyorlangan matnlarni, tasvirlarni va tovush parchalarini qo'llanma qilib qo'shish; gipermatnli murojaatlarini yaratish; ma'lumotlarni kiritishning interaktiv formalarini tashkillashtirish kabilarni yuzaga chiqaradi.

**Gipermedia** - turli ma'lumotlarni kompyuterda ifodalash. Bunda ajratilgan tushunchalar, ob'ektlar va bo'limlar orasidagi ma'noli bog'lanishlar avtomatik tarzda quvvatlanadi. Barcha turdagi axborotlarni ifodalash texnologiyasi. Ifoda o'zaro assotsiativ bog'langan, nisbatan katta bo'lmagan bloklar shaklida bo'ladi. Gipermedia gipermatnga o'xshash, ammo, bog'lanadigan bloklar sifatida matn parchalari emas, balki ixtiyoriy tabiatdagi ma'lumotlar: grafik tasvirlar, videokliplar, tovush fayllari va shu kabilar bo'lishi mumkin.

**Giperhavola** – bir elektron axborot ob'ektidan boshqasiga havola (masalan, matndan eslatmaga yoki adabiyotlar ro'yxatiga, bitta entsiklopedik maqoladan boshqasiga).

**Gipermatn** – hujjatlar o'rtasidagi aloqa (giperhavola) yordamida ma'lumotni taqdim etish usuli; unda o'quv materialini elektron shaklda interfaol matn sifatida giperhavolalar bilan bog'lanadi.

## D

**Dasturiy kod** – bu kompyuter dasturi bo'lib, belgilangan dasturlash tilida tegishli pedagogik va texnologik stsenariy asosidagi algoritmi bo'yicha yoziladi.

**Dyuym** (niderlandcha duim — katta barmoq) — evropaning metrli bo'lmagan o'lchov tizimlarida ishlatiladigan o'lchov birligi. Tarixdan esa dyuym - katta erkak kishining katta barmog'i enini bildiradi. Bugungi kunda kompyuter grafikasidagi o'lchov birligi hisoblanib, 1 dyuym 2,54 santimetrga teng.

## I

**Intellektual o'qitish tizimi** - avtomatlashtirilgan o'rgatuvchi tizim. U o'rganuvchiga o'rganish jarayonida muloqot olib borish, savollarga javob berish va vazifalarni tabiiy tilda bajarishga imkon beruvchi aqliy interfeysga ega.

**Internet portali** - (ingl. portal – darvoza) Internet foydalanuvchisiga turli interaktiv xizmatlarni (pochta, izlash, yangiliklar, forumlar va h.k.) ko'rsatuvchi veb-sayt. Portallar gorizontal (ko'p mavzularni qamrovchi) va vertikal (ma'lum mavzuga bag'ishlangan, masalan avtomobil portali, yangiliklar portali), xalqaro va mintaqaviy (masalan, uznet yoki runetga tegishli bo'lgan), shuningdek, ommaviy va korporativ bo'lishi mumkin.

**Ichki veb-sayt** - tashkilot doirasida yaratilgan va faqat o'sha tashkilot ichki tarmog'idan foydalanish mumkin bo'lgan veb-sayt.

**Interaktiv dastur** – interfaol dasturiy modul (test o'tkazish, modellashtirish, imitatsiya).

---

## K

**Kompyuter grafikasi** – kompyuterlar yordamida tasvirlarni yaratish va ishlov berish texnologiyasi. Hisoblash texnikasidan grafik tasvirlarni yaratish, ularni turli vositalar orqali aks ettirish (masalan, monitor ekranida, qattiq nusxalar shaklida va h.k.) va joyini, shaklini o'zgartirish maqsadida foydalanish sohasi. Kompyuterlar, tasvirlarning sintezi hamda real dunyodan olingan vizual axborotga ishlov berish uchun ishlatiladigan faoliyat turi. Ushbu faoliyatning mahsuloti ham kompyuter grafikasi deb ataladi. Avvaliga, oq-qora chizmalarni va sxemalarni tayyorlash jarayoni kompyuter grafikasi deb atalgan. Ammo, tez orada turli-tuman ranglarni ishlatadigan rasmlar paydo bo'ldi. Harakatlanmaydigan rangli tasvirlardan keyin videofilmlar paydo bo'ldi. Endi esa, uch o'lchamli tasvirlar tobora keng tarqalmoqda. Hozirgi kunda kompyuter grafikasi, uni yangicha tushinishda, hattoki virtual haqiqiylikni yarata oladi. Vizualizatsiya tobora ko'proq ahamiyat kasb etmoqda. Kompyuter grafikasi yordamida vektorli tasvirlar va rastrli tasvirlar yaratilmoqda.

**Korporativ portal** - ichki foydalanish uchun mo'ljallangan korporativ veb-sayt. U kompaniya xodimlariga korporativ axborotga, elektron tijorat maydonchalariga (ta'minotchilar, mijozlar bilan o'zaro harakat va boshqalar), hamda cheklangan sonli tashqi veb-saytlardan foydalanishni taqdim qiladi.

**Kompyuter testlash tizimi** – bir tomondan bilim oluvchining o'zini-o'zi nazorat qilish imkonini bersa, ikkinchi tomondan esa joriy, oraliq va yakuniy baholashga imkon beradigan o'quv resurslardir.

## M

**Masofaviy o'qitish** - o'qituvchi va o'quvchi o'rtasidagi to'g'ridan-to'g'ri, shaxsiy aloqasiz "masofadan o'qitish" imkonini yaratib beruvchi zamonaviy axborot va telekommunikatsion texnologiyalaridan foydalanishga asoslangan o'qitish jarayonini amalga oshirishning yangi uslubi. Eng yaxshi an'anaviy va innovatsion metodlar, o'qitish vositalari va shakllarini o'z ichiga olgan sirtqi va kunduzgi ta'lim singari axborot va telekommunikatsiya texnologiyalariga asoslangan ta'lim shaklidir.

**Masofaviy ta'lim** – ta'lim muassasasidan ma'lum masofada yashovchi shaxsga aniq bir maqsadga yo'naltirilgan, metodik ta'minlangan o'quv faoliyati. Masofadan turib o'quv axborotlarini almashuvchi vositalarga asoslangan, o'qituvchi maxsus axborot muhit yordamida, aholining barcha qatlamlari va chet ellik ta'lim oluvchilarga ta'lim xizmatlarini ko'rsatuvchi ta'lim kompleksidir.

**Ma'lumotlar bazasi** - elektron hisoblash mashinalari yordamida qidirib topilishi va qayta ishlanishi mumkin bo'lgan tarzda tartibga solingan ma'lumotlar to'plami (masalan: maqolalar, hisob-kitob). Aniq qoidalar asosida tashkil qilingan va amaliy dasturlarga bog'liq bo'lmagan ma'lumotlar to'plami. Bu qoidalar ma'lumotlarni ta'riflash, saqlash va joyining o'zgarishiga oid umumiy tamoyillarni nazarda tutadi.

**Ma'lumotlar banki** - ma'lumotlar majmui. Bu ma'lumotlar berilgan mavzuga tegishli bo'lib foydalanuvchilar bilan o'zaro ta'sir qila olishini ta'minlaydigan tarzda tashkil qilingan. Ma'lumotlarni markazlashtirilgan holda saqlash va jamoa bo'lib foydalanishning avtomatlashtirilgan tizimi. Uning tarkibiga ma'lumotlar bazasi yoki ularning majmui, ma'lumotlar bazasi spravochnigi, MBBT, hamda so'rovlar va amaliy dasturlar kutubxonasi kiradi.



---

**Masofaviy ta'lim instituti (MTI)** - zamonaviy axborot va kommunikatsiya texnologiyalari asosida biror universitetning innovatsiya ta'lim muhitini shakllantirish maqsadida tashkil etiladi. U universitet bo'limlari, kafedralari va fakultetlari faoliyatini axborotlashtirish jarayonlarini muvofiqlashtiradi hamda masofaviy ta'lim texnologiyasidan foydalangan holda turli ta'lim dasturlarini amalga oshiradi. Masofaviy ta'lim institutlari multimediyaga va WEB – texnologiyalari asosida o'quv-uslubiy va texnologik bazaga va zamonaviy kompyuter vositalari va litsenziyalangan dasturiy ta'minotga ega bo'lishi lozim. Binobarin, ular yordamida multimediyali kurslarni yaratish va tayyorlash, o'quv jarayonini uslubiy va texnologik jihatdan quvvatlash lozim bo'ladi.

**Multimedia** – inglizchadan olingan: multi – ko'p va media – tashuvchi, muhit. Axborotni turli shakldagi tashuvchilar bo'lmish tovush, tasvir va matnlar birikmasi. Vizual va audio effektlarning o'zaro muloqotli dasturiy ta'minot boshqaruvida birgalikda namoyon bo'lishi. Odatda bu matn, tovush va grafikaning, so'nggi vaqtlarda esa animatsiya va videoning ham birlashishini bildiradi. Multimedia veb-bog'lamalari va ixcham disklarning tavsifli, agar eng muhimi bo'lmasa, xususiyatli giperishoratlardir. Multimedia (multimedia vosita) – turli tipdagi matn, rasm, chizma, jadval, diagramma, fotosurat, video va audio fragmentlar kabi ma'lumotlarni raqamli ko'rinishda yaratish, saqlash, qayta ishlash va ijro qilishning kompyuterli vositalari.

**Multimedia ma'ruza** – turli vositalardan (elektron doska yoki boshqa vositalar) foydalanib o'qituvchi ma'ruzasini videoga yoki raqamli ma'ruza ko'rinishida yozilgan va qabul qilish samaradorligini oshirilgan o'quv materiallari.

## **P**

**Piksel** – kompyuter ekrannining eng kichik nuqtasi.

## **U**

**Uy sahifasi** - brauzer tomonidan dastur yuklangandan so'ng terminalda paydo bo'ladigan veb-sahifaning, portalning, majmuaning birinchi sahifasi. Odatda, prezentatsiya va navigatsiya bo'yicha ham asosiy ish bajaradi. Veb-sayt foydalanuvchisi ochadigan birinchi veb-sahifa. Saytdan foydalanish undan boshlanadi. Odatda, foydalanuvchi o'zining qaerda ekanligi va saytning boshqa sahifalarida nimalarni ko'rishi mumkinligi haqida uy sahifasidan ma'lumot oladi.

## **E**

**Elektron katalog** - mijozlar va hamkorlar uchun mahsulot hamda xizmatlar haqidagi ma'lumotlarni o'z ichiga olgan axborot tizimi. Ishlab chiqaruvchilar va xaridorlar orasida qo'shimcha axborot almashuviga imkon beradi. Elektron kataloglar elektron savdo tizimida keng qo'llaniladi.

**Elektron jurnal** - Internet tarmog'i orqali tarqatilayotgan turli mavzudagi nashr.

**Elektron kutubxona** - navigatsiya va ishlash vositalari bilan ta'minlangan turli xil elektron hujjatlarning tartibga solingan majmuasi. Elektron kutubxonalar universal yoki ixtisoslashgan bo'lishi mumkin. Elektron kutubxona – o'zining hujjatlashtirish va xavfsizlik tizimi bo'lgan, to'la hajmdagi axborot manbalarini yig'ish va

---

foydalanuvchiga havola qilish imkoniyatini beruvchi dasturiy majmua (MT kontsepsiya).

**Elektron ma'lumotlar** – «ma'lumotlar» termini ostida shunday ma'lumotlar nazarda tutilmoqdaki, bunday ma'lumotlarni boshqa kompyuterga yoki kompyuterlarga uzatish mumkin bo'ladi. Shu sababi bu erda elektron ma'lumot degan termin ishlatilmoqda. Bugungi kunga kelib ko'pgina kompyuterlar global Internet tarmog'iga, ma'lum bir yo'nalish yoki soha bo'yicha qurilgan korporativ tarmoqqa (Intranet) ulangandir. Bu esa keng miqyosda masofaviy o'qitishni yo'lga qo'yish imkonini beradi.

**Elektron universitetlar** – bu Internetdan foydalangan holda ta'limning yangi texnologiya va shakli. Elektron universitetda ta'lim jarayoni Internet orqali ish joyida va uyda, oliy va o'rta maxsus ta'lim maskanida va maktabda, hatto xizmat safarida tashkil etilishi mumkin. Elektron universitetlar orasida etakchi universitet Butun jaxon taqsimot universiteti hisoblanadi. Universitetning bo'limlari Moskvada, Ostanada, Sofiyada, Bryusselda, Peterburgda, Olmaotada ochilgan. Universitetni ta'asis etuvchisi - Xalqaro Axborotlashtirish Akademiyasi, u dunyoning 60 mamlakatida o'z bo'limlariga ega.

**Elektron hujjat** - elektron shaklda qayd etilgan, elektron raqamli imzo bilan tasdiqlangan va elektron hujjatning uni identifikatsiya qilish imkoniyatini beradigan boshqa rekvizitlariga ega bo'lgan axborot. Elektron hujjat texnika vositalaridan va axborot tizimlari xizmatlaridan hamda axborot texnologiyalaridan foydalanilgan holda yaratiladi, ishlov beriladi va saqlanadi. Elektron hujjat elektron hujjat aylanishi ishtirokchilarining mazkur hujjatni idrok etish imkoniyatini inobatga olgan holda yaratilishi kerak.

**Elektron ma'ruza** – elektron shakldagi o'quv materiallari to'plami bo'lib, ma'ruza matni, o'quv fani nazariy ma'lumotlarni namoyish qiluvchi, ma'ruzani qo'shimcha ma'lumotlar bilan boyituvchi interfaol elementlar va giperhavolalarni o'z ichida jamlagan multimedik tizimdir.

ayti.

**Elektron o'quv qo'llanma** – giperbog'lanish elementlari, animatsiya va audio effektlari hamda bilimni mustaqil nazorat qilish tizimi bo'lgan, maxsus ishlab chiqilgan yoki tanlangan dastur yordamida ifoda etiluvchi o'quv materiallari (MT konts). Elektron o'quv qo'llanma – yuqori ilmiy va metodik darajada yaratilgan, to'liq ta'lim standartlaridagi mutaxassisliklar va yo'nalishlardagi fanlar mos keluvchi, standart va dasturning birliklari bilan belgilangan, interfaol teskari aloqani tashkil qilishda o'quv jarayoni didaktik siklining uzluksizligi va to'laqonliligini ta'minlovchi asosiy ta'lim nashri hisoblanadi. Elektron o'quv qo'llanma – kompyuter texnologiyasiga asoslangan o'quv uslubini qo'llashga, mustaqil ta'lim olishga hamda fanga oid o'quv materiallar, ilmiy ma'lumotlarning har tomonlama samarador o'zlashtirilishiga mo'ljallangan bo'lib:

- o'quv va ilmiy materiallar faqat verbal (matn) shaklda;
- o'quv va ilmiy materiallar faqat verbal (matn) va ikki o'lchamli grafik shaklda;
- multimedia (multimedia – ko'p axborotli) qo'llanmalar, ya'ni ma'lumot uch o'lchamli grafik ko'rinishda, ovozli, video, animatsiya va qisman verbal (matn) shaklda;

---

- taktil (his qilinuvchi, seziladigan) xususiyatli, o'quvchini «ekran olamida» stereo nusxasi tasvirlangan real olamga kirishi va undagi ob'ektlarga nisbatan harakatlanish tasavvurni yaratadigan shaklda ifodalanadi.

**Elektron ma'lumotnoma** – ilmiy va amaliy xarakterdagi qisqa ko'rsatmalardan tarkib topgan, qidirib topishga qulay va tartib bilan joylashtirilgan elektron resurslar.

**Elektron axborot resurslari** – magnit optik tashuvchi yoki kompyuter tarmoqlarida (lokal, mintaqaviy, global) joylashgan va o'zida o'quv axborotni elektron yozuvini saqlagan yuqori ilmiy metodik va texnik saviyada bajarilgan nashrlardir.

**Elektron ta'lim** (Electronic Learning) - elektron o'qitish tizimi, sinonim elektron o'qitish, masofaviy o'qitish, kabi atamalar sinonimi bo'lib, kompyuterlarni qo'llash bilan o'qitish, tarmoqli o'qitish, virtual o'qitish, shuningdek axborot, elektron texnologiyalar yordamida o'qitish.

## F

**Fraktal** - lotincha Fractus so'zidan olingan va u «qismlardan tuzilgan» ma'nosini anglatadi.

**Fraktal grafika** – bu grafika vositasida shakllangan tasvirlar ham xuddi vektorli grafika kabi matematik hisoblarga asoslangan. Ammo kompyuter xotirasida hech qanday ob'yektni saqlamasligi bilan undan farq qiladi. Tasvir tenglama (yoki tenglamalar tizimi) bo'yicha quriladi. Shuning uchun formulalardan boshqa hech narsani saqlash kerak emas.

## R

**Rastrli grafika** - bu grafika vositasida shakllangan tasvir asosan elektron va poligrafiya nashriyotlarida qo'llaniladi. Rastrli tasvir ikki o'lchovli massiv (matritsa) ko'rinishdagi nuqtalar to'plamidan iborat bo'lib, ular piksellar deb ataladi. Rastrli tasvirning eng kichik elementi peksildan iborat.

## M

**Mengamer ranglar** - spektrlari har xil, ammo bir xil rang beruvchi nurlar.

**Monoxromatik nur** - spektri birgina to'lqin uzunligi mos kelgan bitta chiziqdan iborat bo'lgan nurlanish.

## L

**Lazer** - monoxromatik nurlanishning ancha sifatli manbayidir. Xuddi shu sababli uning nurini fokusda yig'ish oson kechadi.

## K

**Kolorimetriya** - rang va uni o'lchash bilan shug'ullanadigan fan.

---

## FOYDALANILGAN ADABIYOTLAR RO‘YXATI:

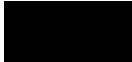



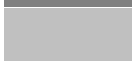











1. Мирзиёев Шавкат Миромонович. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. Ўзбекистон Республикаси Президенти лавозимидаги киришиш тантанали маросимида бағишланган Олий Мажлис палаталарининг кўшма мажлисидаги нутқ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2016. - 56 б.
2. Мирзиёев Шавкат Миромонович. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. Ўзбекистон Республикаси Конституцияси қабул қилинганининг 24 йиллигига бағишланган тантанали маросимдаги маъруза. 2016 йил 7 декабр /Ш.М.Мирзиёев. – Тошкент: “Ўзбекистон”, 2017. – 48 б.
3. Мирзиёев Шавкат Миромонович. Буюк келажакимизни мард ва олижаноб халкимиз билан бирга қурамиз. Мазкур китобдан Ўзбекистон Республикаси Президенти Шавкат Мирзиёевнинг 2016 йил 1 ноябрдан 24 ноябрга қадар Қорақалпоғистон Республикаси, вилоятлар ва Тошкент шаҳри сайловчилари вакиллари билан ўтказилган сайловолди учрашувларида сўзлаган нутқлари ўрин олган. /Ш.М.Мирзиёев. – Тошкент: : “Ўзбекистон”, 2017. – 488 б
4. Ўзбекистон Республикаси Президентининг Фармони. Ўзбекистон республикасини янада ривожлантириш бўйича ҳаракатлар стратегияси тўғрисида. (*Ўзбекистон Республикаси қонун ҳужжатлари тўплами, 2017 й., 6-сон, 70-модда*)
5. Бегимқулов У.Ш., М.Мамаражабов, Турсунов С.Қ. Flash дастури ва ундан таълимда фойдаланиш имкониятлари. Методик қўлланма. –Тошкент: ТДПУ, 2006. – 88 б.
6. Бегимқулов У.Ш., Турсунов С.Қ. Dreamweaver дастурида Web саҳифа яратиш. Методик қўлланма. – Тошкент: ТДПУ, 2006. – 72 б.
7. М.Мамаражабов, S.Tursunov. Kompyuter grafikasi va Web dizayn. o‘quv qo‘llanma. Toshkent, “Cho‘lpon”, 2013.
8. U.Yuldashev, M.Mamarajabov, S.Tursunov. Pedagogik Web dizayn. O‘quv qo‘llanma. Toshkent, “Voriz”, 2013.
9. M.Aripov, M.Fayziyeva, S.Dottayev. Web texnologiyalar. O‘quv qo‘llanma. Toshkent, “Faylasuflar jamiyati”. 2013.
10. Немцова Т. И., Назарова Ю. В. Компьютерная графика и web-дизайн. Учебное пособие. ИД "ФОРУМ", ИНФРА-М. 2011. –С. 288.
11. Тайлақов Н.И. Узлуксиз таълим тизими учун ўқув адабиётларини янги авлодини яратишнинг педагогик асослари: Пед. фан. док. ... дис.- Тошкент: ТДПУ, 2007. - 359 б.
12. Adobe Dreamweaver CS6 Professional. Официальный учебный курс; [пер. с англ. М. А. Райтмана]. – М. : Эксмо, 2013. – 496 с. : ил. + 1 CD-Rom. – (Официальный учебный курс).
13. Against The Clock. Adobe Dreamweaver 2020: The professional Portfolio Spiral-bound. Textbook. 2019. 406 pages.
14. D. Michael Ploor. Certification Prep Adobe Dreamweaver Creative Cloud First Edition, Student Textbook Edition. Goodheart-Willcox; First Edition, Student Textbook edition. 2017. 128 pages.

- 
15. Jennifer Niederst Robbins. Learning Web Design, Fifth Editions. A beginner's guide to HTML, CSS, JavaScript, and Web Graphics. Textbook. O'Reilly Media. Canada. 2018. 808 pages.
  16. Luiza A.C. Exploration guides for educational software: are they helpful? //Advances in technology - Based education: toward a knowledge-Based society. - Badajoz, Spain, 2003. – 658 p.
  17. Manuel Cebrián de la Serna. Tecnologías de la información y comunicación para la formación de docentes // Ediciones Pirámide. 2006. Universidad de Malaga. – 173 p.
  18. Patrick Mendelsohn. Livret stagiaires. Infos supplémentaires// académie de grenoble. Grenoble. 2005. – 176 p.
  19. Paul McFedries. Web Design Playground: HTML & CSS the Interactive Way 1st Edition, Textbook. Manning Publications; 1st edition, 2019. 440 pages.
  20. Ruiz Dávila, M. y otras (2004). Las TIC, un reto para nuevos aprendizajes. Usar información, comunicarse y utilizar los recursos, Madrid: Narcea.
  21. Software psychology. Human factors in computer and information systems. Ben Shneiderman. Associate Professor Department of Computer Science Cambridge, Massachusetts. 2006. – 278 p.
  22. [http://www.academiaxxi.ru/Meth\\_Papers/AO\\_recom\\_t.htm](http://www.academiaxxi.ru/Meth_Papers/AO_recom_t.htm) - Рекомендации по созданию электронного учебника.
  23. <http://www.ziyonet.uz> – Ахборот таълим портали.
  24. <http://www.webmascon.com/topics/imho/2a.asp> - Что же такое web-дизайн?
  25. [www.tdpu.uz](http://www.tdpu.uz) – Nizomiy nomidagi TDPU rasmiy sayti.

## Web-dasturlashda ishlatiladigan ranglar nomlari va ularning kodlari

### Asosiy ranglar jadvali

Quyidagi jadvalda HTMLda ishlatiladigan asosiy ranglar keltirilgan. Ularning barchasi videokartaning 256-rangli rejimida ham aniq tasvirlanadi. Ammo bu ranglar professional darajali dizayndagi saytlarni yaratish uchun yetarli emas.

|                                                                                   |         |                                                                                   |        |
|-----------------------------------------------------------------------------------|---------|-----------------------------------------------------------------------------------|--------|
|  | Black   |  | Navy   |
|  | Gray    |  | Blue   |
|  | Silver  |  | Aqua   |
|  | White   |  | Green  |
|  | Red     |  | Lime   |
|  | Fuchsia |  | Teal   |
|  | Maroon  |  | Yellow |
|  | Purple  |  | Olive  |

Bu ranglarning kombinatsiyasidan har qanday boshqa ranglarni hosil qilish mumkin. Pastda keltirilgan 216 ranglar «havfsiz» hisoblanib (browser-safe) 256-rangli rejimdagi monitorlardagi brauzerlarda ham yaxshi tasvirlanadi. Aynan shu ranglar dizaynerlar tomonidan keng qo'llaniladi. Quyidagi 3 ta jadvalda ushbu ranglarning 216 tadan ranglari joylashtirilgan.

## Qizil va Yashil – ko‘k qo‘shilgan

|        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|
| 000000 | 000033 | 000066 | 000099 | 0000cc | 0000ff |
| 330000 | 330033 | 330066 | 330099 | 3300cc | 3300ff |
| 660000 | 660033 | 660066 | 660099 | 6600cc | 6600ff |
| 990000 | 990033 | 990066 | 990099 | 9900cc | 9900ff |
| cc0000 | cc0033 | cc0066 | cc0099 | cc00cc | cc00ff |
| ff0000 | ff0033 | ff0066 | ff0099 | ff00cc | ff00ff |
| 003300 | 003333 | 003366 | 003399 | 0033cc | 0033ff |
| 333300 | 333333 | 333366 | 333399 | 3333cc | 3333ff |
| 663300 | 663333 | 663366 | 663399 | 6633cc | 6633ff |
| 993300 | 993333 | 993366 | 993399 | 9933cc | 9933ff |
| cc3300 | cc3333 | cc3366 | cc3399 | cc33cc | cc33ff |
| ff3300 | ff3333 | ff3366 | ff3399 | ff33cc | ff33ff |
| 006600 | 006633 | 006666 | 006699 | 0066cc | 0066ff |
| 336600 | 336633 | 336666 | 336699 | 3366cc | 3366ff |
| 666600 | 666633 | 666666 | 666699 | 6666cc | 6666ff |
| 996600 | 996633 | 996666 | 996699 | 9966cc | 9966ff |
| cc6600 | cc6633 | cc6666 | cc6699 | cc66cc | cc66ff |
| ff6600 | ff6633 | ff6666 | ff6699 | ff66cc | ff66ff |
| 009900 | 009933 | 009966 | 009999 | 0099cc | 0099ff |
| 339900 | 339933 | 339966 | 339999 | 3399cc | 3399ff |
| 669900 | 669933 | 669966 | 669999 | 6699cc | 6699ff |
| 999900 | 999933 | 999966 | 999999 | 9999cc | 9999ff |
| cc9900 | cc9933 | cc9966 | cc9999 | cc99cc | cc99ff |
| ff9900 | ff9933 | ff9966 | ff9999 | ff99cc | ff99ff |
| 00cc00 | 00cc33 | 00cc66 | 00cc99 | 00cccc | 00ccff |
| 33cc00 | 33cc33 | 33cc66 | 33cc99 | 33cccc | 33ccff |
| 66cc00 | 66cc33 | 66cc66 | 66cc99 | 66cccc | 66ccff |
| 99cc00 | 99cc33 | 99cc66 | 99cc99 | 99cccc | 99ccff |
| cccc00 | cccc33 | cccc66 | cccc99 | cccccc | ccccff |
| ffcc00 | ffcc33 | ffcc66 | ffcc99 | ffcccc | ffccff |
| 00ff00 | 00ff33 | 00ff66 | 00ff99 | 00ffcc | 00ffff |
| 33ff00 | 33ff33 | 33ff66 | 33ff99 | 33ffcc | 33ffff |
| 66ff00 | 66ff33 | 66ff66 | 66ff99 | 66ffcc | 66ffff |
| 99ff00 | 99ff33 | 99ff66 | 99ff99 | 99ffcc | 99ffff |
| ccff00 | ccff33 | ccff66 | ccff99 | ccffcc | ccffff |
| ffff00 | ffff33 | ffff66 | ffff99 | ffffcc | ffffff |

## Qizil va ko‘k – yashil qo‘shilgan

|        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|
| 000000 | 003300 | 006600 | 009900 | 00cc00 | 00ff00 |
| 330000 | 333300 | 336600 | 339900 | 33cc00 | 33ff00 |
| 660000 | 663300 | 666600 | 669900 | 66cc00 | 66ff00 |
| 990000 | 993300 | 996600 | 999900 | 99cc00 | 99ff00 |
| cc0000 | cc3300 | cc6600 | cc9900 | cccc00 | ccff00 |
| ff0000 | ff3300 | ff6600 | ff9900 | ffcc00 | ffff00 |
| 000033 | 003333 | 006633 | 009933 | 00cc33 | 00ff33 |
| 330033 | 333333 | 336633 | 339933 | 33cc33 | 33ff33 |
| 660033 | 663333 | 666633 | 669933 | 66cc33 | 66ff33 |
| 990033 | 993333 | 996633 | 999933 | 99cc33 | 99ff33 |
| cc0033 | cc3333 | cc6633 | cc9933 | cccc33 | ccff33 |
| ff0033 | ff3333 | ff6633 | ff9933 | ffcc33 | ffff33 |
| 000066 | 003366 | 006666 | 009966 | 00cc66 | 00ff66 |
| 330066 | 333366 | 336666 | 339966 | 33cc66 | 33ff66 |
| 660066 | 663366 | 666666 | 669966 | 66cc66 | 66ff66 |
| 990066 | 993366 | 996666 | 999966 | 99cc66 | 99ff66 |
| cc0066 | cc3366 | cc6666 | cc9966 | cccc66 | ccff66 |
| ff0066 | ff3366 | ff6666 | ff9966 | ffcc66 | ffff66 |
| 000099 | 003399 | 006699 | 009999 | 00cc99 | 00ff99 |
| 330099 | 333399 | 336699 | 339999 | 33cc99 | 33ff99 |
| 660099 | 663399 | 666699 | 669999 | 66cc99 | 66ff99 |
| 990099 | 993399 | 996699 | 999999 | 99cc99 | 99ff99 |
| cc0099 | cc3399 | cc6699 | cc9999 | cccc99 | ccff99 |
| ff0099 | ff3399 | ff6699 | ff9999 | ffcc99 | ffff99 |
| 0000cc | 0033cc | 0066cc | 0099cc | 00cccc | 00ffcc |
| 3300cc | 3333cc | 3366cc | 3399cc | 33cccc | 33ffcc |
| 6600cc | 6633cc | 6666cc | 6699cc | 66cccc | 66ffcc |
| 9900cc | 9933cc | 9966cc | 9999cc | 99cccc | 99ffcc |
| cc00cc | cc33cc | cc66cc | cc99cc | cccccc | ccffcc |
| ff00cc | ff33cc | ff66cc | ff99cc | ffcccc | ffffcc |
| 0000ff | 0033ff | 0066ff | 0099ff | 00ccff | 00ffff |
| 3300ff | 3333ff | 3366ff | 3399ff | 33ccff | 33ffff |
| 6600ff | 6633ff | 6666ff | 6699ff | 66ccff | 66ffff |
| 9900ff | 9933ff | 9966ff | 9999ff | 99ccff | 99ffff |
| cc00ff | cc33ff | cc66ff | cc99ff | ccccff | ccffff |
| ff00ff | ff33ff | ff66ff | ff99ff | ffccff | ffffff |



## Ko‘k va Yashil – qizil qo‘shilgan

|        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|
| 000000 | 000033 | 000066 | 000099 | 0000cc | 0000ff |
| 003300 | 003333 | 003366 | 003399 | 0033cc | 0033ff |
| 006600 | 006633 | 006666 | 006699 | 0066cc | 0066ff |
| 009900 | 009933 | 009966 | 009999 | 0099cc | 0099ff |
| 00cc00 | 00cc33 | 00cc66 | 00cc99 | 00cccc | 00ccff |
| 00ff00 | 00ff33 | 00ff66 | 00ff99 | 00ffcc | 00ffff |
| 330000 | 330033 | 330066 | 330099 | 3300cc | 3300ff |
| 333300 | 333333 | 333366 | 333399 | 3333cc | 3333ff |
| 336600 | 336633 | 336666 | 336699 | 3366cc | 3366ff |
| 339900 | 339933 | 339966 | 339999 | 3399cc | 3399ff |
| 33cc00 | 33cc33 | 33cc66 | 33cc99 | 33cccc | 33ccff |
| 33ff00 | 33ff33 | 33ff66 | 33ff99 | 33ffcc | 33ffff |
| 660000 | 660033 | 660066 | 660099 | 6600cc | 6600ff |
| 663300 | 663333 | 663366 | 663399 | 6633cc | 6633ff |
| 666600 | 666633 | 666666 | 666699 | 6666cc | 6666ff |
| 669900 | 669933 | 669966 | 669999 | 6699cc | 6699ff |
| 66cc00 | 66cc33 | 66cc66 | 66cc99 | 66cccc | 66ccff |
| 66ff00 | 66ff33 | 66ff66 | 66ff99 | 66ffcc | 66ffff |
| 990000 | 990033 | 990066 | 990099 | 9900cc | 9900ff |
| 993300 | 993333 | 993366 | 993399 | 9933cc | 9933ff |
| 996600 | 996633 | 996666 | 996699 | 9966cc | 9966ff |
| 999900 | 999933 | 999966 | 999999 | 9999cc | 9999ff |
| 99cc00 | 99cc33 | 99cc66 | 99cc99 | 99cccc | 99ccff |
| 99ff00 | 99ff33 | 99ff66 | 99ff99 | 99ffcc | 99ffff |
| cc0000 | cc0033 | cc0066 | cc0099 | cc00cc | cc00ff |
| cc3300 | cc3333 | cc3366 | cc3399 | cc33cc | cc33ff |
| cc6600 | cc6633 | cc6666 | cc6699 | cc66cc | cc66ff |
| cc9900 | cc9933 | cc9966 | cc9999 | cc99cc | cc99ff |
| cccc00 | cccc33 | cccc66 | cccc99 | cccccc | ccccff |
| ccff00 | ccff33 | ccff66 | ccff99 | ccffcc | ccffff |
| ff0000 | ff0033 | ff0066 | ff0099 | ff00cc | ff00ff |
| ff3300 | ff3333 | ff3366 | ff3399 | ff33cc | ff33ff |
| ff6600 | ff6633 | ff6666 | ff6699 | ff66cc | ff66ff |
| ff9900 | ff9933 | ff9966 | ff9999 | ff99cc | ff99ff |
| ffcc00 | ffcc33 | ffcc66 | ffcc99 | ffcccc | ffccff |
| ffff00 | ffff33 | ffff66 | ffff99 | ffffcc | ffffff |

---

**O‘quv qo‘llanma**  
(o‘zbek tilida)

Texnik muharrir: **N.Islomova**  
Musahhah:

Bosishga ruxsat etildi: \_\_\_ / \_\_\_ 2020.  
Hajmi: 19 b.t. Nusxasi 2000 ta  
Buyutirma. Bahosi kelishilgan narhda

