

004
W33



X.N.ZAYNIDINOV, E.SH.NAZIROVA,
D.S.YAXSHIBAYEV, S.U.MAXMUDJANOV

WEB ILOVALARNI YARATISH



O'ZBEKISTON RESPUBLIKASI
AXBOROT VA KOMMUNIKASIYA TEXNOLOGIYALARINI
RIVOJLANTIRISH VAZIRLIGI

MUHAMMAD AL-XORAZMIY NOMIDAGI
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI

X.N.ZAYNIDINOV, E.SH.NAZIROVA,
D.S.YAXSHIBAYEV, S.U.MAXMUDJANOV

WEB ILOVALARNI YARATISH

O'zbekiston Respublikasi
Oliy va o'rta maxsus ta'lim vazirligi tomonidan
darslik sifatida tavsiya etilgan.

5330500 – Kompyuter injiniringi (“Kompyuter injiniringi”,
“AT-servisi” va “Multimedia texnologiyalari”),
5330300 – Axborot xavfsizligi ta'lim yo'nalishlari
talabalari uchun mo'ljallangan

Toshkent – 2020

UO'K: 004.946(075)

KBK: 32.973.202

W 33

X.N.Zaynidinov, E.Sh.Nazirova, D.S.Yaxshibayev, S.U.Maxmudjanov.
Web ilovalarni yaratish. (Darslik). – T.: «Aloqachi», 2020. – 348 b.

ISBN 978–9943–6394–7–8

Darslik web-ilova yaratishning asosiy tushunchalarini, HTML5 teglaridan foydalanib sahifalar yaratish, JavaScriptda funksiyalar, strukturalar, dinamik va statik web-saytlar bilan ishlash asoslarini yaratishga qaratilgan. Shu bilan birga Cookie larni o'rnatish va MySQL da ma'lumotlar bazasini yaratish va uni PHP bilan bog'lab web sahifalar yaratishga bag'ishlangan. Darslik 5330500 – Kompyuter injiniringi ("Kompyuter injiniringi", "AT-servisi" va "Multimedia texnologiyalari") va 5330300 – Axborot xavfsizligi ta'lim yo'nalishlari talabalari uchun mo'ljallangan.

UO'K: 004.946(075)

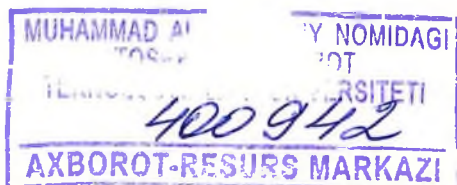
KBK: 32.973.202

Taqrizchilar:

- J.D.Jumanov** – Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti, "Kompyuter tizimlari" kafedrası mudiri
- A.M.Polatov** – Mirzo Ulug'bek nomidagi O'zbekiston Milliy universiteti, "Algoritmlar va dasturlash texnologiyalari" kafedrası professori

ISBN 978–9943–6394–7–8

© «Aloqachi» nashriyoti, 2020



MUNDARIJA

KIRISH.....	9
1-BOB. WEB DASTURLASHGA KIRISH. ASOSIY TUSHUNCHALAR. WWW RIVOJLANISH BOSQICHLARI.....	12
1.1. Web-sahifa, Web-sayt, Web-server, HTML, XML, XHTML, WML belgilash tillari	12
1.2. Web dasturlash texnologiyalari, ssenariyli tillar “Mijoz-server” texnologiyasi.....	14
1.3. WWW rivojlanish bosqichlari.....	16
NAZORAT SAVOLLARI.....	17
2-BOB. WEB LOYIHALASH.....	19
2.1. Loyihalash asosiy tushunchalari	19
2.2. Web loyihalash bosqichlari	21
2.3. Loyihalash shablonlari. Loyihalash shablon asosiy elementlari. Loyihalash shablonlari kategoriyalari	26
NAZORAT SAVOLLARI.....	38
3-BOB. HTML TILIGA KIRISH. HTML HUJJATNING STRUKTURASI. SARLAVHA ELEMENTLARI.....	39
3.1. HTML gipermatnli hujjatni belgilash tiliga kirish.....	39
3.2. Gipermatnli hujjat tuzilishi. Hujjat sarlavhasi (HEAD bo’limi) konteyner elementlari va atributlari.....	40
3.3. Meta ma’lumotlar.....	43
3.4. Gipermatnli hujjat. Giper murojaat.....	47
NAZORAT SAVOLLARI.....	52
4-BOB. HTML ASOSIY ELEMENT TEGLARI. JADVAL YORDAMIDA SAHIFALASH.....	53
4.1. HTML gipermatn tilining asosiy teglari: center, hr, br, h1, p, font....	53
4.2. Ro’yxatlar va ularning turlari.....	55
4.3. HTML da jadvallarni hosil qilish. Sahifalarni jadvallar yordamida verstka qilish.....	63
NAZORAT SAVOLLARI.....	73
5-BOB. HTMLDA FREYM SAHIFALASH, FORMA VA OB’YEKTLAR.....	74
5.1. HTML formalar va ularning atributlari. HTML forma ob’yektlari va atributlari.....	74
5.2. Freymlar. Freym tegi sintaksisi va atributlari. Sahifalarni freym yordamida hosil qilish	78
NAZORAT SAVOLLARI.....	81

6-BOB. HTML5 YANGI STANDARTLAR. TEG VA ATRIBUTLAR. GRAFIKA, AUDIO VA VIDEO BILAN ISHLASH. VIZUAL EFFEKTLAR	83
6.1. Yangi HTML5 razmetkani qo'llash. Yangi strukturali teg va atributlar	83
6.2. HTML5 standartlari yordamida takomillashgan forma interfeysini ishlab chiqish	85
6.3. Web sahifaga audio, video va vektor grafikalarini qo'shish	90
NAZORAT SAVOLLARI	94
7-BOB. CSS GA KIRISH. HTMLGA ULANISH USULLARI. SELEKTOR TURLARI. ELEMENTLAR VA SINFLAR. O'LCHOV BIRLIKLARI	95
7.1. CSS ga kirish. HTML ga ulanish usullari	95
7.2. CSS kaskadli jadvallar stillarida selektorlar	96
7.3. Psevdosinflar va psevdoelementlar	100
NAZORAT SAVOLLARI	105
8-BOB. CSS HUSUSIYATLARI. CSS SAHIFALASH	106
8.1. CSS hususiyatlari	106
8.2. Bloklarni joylashtirish usullari	111
NAZORAT SAVOLLARI	116
9-BOB. CSS3 KASKADLI JADVALLAR STILI HUSUSIYATLARI	117
9.1. HTML5 va CSS3 standart. CSS3 kategoriyalar bo'yicha yangi hususiyatlari	117
9.2. CSS3-transformasiyalar	120
NAZORAT SAVOLLARI	125
10-BOB. JAVASCRIPT ASOSLARI. O'ZGARUVCHILAR. MA'LUMOTLAR TURI. MASSIVLAR. OPERATORLAR. DASTURLASH ASOSLARI	126
10.1. JavaScript ssenariyli tili asoslari	126
10.2. O'zgaruvchilar va ma'lumotlar turi	127
10.3. JavaScript tili operatorlari. Unar operatori	130
NAZORAT SAVOLLARI	143
11-BOB. JAVASCRIPT BIRIKTIRILGAN FUNKSIYA VA OB'YEKTLARI. GRAFIKA VA DOIMIY IFODALAR	144
11.1. JavaScript da funksiyalar	144
11.2. JavaScriptda sikllar va ularning turlari	150
11.3. JavaScriptda grafika va doimiy ifodalar bilan ishlash	154
NAZORAT SAVOLLARI	157

12-BOB. JAVASCRIPTDA BRAUZER VA WEB HUJJAT OB'YEKTLARI MODELI BILAN ISHLASH	158
12.1. JavaScript da ob'yektlar	158
12.2. Brauzer ob'yektlari.....	159
NAZORAT SAVOLLARI	161
13-BOB. PHP ASOSLARI. O'ZGARUVCHILAR, KONSTANTALAR, MA'LUMOTLAR TURI. OPERATORLAR	162
13.1. PHP ilovalar komponentalari. PHP loyihalarni yaratish. PHP-kodni HTML faylga joylashtirish.....	162
13.2. PHP asosiy sintaksislar	164
13.3. PHP da o'zgaruvchilar, o'zgaruvchilar va operatorlar	166
13.4. PHP da ma'lumotlar turlari	170
13.5. PHP da massivlar	175
NAZORAT SAVOLLARI	178
14-BOB. PHP FUNKSIYALAR VA OB'YEKTLAR	179
14.1. PHP standart funksiyalar	179
14.2. PHP da grafik ma'lumotlar va GD kutubxona.....	186
14.3. PHP da GET va POST usullar	204
NAZORAT SAVOLLARI	206
15-BOB. PHP XATOLARNI QAYTA ISHLASH VA MUSTASNO QILISH 207	
15.1. Xatolarning roli va turlari.....	207
15.2. Jiddiy bo'lmagan xatolar	208
15.3. Jiddiy xatolar	209
NAZORAT SAVOLLARI	212
16-BOB. PHP COOKIE VA SESSİYALAR. O'ZGARUVCHILARNING KO'RINISH SOHALARI	213
16.1. Cookie ni o'rnatish va qabul qilish	213
16.2. Cookie ni o'qish	214
16.3. PHP da sessiyalarning ishlash mexanizmlari. Sessiya hosil qilish va o'chirish 215	
NAZORAT SAVOLLARI	218
17-BOB. PHP TILIDA FAYLLAR BILAN ISHLASH.....	219
17.1. Fayllar ustida amallar	219
17.2. Fayllardan o'qish.....	221
17.3. Fayllarga yozish	223
17.4. Kattaloglar bilan ishlash.....	227
NAZORAT SAVOLLARI	230
18-BOB. PHP SINFLARI	231

18.1. Sinf ma'lumotlar turi misolida	231
18.2. O'zgaruvchilarni initsiallashtirish	232
18.3. Sinflar bilan ishlash	234
NAZORAT SAVOLLARI	236
19-BOB. MA'LUMOTLAR BAZASI. MA'LUMOTLAR BAZASINI BOSHQARISH TIZIMI	237
19.1. Ma'lumotlar bazasi asoslari. Ma'lumotlar bazasini boshqarish tizimi (MBBT)	237
19.2. MySQL ma'lumotlar bazasi boshqarish tizimi	238
19.3. MySQL bilan ishlash interfeysi	239
NAZORAT SAVOLLARI	240
20-BOB. MA'LUMOTLAR BAZASIGA SQL SO'ROVLAR TILI ..	241
20.1. Jadval yaratish	241
20.2. Jadvallar uchun cheklanishlar	243
20.3. SQL tilida maydon qiymatlarini tekshirish (CHECK cheklanishi)	245
NAZORAT SAVOLLARI	247
21-BOB. WEB SAYTLARNI YARATISHDA CMS TEXNOLOGIYASINI QO'LLASH	248
21.1. CMS lar turlari va ularning paydo bo'lishi	248
21.2. CMS larning afzalliklari va qiyinchiliklari	253
21.3. CMS texnologiyasi - Joomla da sayt yaratish	256
NAZORAT SAVOLLARI	266
22-BOB. FREYMVORK PLATFORMADA WEB DASTURLASH ..	267
22.1. Framework dasturlash asoslari	267
22.2. Symfony, Laravel va Yii freymworklar	268
22.3. Freymwork platformada ma'lumotlar ba'zasi bilan ishlash	274
NAZORAT SAVOLLARI	279
23-BOB. WEB DASTURI MUHARRIRI ILOVALARI DREAMWEAVER MISOLIDA	280
23.1. Macromedia Dreamweaver 8 dasturi	280
23.2. Macromedia Dreamweaver 8 dasturining ishchi muhiti	280
NAZORAT SAVOLLARI	296
24-BOB. WEB SERVERNI SOZLASH. XAMPP, DENVER, WAMPP, OPEN SERVER PAKETLAR	297
24.1. XAMPP – krossplatformali web – server jamlanmasi	297
24.2. Denwer dasturi bilan ishlash	300
24.3. Open Server ni o'rnatish	306
NAZORAT SAVOLLARI	311
25-BOB. WEB DASTURLASH USULLARI	312

25.1. Fayllarni serverga yuklash	312
25.2. PDF hujjatlarni hosil qilish	314
25.3. Model-View-Controller (Model-Ko'rinish-Nazorat)	317
NAZORAT SAVOLLARI	319
26-BOB. SAYTNI RASKRUTKA QILISH. COAP, GOOGLE ANALITIKSDAN FOYDALANISH	320
26.1. DNS xizmat	320
26.2. Hosting xizmat xillari.....	325
26.3. Web analitika. Google Analytics dan foydalanish.....	327
NAZORAT SAVOLLARI	343
GLOSSARIY	344
FOYDALANILGAN ADABIYOTLAR RO'YHATI.....	348

KIRISH

Hozirgi kunda axborot texnologiyalarining jadal suratlar bilan rivojlanib borishi va uni inson faoliyatining barcha sohalariga kirib borishi kuzatilmoqda. Internetga kirish axborot xazinasiga kirishdek, foydalanuvchilarga cheksiz bilim qatlamini ochib beradi va qiziqarli onlarga, yaxshi tadqiqotlarga ega bo'lgan foydalanuvchilar uchun juda katta keng miqyosdagi imkoniyatlar beradi. Axborot texnologiyalarida web texnologiyalari vositalari Web-sahifa va ma'lumotlar bazasi, Web-interfeyslarni dasturlash uchun JavaScript, PHP vositalarini amalga oshirish eng dolzarb masalalardan hisoblanadi.

Fan va texnikaning rivojlanishi bilan inson faoliyati chegarasi nihoyatda kengayib bormoqda, auditoriyaga o'qitish imkoniyatlari keng bo'lgan yangi texnologiyalar kirib kelmoqda.

Internet bu yagona standart asosida faoliyat ko'rsatuvchi jahon global kompyuter tarmog'idir. U mahalliy (lokal) kompyuter tarmoqlarni birlashtiruvchi axborot tizimi bo'lib, o'zining alohida axborot maydoniga ega bo'lgan virtual to'plamdan tashkil topadi. Internet tarmog'i, unga ulangan barcha kompyuterlarning o'zaro ma'lumotlar almashish imkoniyatini yaratib beradi.

Ushbu darslik web ilovalarni yaratishda foydalaniladigan asosiy va zamonaviy dasturlash tillari, ularning tuzilishi va ulardan foydalanish metodlari, har bir dasturning afzalligi va kamchiligi, rivojlanish tendensiyasi, istiqboli hamda respublikamizdagi bu borada olib borilayotgan ishlar va olinayotgan natijalar web ilovalarni yaratish sohasida foydalaniladigan dasturlash tillari istiqboliga ta'siri masalasini qamraydi.

Web ilovalarni yaratish darslining maqsadi har bir mutaxassis o'z sohasida qo'llashi lozim bo'lgan axborot-kommunikatsiya texnologiyalaridan foydalanib, turli xil saytlar yaratish va unga ishlov berish imkoniyatlari haqidagi bilimlarga ega bo'lishi va ulardan foydalanish uchun ko'nikma va malakalarni shakllantirish va rivojlantirishdan iborat.

Darslik 26 ta bobdan iborat bo'lib, har bir bobda mavzular misollar bilan yoritib berilgan va har bir bob so'ngida mavzuga doir nazorat savollari berilgan.

Birinchi bobda dinamik Web sahifalar strukturasi bag'ishlangan bo'lib, Web-sahifa, Web-sayt, Web-server, ssenariyli tillar, WWW

rivojlanish bosqichlari, PHP, MySQL, JavaScript va CSS haqida umumiy tushunchalar berib o'tilgan.

Ikkinchi bob web loyihalashga bag'ishlangan bo'lib, loyihalashning asosiy tushunchalari, web loyihalash bosqichlari, loyihalash shablonlari, elementlari va shablonlarning kategoriyalari yoritib berilgan.

Uchinchi, to'rtinchi, beshinchi va oltinchi boblarda, Html hujjatning strukturasi, hujjat sarlavhasi konteyner elementlari va atributlari, meta ma'lumotlar, gipermurojaatlar, Html tilining asosiy teg elementlari, ro'yhatlar va ularning turlari, Html da jadvallar hosil qilish, Html da freymlar, forma va ob'yektlar bilan ishlash, Html5 ning yangi standartlari, teg va atributlari, grafika, audio va video teglari yoritib berilgan.

Ettinchi, sakkizinchi va to'qqizinchi boblarda CSS, CSS3 kaskadli jadvallar stillari bilan ishlash, psevdosinflar va psevdoelementlar, CSS3 da transformasiyalar bo'icha ma'lumotlar yoritib berilgan.

O'ninchi, o'n birinchi va o'n ikkinchi boblarda JavaScript ssenariyli tili hamda funksiyalari, operatorlari, ob'yektlar va grafikalar bilan ishlash yoritib berilgan.

O'n uchinchi, o'n to'rtinchi va o'n beshinchi boblar PHP va asosiy sintaksislari, ma'lumotlar turlari, PHP boshqaruv elementlari va funksiyalari, PHP da grafik ma'lumotlar va GD kutubxona, PHP da Get va Post usullar, PHP da xatolarni qayta ishlash va ularning turlari bilan ishlashga bag'ishlangan.

O'n oltinchi, o'n ettinchi va o'n sakkizinchi boblar COOKIE va SESSI larni sozlash, o'qishga, PHP tilida fayllar bilan ishlash, ularning turlari, fayllarni o'qish va yozish, kataloglar bilan ishlash va PHP tilida sinflar bilan ishlashga bag'ishlangan.

O'n to'qqizinchi va yigirmanchi boblar MySQL ma'lumotlar bazasi bilan, jadvallar yaratish va SQL so'rovlarga bag'ishlangan.

Yigirma birinchi bob web saytlar yaratishda CMS texnologiyalarni qo'llash, CMS lar turlari va ularning paydo bo'lishi, afzalliklari va kamchiliklariga bag'ishlangan.

Yigirma ikkinchi bobda Symfony, Laravel va Yii freymworklar bilan va freymwork platformada ma'lumotlar bazasi bilan ishlash yoritib berilgan.

Yigirma uchinchi va yigirma to'rtinchi boblarda Macromedia Dreamweaver dasturi, uning ishchi muhiti, web serverlarni sozlash, ya'ni XAMPP, DENVER, WAMPP, OPEN SERVER paketlar bilan ishlash yoritib berilgan.

Yigirma beshinchi bob web dasturlash usullari, fayllarni serverga joylash, PDF hujjatlarni hosil qilish va Model-View-Controller larga bag'ishlangan.

Yigirma oltinchi bob esa DNS xizmat, Hosting xizmat xillari va web analitika hamda Google analitiksdan foydalanishga bag'ishlangan.

Dasrlik talabalarga HTML, HTML5 gipermatn tili asoslari, CSS, CSS3 stillar majmuasi, web-sahifa va ma'lumotlar bazasiga web-interfeyslarni dasturlash uchun javascript, php dasturlash tillarini o'rganishda yordamchi qo'llanma bo'lib xizmat qiladi.

1-BOB. WEB DASTURLASHGA KIRISH. ASOSIY TUSHUNCHALAR. WWW RIVOJLANISH BOSQICHLARI

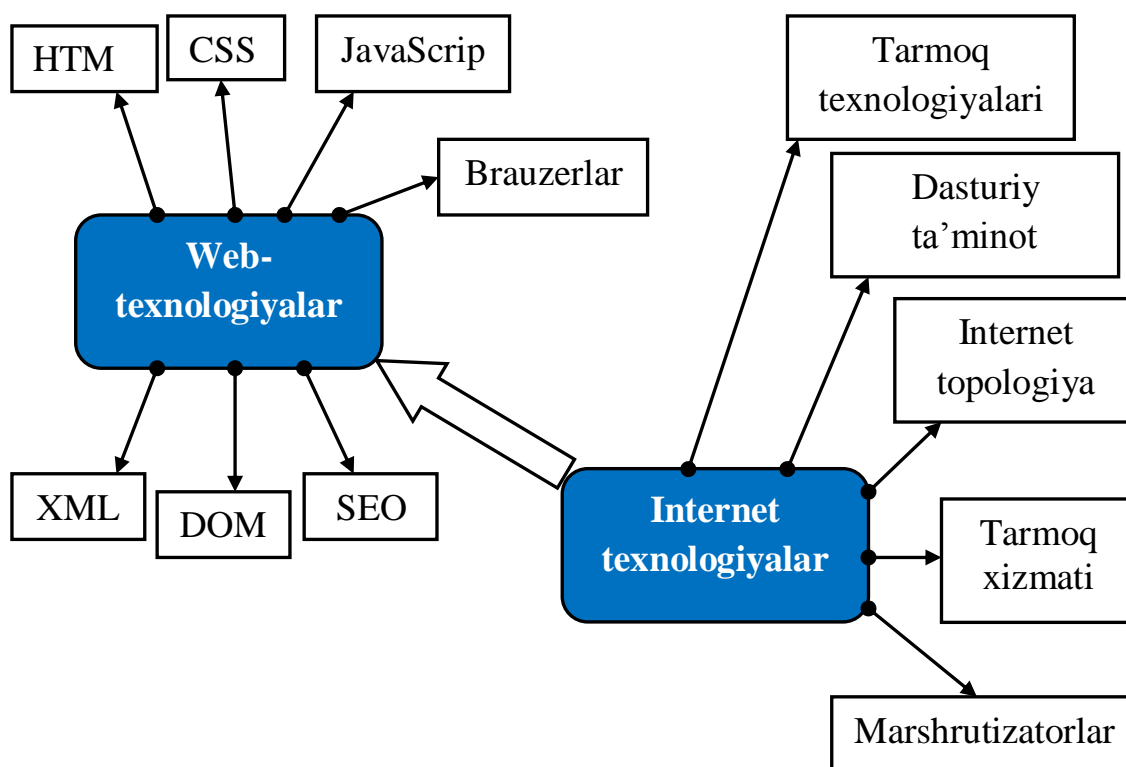
1.1. Web-sahifa, Web-sayt, Web-server, HTML, XML, XHTML, WML belgilash tillari

Web-texnologiyani (Internet-texnologiya) o'rganishni Web-dizaynning quyidagi uchta tushunchasini o'rganishdan boshlaymiz: Web-sahifa, Web-sayt va Web-server.

Web-sahifa – o'zining unikal manziliga 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.

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.



1.1-rasm. Web-texnologiya sinfsifikatsiyasi

Biz Internet tarmog'idagi Web-sahifalarni ko'rishimiz uchun WWW (World Wide Web) deb ataluvchi Internet xizmatdan foydalanamiz.

World Wide Web (WWW, Butun dunyo o'rgimchak to'ri) – bu mijoz-server texnologiyasi asosida tashkil etilgan, keng tarqalgan Internet xizmatidir.

Web-texnologiyaning (Internet-texnologiya) Web-dizayn qismini o'rganishni belgilili til tasnifi bilan boshlaymiz.

Bu til yordamida matnlar, grafik ma'lumotlar Web-sahifa hujjatga joylashtiriladi va bu hujjatni barcha kompyuterda ko'rish imkoniyati mavjuddir. Bunday maxsus tillar belgili tillar deb ataladi. Ularning asosiy vazifasi – Web-sahifaga “ma'lumotlarni joylashtirish” va ular orasidagi aloqani (giperurojaatlar) ta'minlashdan iborat.

HTML (HyperText Markup Language). Dastlab World Wide Web tizimi matnli ma'lumotlarni va HTML hujjatlarni ko'rishga mo'ljallangan, matnni tahrirlovchi tilga o'xshash tizim bo'lgan. Ayni damda HTML tili WWW dagi eng ommabop tillardan biri hisoblanadi. HTML tilida yozilgan ma'lumotlar o'z ichiga matnli fayllar, grafik ma'lumotlar va boshqalarni oladi.

Hujjatlar orasidagi aloqani ta'minlash va ma'lumotlarni formatlash vositalari teg (tag) deb ataluvchi vosita orqali amalga oshiriladi.

Web-sahifaning matn va teglari aralash ravishda HTML-hujjat deb ataluvchi faylining ichiga joylashtiriladi. Qanday tegni qo'llaganingizga qarab brauzer oynasida ma'lumotlar turlicha ko'rinadi. HTML hujjatga ma'lumotlarni joylashtirish va tahrirlash uchun yuzlab teglar mavjud. Masalan, <p> va </p> teglari xat boshini tashkil etadi, <i> va </i> juft teglari esa, matnni yozma (kursiv) holda ko'rsatish uchun qo'llaniladi. Shu bilan birga gipermatnli xavola (ssilka)lar teglari ham mavjud. Ushbu elementlar foydalanuvchiga gipermatn ustiga sichqoncha kursori bosilganda boshqa hujjatga bog'lanish imkonini beradi.

XML (eXtensible Markup Language). XML tili ham HTML tiliga o'xshash til hisoblanadi. HTML dan farqli tomoni shundaki, XML da dasturchi o'zining shaxsiy teglarini yaratadi va ular orasiga ma'lumotlar joylashtiradi. XML-teglar harflar katta kichikligini farqlaydi.

XHTML (eXtensible HyperText Markup Language). XHTML tili HTML va XML tillarining birlashmasini tashkil etadi. XHTML tilida yozilgan hujjatning tashqi ko'rinishi platformaga bog'liq (Windows, Mac yoki Unix) ravishda o'zgarib ketmaydi. Shunga qaramay XHTML tarkibida HTML diskriptorlardan foydalaniladi.

Bugungi kunda mobil aloqa vositalaridan foydalanuvchilar uchun yangi til ishlab chiqilgan bo'lib, u WML (Wireless Markup Language) deb ataladi; CDF (Channel Definition Format) - Microsoft ishlab chiqqan brauzerlarda push-kanal hosil qilishda qo'llaniladi.

1.2. Web dasturlash texnologiyalari, ssenariyli tillar “Mijoz-server” texnologiyasi

Hozirda Web-sahifaning rivojlanishi yanada interaktiv pog'onaga chiqqan. Web-saytlar asta sekinlik bilan ilovalar interfeysiga o'xshab bormoqda. Bularning barchasi zamonaviy Web-dasturlash texnologiyasi yordamida amalga oshmoqda.

Web-dasturlash texnologiyalarini, dasturlarini asosan ikkita qismga ajratish mumkin: mijoz tomonidagi dasturlarlash (client-side) va server tomonidagi (server-side). Ushbu texnologiyalarni tushunish uchun avvalo bevosita "mijoz-server" texnologiyasini tushunish kerak. Web-sahifaning interaktiv dasturi ssenariy deb ataladi. Bunday atama dasturning natijasiga bog'liq holda vujudga kelgan. Uning asosiy vazifasi Web-sahifada foydalanuvchi holatiga, xarakteriga «reaksiya» berishdir. Shu tariqa ssenariylar mijoz tomonida bajariluvchi va server tomonida bajariluvchi ssenariylarga bo'linadi. Mijoz tomonida bajariluvchi ssenariylar brauzer yordamida bajariladi. Server tomonida bajariluvchi ssenariylar esa Web-server yordamida bajariladi.

Mijoz tomonidagi ssenariylar. Mijoz tomonidagi ssenariylar foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni to'g'riligini serverga murojaat qilmasdan tekshiradi. Ko'p hollarda bu ssenariylar JavaScript va VBScript tillarida yoziladi.

JavaScript (Dasturlash tili). JavaScript – bu til Netscape va Sun Microsystems tomonidan yaratilgan bo'lib, Web-sahifaning funksional imkoniyatlarini orttirish maqsadida qo'llaniladi.

JavaScript yordamida odatda ma'lumotli va muloqot oynalarini chiqarish, animatsiyalarni ko'rsatish kabi vazifalarni bajarish mumkin. Bundan tashqari, JavaScript-ssenariy ba'zan o'zi ishlab turgan brauzer va platforma turini aniqlashi mumkin. JavaScript-ssenariylar foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni to'g'riligini tekshirishda ham qulay hisoblanadi.

VBScript tili. 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 til hisoblanib, Microsoft ning Web-texnologiyalari bilan hamkorlikda ishlay oladi, masalan ASP (Active Server Page). Shunga qaramay VBScript mijoz tomonida ishlovchi ssenariy hisoblanadi, ASP esa server tomonida ishlaydi.

Server tomonidagi ssenariylar. Server tomonida bajarilishi kerak bo'lgan ssenariylar odatda sayt papkasining ichidagi maxsus papkaga joylashtiriladi. Foydalanuvchi so'roviga asosan server bu ssenariyni bajaradi. Bajarilgan ssenariy natijasi web-serverga uzatiladi va undan so'ng mijozga uzatiladi. Server tomonidagi ssenariylarni tashkil etish uchun odatda Perl, ASP, PHP, JSP va SSI kabi til va texnologiyalardan foydalaniladi.

Perl interpretatorli tili. Perl tili Web-illovalar yaratishda eng ommabop tillardan biri hisoblanadi. Matnlarni qidirish va tahrirlash, fayllar bilan qulay ishlay olish qoidalari bilan Perl tili Internetning asosiy tillaridan biri bo'lib qoldi. Perl – interpretatorli til hisoblanadi, shu bois unda yaratilgan ssenariylar 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 biri, uning barcha platformalar uchun ishlay olishi va uning barcha resurslari bepul tarqatilayotganligidir. Ko'pgina Web-serverlar UNIX da ishlaydi, Perl interpretator esa bu operatsion tizimning bir qismi hisoblanadi.

ASP (Active Server Pages). ASP-ma'lumotlar bazalari tashkil etish va ular bilan ishlash vazifalarini bajarishga moslashuvchan, qulay vositadir. ASP vositalari server tomonida ishlaydi va HTML-kod va ssenariylar kabi fayllarni qayta ishlaydi. ASP texnologiyasi VBScript, Java va JavaScript tillarini qo'llab quvvatlaydi. ASP-kod ixtiyoriy HTML-hujjatdan, shu bilan birga boshqa ASP-hujjatdan chaqirilishi mumkin. 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 ASP ni qo'llab quvvatlash imkoniyatlarini kiritmoqdalar. Ko'pgina HTML-muharrirlar, masalan Adobe GoLive ham ASP ni qo'llab quvvatlaydi. ASP texnologiyasi bir

nechta qulayliklarni o'zida jamlagan: HTML-hujjatni dinamik generatsilaydi, formalarni qo'llab quvvatlaydi, ma'lumotlar bazasiga ruxsatni tashkil etadi va u bilan ishlay oladi. ASP – dasturlash tili ham, ilova ham emas, u interaktiv Web-sahifa hosil qilish texnologiyasi.

PHP (Personal home page). PHP – bu serverda qayta ishlanuvchi ssenariylar tilidir. ASP kabi PHP kodlar ham bevosita HTML-hujjatni tarkibiga qo'shiladi. Ushbu tilning nomi Personal Home Page Tools so'zlarining qisqartmasidan olingan. PHP da 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.

JSP (JavaServerPage). JSP texnologiyasi o'zining funksional imkoniyatlariga ko'ra ASP ga o'xshashdir. Asosiy farqi shundaki, bunda VBScript va JavaScript bilan birga Java tili ham qo'llanila oladi. Shunga qaramay JSP Java dan oldinroq qo'llanilgan va ushbu texnologiya mukammal Web-illovalar yaratish uchun yetarli imkoniyatga ega.

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

1.3. WWW rivojlanish bosqichlari

1992 - 93-yillarda axborot texnologiyasining rivojlanishi sababli tasviriy va tovushli axborotlarni olis masofalardan qisqa vaqtda uzatishning shunday imkoniyati yaratilganki, u World Wide Web deb nomlangan.

World Wide Webning yaratilishiga 1989-yil Shvesariyadagi Yevropa Yadroviy Tadqiqotlar Kengashining loyihasi asos bo'ldi. Bu loyihaning maqsadi Internetda axborot tarqatishning samarali usullarini izlash va uning oqibatlarini kuzatishdan iborat edi. Hozirgi kunda World Wide Web Internetning eng tez rivojlanayotgan sohalaridan biri bo'lib qoldi.

Internet deganda ko'pchilik World Wide Web (qisqacha Web yoki WWW) ni tushunadi. Aslida World Wide Web Internetning bir qismi bo'lib, xalqaro o'rgimchak to'ri ma'nosini anglatadi. World Wide Web multimedia (multimedia — rasm va matnli axborotni tovushli va

harakatdagi shakllardan iborat axborot bilan birlashtirish texnologiyasi) imkoniyatlariga ega bo'lgani uchun foydalanuvchilar e'tiborini juda tez qozondi.

WWW tarkibi Web-sahifaga matn, rasm, tovush, videotasvir va hokazo ko'rinishdagi axborotlarni joylashtirish mumkin. Bu esa o'z navbatida reklama, tijorat, ta'lim va boshqa ko'pgina soha vakillariga beqiyos imkoniyatlar ochib berdi. Masalan, juda ko'p kinostudiyalar o'z mahsulotlarini reklama qilish uchun Web-sahifalar yaratishadi. Mazkur Web-sahifalarda asosan yangi filmlar haqidagi ma'lumotlar bilan birga shu filmlardan 1-2 daqiqali parchalar aks ettiriladi. WWW yaratilgunga qadar bunday imkoniyat faqat kinoteatrlar yoki televideniya orqaligina mumkin edi.

Kinoteatr va televideniya orqali namoyishlar belgilangan vaqtga bog'liq bo'lsa, WWW dan xohlagan kishi istalgan vaqtda yangi filmlar haqida to'liq ma'lumot olishi mumkin.

WWWning ommalashishiga yana bir omil gipermatndir. Gipermatn Web-sahifaning biror qismiga yoki boshqa Web-sahifaga bog'liqligini ko'rsatuvchi ilova bo'lib, u so'z yoki rasm bo'lishi mumkin. Gipermatn yordamida Web-sahifaning kerakli qismiga yoki boshqa Web-sahifaga tez va oson o'tiladi, bu hol kerakli qism yoki Web-sahifaga gipermurojaat deb ataladi.

Bitta tashkilot yoki xususiy shaxsga tegishli va mazmuniga ko'ra o'zaro bog'langan bir necha Web-sahifalar majmuyi Web-sayt deyiladi. Web-saytni kitobga, Web-sahifani esa kitobning sahifasiga o'xshatish mumkin. Web-saytdagi Web-sahifalar o'zaro gipermatn yordamida bog'lanadi. Web-saytlar ham, Web-sahifalar ham Web-server deb ataluvchi Internetga ulangan maxsus kompyuterlarda saqlanadi va o'z manziliga ega bo'ladi. Bu manzil URL deb ataladi. URL hamisha `http://` yozuvidan boshlanadi (`http` — Hyper Text Transfer Protocol). So'ngra Web-sahifa joylashgan tarmoq manzili (masalan, `www.uzsci.net`), keyin Web-sahifa nomi (masalan, `~rtm`) yoziladi. Provayderlar (ingliz. provide — imkon berish, ta'minlash) — Internetning ba'zi qismlaridan foydalanishga imkon beruvchi va foydalanuvchilarga turli xizmatlarni ko'rsatuvchi firmalardir. Shunday qilib, misolda keltirilgan Web-sahifaning Internetdagi manzili `http://www.uzsci.net/~rtm` ko'rinishida bo'ladi.

NAZORAT SAVOLLARI

1. Internet texnologiya deganda nimani tushunasiz?

2. Qanday belgili tillarni bilasiz?
3. Mijoz-server texnologiyasini qanday tushunasiz?
4. WWW rivojlanish bosqichlari?
5. PHP, MySQL, JavaScript va CSS dan foydalanish afzalliklari?

2-BOB. WEB LOYIHALASH

2.1. Loyihalash asosiy tushunchalari

Web-texnologiyalar oddiy web-ilovalar kabi, murakkab axborotli web-tizimlarni ishlab chiqishga imkon beradi. Bunday ilovalarni muvaffaqiyatli ishlab chiqarish uchun dasturiy ta'minotni ishlab chiqish umumiy jarayonini tushunish talab etiladi (DT): bajarilishi kerak bo'lgan, amalga oshirish bo'yicha asosiy faoliyat; bu faoliyatlar o'zaro qanday bog'langanligi; ular bajarilishidagi tartib; bu jarayonda qanday mutaxassislar ishlashi kerakligi.

Ishlab chiqarish jarayoni muayyan chegaralar va ma'nolarga ega bo'lgan asosiy ishlar majmuasiga bo'linishi mumkin:

- muammoni tushunish;
- yechimni rejalashtirish;
- rejani bajarish;
- olingan natijaning aniqligini tekshirish;
- mumkin bo'lgan xatolar yoki noaniqliklarni olib tashlash maqsadida takomillashtirish.

Ixtiyoriy DT ishlab chiqish jarayonida quyidagi faoliyatlarning asosiy ko'rinishlarini belgilab olish mumkin:

- DT bo'lgan talabni belgilash (muhandislik talablari): yechilayotgan muammoni tushunish uchun mo'ljallangan.
- Loyihalashtirish: muammoni yechishni rejalashtirish uchun mo'ljallangan.
- Amalga oshirish: rejani ishchi dastur kodiga aylantirish.
- Sifatni baholash va tekshirish: kodlash xatolarini (dastur kodi) yoki muayyan talablar va ularni amalga oshirish o'rtasidagi kelishmovchiliklarni aniqlash uchun mo'ljallangan.
- Traqatish: foydalanuvchilarga yaratilgan dasturiy ta'minot bilan ishlash imkoniyatini taqdim etish.
- Qo'llab-quvvatlash: ishlaydigan tizimdan foydalanishni kuzatish va uni tezkor saqlash uchun mo'ljallangan.
- Rivojlanish: ishlab chiqilgan yechimni vaqt mobaynida yaxshilash uchun mo'ljallangan; yangi talablar shaklida ishlab chiqarish jarayoni uchun yangi kiruvchi ma'lumotlarni taqdim etish.

Muhandislik talabi yaratilayotgan DTning talab qilinayotgan imkoniyatlari va tavsifini tushunish uchun mo'ljallangan. Berilgan tahlil funksional talablarni (tizim qanday funksiyalarni bajarishi kerak) va

funksional bo'lmagan talablarni (taklif etilayotgan yechim sifati) aniqlash uchun yo'naltirilgan. Muhandislik talablari, ishlab chiqilayotgan tizim orqasidagi umumiy g'oyani aniqlashni ham o'z ichiga oladi; yangi tizim talab qiladigan asosiy manfaatdor tomonlar va tizimdan foydalanish shartlari. Belgilangan talablar ushbu tizimning yuqori darajadagi modellarini yaratish uchun ko'rib chiqiladi va bu muammoning kerakli maydonlarining keraksiz tafsilotlarini qisqartiradi.

Loyihalashtirish funksional talablarga va ishlash talablariga javob berishi kerak bo'lgan echimni, shuningdek, uning ishlashi mumkin bo'lgan muhitni cheklashlarini tasvirlash uchun mo'ljallangan. Avallroq to'olangan talablar texnologik cheklovlarni qondirish uchun aniqlanmoqda va takomillashtirilmoqda. Loyihalash quyidagi faoliyatlarni o'z ichiga oladi:

- Ma'lumotlar tuzilmasi va sinflarini loyihalashtirish.
- Komponentlarni loyihalashtirish.
- Grafik interfeysni loyihalashtirish.
- Arhitektura tizimini loyihalashtirish.

Tizimning strukturasi, xatti-harakatlari, ta'sir o'tkazish, ma'lumotlar va boshqarish oqimi kabi o'ziga xos xususiyatlarini yaxshiroq shakllantirishga yordam beradi. Mas'uliyat sohalari, dasturiy ta'minotni ishlab chiqarishning asosiy printsiplari - muammoni turli subtasklarga bo'lish yo'li bilan ajratishga imkon beradi - murakkablik bilan kurashishga va moslashuvchanlik, qo'llab-quvvatlashning soddaligi, kengaytirilishi va ko'p marta foydalanish kabi zarur texnik fazilatlarga erishishga yordam berishi mumkin.

Loyihani amalga oshirish bosqichida ishlab chiqilgan loyiha yechimlari tegishli dastur kodiga aylantiriladi (qo'lda yoki dasturlashning avtomatlashtirilgan asboblari yordamida). Dastur kutubxonasi, turli dasturlash tillari, turli xil kommunikasion protokollar va texnik qurilmalar talab qilinishi mumkin.

Tekshiruv va sifatni baholash, odatda, ishlab chiqarish bilan parallel ravishda amalga oshiriladi, chunki qidiruv natijalarning to'g'riligi va ishonchliligi faqat yakuniy mahsulot emas, balki butun dasturning sifatini kafolatlash uchun juda muhimdir. Sifat asosan quyidagi mezonlarga bog'liq:

- funksional baxolashga, ya'ni belgilangan funksional talablarga nisbatan ilovalar xatti-harakatlarning to'g'riligi;
- ishlab chiqarishdagi baxolashga, yani to'liq yuklamalarda odatiy sharoitlarda ilovalardan javob kutish vaqti;

- foydalanish qulayligini baholash (usability), ya'ni foydalanish qulayligi, aloqa samaradorligi va foydalanish standartlariga muvofiqligi.

Ilovani tarqatish foydalanuvchilarga ushbu dasturdan foydalanish imkonini beradi. Ilovaning turiga qarab, tarqatish jarayoni o'z ichiga quyidagilarni olishi mumkin:

- mijoz kompyuterlariga DT o'rnatish;
- serverga ma'lumotlar bazasi va markaziy ilovalarni o'rnatish;
- oraliq aloqa DT konfiguratsiyasi;
- bo'lajak foydalanuvchilarni o'rgatish va o'qitish, ayniqsa, agar mavjud bo'lgan ilovaning yangi versiyasi emas, balki butunlay yangi dastur o'rnatilgan bo'lsa.

O'rnatilgan va ishlaydigan ilovani qo'llab-quvvatlash, uning mavjudligini va kamchiliklarni kamaytirish kafolatidan tashkil topgan ishchi holatni ta'minlashni anglatadi. O'z ichiga olishi mumkin: jurnal fayllarini davriy tekshirish; xato haqidagi hisobot va vaqtinchalik fayllarni tozalash; xatolarni to'g'irlash; to'g'irlashni o'rnatish.

Ilovaning rivojlanishi. Ilova foydalanuvchilarning aniq vazifalarini yechish uchun mo'ljallangan. Biroq, tashkilotning rivojlanishi bilan bog'liq bo'lgan vazifalarni bajarishning murakkabligi, ushbu dasturiy ta'minotning imkoniyatlarini tushunishning yaxshilanishida, yaratilgan dasturiy ta'minotning rivojlanishi muqarrar ravishda talab qilinadi. Foydalanuvchilar tomonidan yangi talablar, ular yaratilgan DT bilan ayrim vaqt mobaynida ishlaganlaridan so'ngina paydo bo'ladi. Shundan so'ng ular o'zlarining taklif va izohlarini bera boshlaydilar. Yangi talablarning paydo bo'lishi ishlab chiqarishning barcha jarayonlarini qaytadan ishga tushirishga ilib kelishi mumkin. Dasturiy ta'minotni ishlab chiqish jarayonini to'g'ri tashkil etish bo'yicha tavsiyalarga qat'iy amal qilinishiga qaramay, ko'p hollarda dasturiy ta'minot bilan foydalanuvchilarning ayrim tajribalarini to'ldirish va to'plashdan so'nggina ba'zi talablar to'liq bajarilmaganligi va dasturni yaxshilash kerakligi aniq bo'ladi.

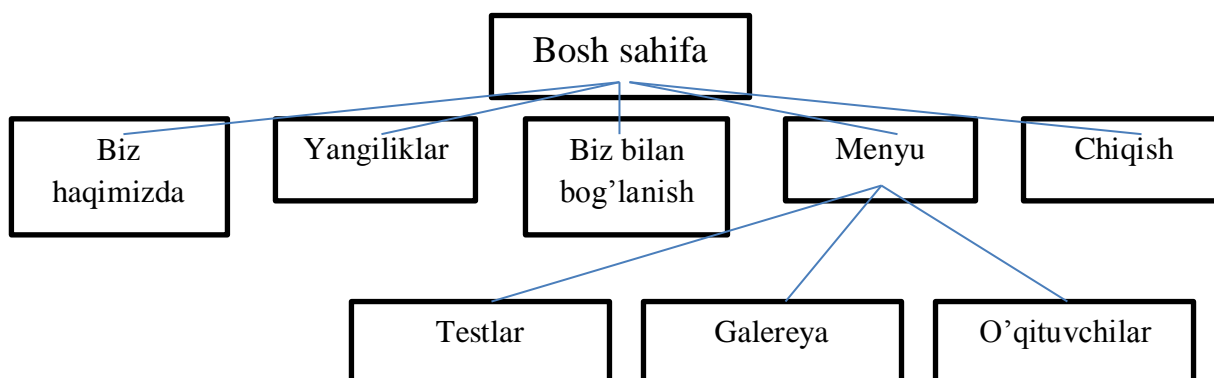
2.2. Web loyihalash bosqichlari

Umumiy qabul qilingan fikrlarga qaramasdan, web-saytlarni ishlab chiqish va dizayn jarayonida rivojlantirishning markaziy bosqichini har doim ham kod yozish fazasi egallamaydi. Avvalo, HTML, CSS va JavaScript kabi texnologiyalar haqiqatan ham biz o'rganib qolgan va axborot bilan o'zaro muloqot qilish usullarini aniqlaydigan tarmoq ko'rinishini yaratadi. Ko'pincha ko'zdan kechirilmaydigan, ammo ayni

paytda amalga oshirish jarayonining eng muhim qismi hisoblanadi, ya'ni bu ma'lumot to'plashning, mukammal loyihalashtirish dastlabko holati, hamda saytni ishga tushirgandan so'ng qo'llab-quvvatlash. Amalga oshirish jarayonidan tashkil topuvchi turli hil bosqichlar sonini belgilash mumkin. Odatda bu son beshtadan sakkiztagacha bo'ladi, ammo har bir holatda umumiy ko'rinish bir hil bo'lib qoladi. Keling o'rtacha qiymatda to'htaymiz. Demak, amalga oshirishning etti asosiy bosqichi: 1) Ma'lumotlarni to'plash, 2) Loyihalashtirish, 3) Dizayn, 4) Kontent yaratish, 5) Amalga oshirish, 6) Testlash, tadqiq va ishga tushirish, 7) Qo'llab-quvvatlash.

Web-saytni amalga oshirish hayotiy sikli.

1-bosqich. Ma'lumotlarni to'plash: belgilanishi, asosiy maqsad va maqsadli auditoriya.



2.1-rasm. Sayt xaritasini yaratish

Dastlabki tadqiqot va ma'lumotlar to'plash bosqichi, rivojlanishning keyingi bosqichlari qanday davom etishini aniqlaydi. Bu bosqichda eng muhim narsa – saytingizning kelajakdagi belgilanishi, uning yordamida qanday maqsadlarga erishmoqchi ekanligingiz hamda unga jalb qilmoqchi bo'lgan maqsadli auditoriyani aniq va to'liq tushunib olishdir. Web-amalga oshirishning bunday o'ziga hosligi kelajakda yahshiroq loyiha strategiyalarini rivojlanishini belgilab berishga imkon beradi. Yangiliklar uchun mo'ljallangan portallar ko'ngil ochar saytlardan, o'smirlar uchun mo'ljallangan saytlar esa kattalar uchun mo'ljallangan auditoriyalardan farq qiladi. Turli hil saytlar tashrif buyuruvchilarga turli hil funksional imkoniyatlarni taqdim etadi, shunday ekan bu yoki boshqa holatlarda turli texnologiyalar qo'llanilishi kerak. Bu bosqichda olingan ma'lumotlar asosida yaratilgan, batafsil tuzilgan reja, avval boshida nazarda tutilmagan

funksional imkoniyatlarni qo'shish yoki dizayn almashinuvi kabi kutilmagan qiyinchiliklarni yechishda qo'shimcha resurslar harajatlarini oldini olishi mumkin.

2-bosqich. Loyihalash: sayt maketi va xaritasini yaratish.

Amalga oshirishning ushbu bosqichida buyurtmachi kelajakdagi sayt qanday bo'lishi haqida tasavvurga ega bo'lishi mumkin. Dastlabki bosqichda to'plangan ma'lumotlar asosida sayt xaritasi yaratiladi (sitemap). Masalan quyidagi ko'rinishda bo'ladi:

Sayt xaritasi saytingizning turli qismlari o'rtasidagi o'zaro bog'lanishni tasvirlaydi. Bu foydalanishda qanchalik qulaylik bo'lishini tushunishga yordam beradi. Sayt xaritasida siz asosiy sahifadan boshqa sahifalarga "masofa" ni tanlashingiz mumkin, bu foydalanuvchi uchun qiziqarli bo'lgan ma'lumotlarga qanchalik osonlik bilan erishishini bilishga yordam beradi. Sayt xaritasini yaratishning asosiy maqsadi - oson navigatsiya va foydalanuvchilarga qulay mahsulot yaratishdir. Bu kelajakdagi sayt ichki strukturasi tushunishga imkon beradi, lekin saytning qanday ko'rinishini tasvirlamaydi. Ba'zan, kodni yozishni yoki dizaynni yaratishni boshlashdan oldin, mijozning roziligini olish juda muhim bo'lishi mumkin. Bunday hollarda maket yaratiladi (wireframe yoki mock-up). Mket o'zida kelajakdagi sayt interfeysining vizual ko'rinishini tasvirlaydi. Ammo, masalan, shablondan farqli ravishda, bu rang, logotip va boshqalar kabi dizayn elementlarini o'z ichiga olmaydi. U faqatgina sahifaga qaysi elementlar joylashishi va qanday joylashtirilishi haqida tasvirlaydi. Siz maket yaratish uchun mavjud onlayn-xizmatlardan foydalanishingiz mumkin.

Bundan tashqari, ushbu bosqichda qaysi stek texnologiyalar (dasturlash tili, freymworklar, CMS) ishlatilishini aniqlash kerak.

3-bosqich. Dizayn: sahifa shabloni, tadqiq va tasdiqlash.

Bu bosqichda web-sayt o'zining yakuniy shakliga yaqinlashib keladi. Tasvir, rasm va video kabi vizual kontentlar, aynan shu bosqichda yaratiladi. Va yana, loyihaning dastlabki bosqichida yig'ilgan barcha ma'lumotlar bu bosqichda ham juda muhimdir. Buyurtmachi qiziqishlari hamda to'liq auditoriya, birinchi navbatda dizayn ustida ishlash vaqtida hisobga olinishi kerak. Bu bosqichda dizayner tomonidan sahifa shabloni yaratiladi (page layout). Shablonning asosiy maqsadi – sahifa tuzilmasini vizuallashtirish, uning tarkibi hamda asosiy funkcionallikni aks ettirishdir. Bu safar, maketdan farqli ravishda, dizayn elementlaridan foydalaniladi. Shablon rang, logotip va tasvirdan tashkil topadi. U natija so'ngida tayyor sayt qanday ko'rinishda bo'lishini qaror qilishga imkon beradi. Yaratib

bo'lingandan so'ng shablon buyurtmachiga yuborilishi mumkin. Buyurtmachi olib borilgan ishlarni ko'rib chiqib, o'z mulohazalarini yuboradi. Dizaynning ayrim jihatlaridan qoniqmasa, mavjud shablonni o'zgartirib, uni mijozga qayta yuborishingiz kerak. Bu sikl buyurtmachi natijadan to'liq qoniqmagunga qadar takrorlanadi.

4-bosqich. Kontent yaratish.

Kontent yaratish jarayoni, odatda, boshqa rivojlanish bosqichlari bilan bir qatorda bajariladi va uning rolini ham e'tibordan chetda qoldirmaslik kerak. Ushbu qadamda, web-saytingiz tinglovchilarga etkazishi kerak bo'lgan narsalarning mohiyatini tavsiflashingiz kerak va CTA (harakatga chaqiruv) qo'shishingiz kerak. Bu bosqich o'z ichiga jozibali va qiziqarli sarlavhalarni, matnni kiritish va tahrirlash, mavjud matnlar kompilyasiyasi va boshqalarni oladi. Buning barchasi qo'shimcha vaqt va harakatlarni sarflashni talab etadi. Qoidaga ko'ra, mijoz saytga joylashtiriladigan barcha tayyor kontentni taqdim etadi. Barcha kontent amalga oshirish bosqichi vaqtida yoki undan avval tayoyrlanishi muhim.

5-bosqich. Joylashish va rivojlanish.

Endi siz to'g'ridan-to'g'ri sayt joylashishiga o'tishingiz mumkin. Avvaldan yaratilgan barcha grafik elementlar shu bosqichda foydalaniladi. Odatda birinchi navbatda bosh sahifa yaratiladi, undan so'ng sayt xaritasi yaratish bosqichida amalga oshirilgan, qolgan sahifalar iyerarhiya bilan mos holda qo'shiladi. Ushbu bosqichda yana CMS o'rnatish amalga oshiriladi. Dizayni shablon yaratishda amalga oshirilgan web-sahifaning barcha statik elementlari, web-sahifaning haqiqiy dinamik interaktiv elementlariga aylantiriladi.

6-bosqich. Testlash va ishga tushirish.

Testlash, ehtimol, amalga oshirishning eng muntazam qismidir. Har bir havola tekshirilgan bo'lishi kerak, har bir shakl va har bir skript testlangan bo'lishi kerak. Mumkin bo'lgan hatolarni oldini olish uchun matn orfografik tekshiruvchi dasturda tekshirilgan bo'lishi kerak. Kodni tekshiruvchilar avvalgi bosqichdagi kodning zamonaviy web-standartlarga to'liq mos kelishiga ishonch hosil qilish uchun foydalaniladi. Bu, masalan, o'zaro faoliyat brauzerlarning muvofiqligi siz uchun juda muhim bo'lsa, bu juda muhim bo'lishi mumkin. Shundan so'ng, saytni tekshirib va qayta tekshirib bo'lgach, uni serverga yuklash mumkin. Odatda buning uchun FTP-klient (mijoz) foydalaniladi. Serverga saytni yuklab bo'lgach, yuklash vaqtida kutilmagan xatoliklar yuzaga kelmagani va barcha fayllar to'liq va zararlanmaganiga ishonch hosil qilish uchun yana bir marta testdan o'tkazish kerak.

7-bosqich. Qo'llab-quvvatlash: foydalanuvchi sharhlari va muntazam yangilanishlar.

Web-sayt mahsulotga qaraganda ko'proq xizmat ekanligini tushunish muhimdir. Uni iste'molchiga "etkazib berish" etarli emas. Shunga ishonch hosil qilish kerakki, barcha rejalashtirilgandek ishlaydi, foydalanuvchilar esa oxirgi mahsulotdan qoniqish hosil qiladi. Kerak bo'lsa, tezda o'zgarishlar kiritishga tayyor turish kerak. Fikr-mulohaza tizimi sizga saytga tashrif buyuruvchilarning duch keladigan muammolarini aniqlash imkonini beradi. Bu kabi holatlarda eng muhim vazifa muammoni iloji boricha tezroq echishdir. Aks holda, sizning foydalanuvchilaringiz noqulayliklarga toqat qilishdan ko'ra, boshqa manbalardan foydalanishni afzal ko'radi. Shu bilan birga CMS ni muntazam ravishda yangilab turishni sedan chiqarmaslik kerak. Muntazam ravishda yangilab boorish sizni xatolar va xavfsizlikdagi muammolarda yordam beradi.

Bonus. Rivojlanishning asosiy bosqichlarining nazorat ro'yxati.

Siz hech narsa o'tkazib yubormadingiz va barcha rejalashtirilgan ishni o'z vaqtida bajarganingizga ishonch hosil qilish uchun ushbu ro'yxatni o'zingiz uchun belgilab oling:

1-bosqich. Ma'lumotlarni to'plash:

- Sayt uchun asosiy maqsad
- Maqsadli auditoriyani aniqlash

2-bosqich. Loyihalash:

- Sayt xaritasini yaratish
- Maketni yaratish
- Texnologik stekni tanlash (dasturlash tili, freymwork, CMS)

3-bosqich. Dizayn:

- Sahifa shablonini yaratish
- Yaratailgan shablonlarni taqqoslash
- Mijozdan taqrizlar olish
- Kerak bo'lganda o'zgartirishlar kiritish

4-bosqich. Kontent yaratish:

- Yangi kontentni yozish
- Kontentni saytga o'tkazishga tayyorlash

5-bosqich. Joylashish va rivojlanish:

- Saytni yaratish va rivojlantirish
- Interaktiv elementlarni qo'shish
- SEO optimallashtirish

6-bosqich. Testlash va ishga tushirish:

- Yaratilgan saytni testdan o'tkazish

- Serverga saytni yuklash
- Yakiniy testlash va ishga tushirish
- 7-bosqich. Qo'llab-quvvatlash:
 - Foydalanuvchilar taqrizlarini tizimga qo'shish
 - Qisqa muddatda xatoliklarni bartaraf etish
 - Saytni doimiy yangilab borish

Doimo esda tutish kerakki, web-saytni yaratish jarayoni kodni yozish bilan boshlanmaydi va sayt ishga tushirilgandan so'ng tugamaydi. Tayyorlash bosqichi barcha keyingi bosqichlarga ta'sir ko'rsatadi va loyiha ishining samaradorligi qanchalik samarali bo'lishini belgilaydi. Jins, yosh va oxirgi foydalanuvchilarning manfaatlari kabi jiddiy va chuqur tadqiqotlar juda muhimdir. Saytni qo'llab-quvvatlash uni ishga tushirgandan so'ng ham muhimdir. Siz yuzaga kelgan xatolarni tezda tuzatib, foydalanuvchilarning muammolarini hal qilish uchun tezkor bo'lishingiz kerak. Saytning rivojlanish bosqichlari orasida muhim ahamiyatga ega bo'lmagan yoki keraksiz deb hisoblanmaydigan narsalarni tushunish sizni ortiqcha muammolardan qochishga yordam beradi va loyihadagi ishlar sizning maqsadingizga muvofiq harakat qilayotganligini va to'liq nazoratga ega ekanligingizga ishonch hosil qiladi.

2.3. Loyihalash shablonlari. Loyihalash shablon asosiy elementlari. Loyihalash shablonlari kategoriyalari

Mustaqil loyihalashtirish bilan to'liq solishtirganda, shablonlar bir qator avfzalliklarga ega. Shablonlardan foydalanishning asosiy foydasi, butun muammolar sinfini yechish uchun tayyor abstraksiyalar hisobiga ishlab chiqarish qiyinchiliklarini kamaytirishdan iborat. Shablon, ma'lum shablonlarga murojaat qilish imkonini berish, ishlab chiqaruvchilar o'rtasidagi aloqani osonlashtirish uchun yechimga o'z nomini beradi. Shunday qilib, shablonlar hisobiga yechim detallarini unifikasiyasi ishlab chiqiladi: modullar, loyiha elementi, - hatolar soni kamayadi. Shablonlar ishlatish kontseptual ravishda tayyor kodli kutubxonalardan foydalanishga o'xshaydi. To'g'ri shakllantirilgan dizayn namunasi, muvaffaqiyatli yechim topib, uni qayta ishlatish imkonini beradi. Shablonlar to'plami ishlab chiqaruvchiga loyihalashning mos bo'lgan variantini tanlashga yordam beradi.

Loyihalash shablon turlari. Asosiylari.

Nomlanishi	Haqiqiy nomlanishi	Tavsif	Design Patterns da
------------	--------------------	--------	--------------------

			yozilgan
Asosiy shablonlar (Fundamental)			
Delegatsiya shablони	Delegation pattern	Ob'ekt tashqi ko'rinishda ba'zi xatti-harakatlarni ifodalaydi, lekin aslida bu xatti-harakatni tegishli ob'ektga bajarish uchun javobgarlikni o'z zimmasiga oladi.	h/y
Funksional dizayn shablonlari	Functional design	Kompyuter dasturining har bir moduli faqatgina bitta majburiyatga egaligini kafolatlaydi va uni dasturning boshqa qismlariga yomon ta'sirini kamaytirib bajaradi.	h/y
O'zgartirib bo'lmaydigan interfeys	Immutable interface	O'zgartirib bo'lmaydigan ob'yektni yaratish.	h/y
Interfeys	Interface	Kompyuter dasturlarini tushunishni osonlashtiradigan dasturlarni tuzishning umumiy usuli.	h/y
Interfeys-marker	Marker interface	Atribut sifatida (shahs ob'yektlarini belgilash kabi) marker interfeysi	h/y

		<p>ishlab chiqishning mavjudligi yoki yo'qligi qo'llaniladi. Buning o'rniga zamonaviy dasturlash tillarida atributlar yoki annotasiyalar qo'llaniladi.</p>	
Xususiyatlar konteyneri	Property container	<p>Yangi xususiyatlar bilan sinfni kengaytirmasdan, (sinf ichida) konteynerga sinf uchun qo'shimcha xususiyatlarni qo'shishga imkon beradi.</p>	h/y
Event Channel	Event channel	<p>Tadbirlar uchun markazlashgan kanalni yaratgan holda Publish/Subscribe shablonini kengaytiradi. Obuna uchun vakolatli ob'ektni va kanaldagi voqeani nashr etish uchun vakolatli ob'ektdan foydalanadi. Vakil haqiqiy noshir yoki abonentdan alohida mavjud bo'ladi. Abonent bir nechta ob'ektdan nashr</p>	h/y

		etilgan voqealarni, hatto bitta kanalda ro'yxatdan o'tgan bo'lsa ham, qabul qilishi mumkin.	
<p>Yaratilgan shablonlar (Creational) – mavhum jarayonlarning namunasini yaratuvchi loyiha shablonlari. Ular tizimni yaratilish usullari, tarkibi va ob'yektlarni taqdim etishda mustaqil bo'lishiga imkon beradi. Sinflarni ishlab chiqaradigan shablon eskirgan sinfni o'zgartirish uchun merosdan foydalanadi va ob'yektlar vakillarini boshqa ob'jektga ishlab chiqaradigan shablonni ishlatadi.</p>			
Abstrakt fabrika	Abstract factory	Tizimlar komponentalarini yaratish uchun o'zida interfeysni taqdim etuvchi sinf.	Ha
Quruvchi	Builder	Murakkab ob'yektlarni yaratish uchun o'zida interfeysni taqdim etuvchi sinf.	Ha
Standart usul	Factory method	Ob'jektning yaratish uchun interfeysni belgilaydi, lekin qanday sinfning paydo bo'lishini belgilash yechimini qismlarida qoldiradi.	Ha
Ishga tushirishni keyinga qoldirish	Lazy initialization	Birinchi kirish paytida ishga tushirilgan ob'jekt.	Yo'q
Multiton	Multiton	Sinf nomi	Yo'q

		o'zgartirilgan ob'yekt nusxalariga ega va ularga global kirish nuqtasini taqdim etilishini ta'minlaydi.	
Ob'yektlar majmuasi	Object pool	Boshlang'ich va ishlatishga tayyor ob'yektlar majmuasi bilan ishlash uchun interfeysga ega bo'lgan sinf.	Yo'q
Prototip	Prototype	Ob'yektни konstruktordan yaratish o'rniga boshqa ob'yektни klonlash yo'li bilan yaratish uchun interfeysni belgilaydi.	Ha
Ba'zi resurslarni olish	Resource acquisition is initialization (RAII)	Ba'zi resurslarni olish ishga tushirish bilan, bo'shatish esa – ob'yektни yo'q qilish bilan moslashtirilgan.	Yo'q
Yolg'iz	Singleton	Faqatgina bitta nusxaga ega bo'lgan sinf.	Ha
Tuzilmaviy shablonlar (Structural) mavjud moslamalarni interfeysini o'zgartiradigan yoki uni amalga oshirishni o'zgartiradigan turli murakkab tuzilmalarni aniqlaydi, bu dasturni ishlab chiqishni va optimallashtirishni osonlashtiradi.			
Adapter	Adapter /	Ulardan biri	Ha

	Wrapper	foydalanadigan va boshqasi bilan mos kelmaydigan interfeysni ta'minlovchi ikkita boshqa moslamaning ta'sirini ta'minlovchi ob'ekt.	
Ko'prik	Bridge	Ishlab chiqarish interfeysi va sinfni amalga oshirish interfeysini mustaqil ravishda o'zgartirish imkonini beruvchi tizim.	Ha
Kompozit	Composite	O'ziga o'hshash ob'yektlarni o'zida birlashtiruvchi ob'yekt.	Ha
Dekorator yoki Wrapper	Decorator	Merosiylikdan foydalanmagan holda boshqa sinf funksional imkoniyatlarini kengaytiruvchi sinf.	Ha
Fasad	Facade	Ularni bir butun sifatida birlashtirib, bir nechta sinflar bilan ishlashni ifodalovchi ob'yekt.	Ha
Kirishning	Front controller	Qism tizim	Yo'q

yagona nuqtasi		interfeyslari uchun birlashtiruvchi interfeysni ta'minlaydi. Front Controller qism tizimdan foydalanishni osonlashtiradigan yuqori darajali interfeyslarni belgilaydi.	
Moslashirilgan	Flyweight	Bu dasturning turli qismlarida o'ziga xos nusxa sifatida o'zini namoyon qiladigan ob'yekt, lekin aslida unday emas.	Ha
O'rinbosar	Proxy	Ikki boshqa narsalar o'rtasida vositachilik qiladigan va u orqali kiritilgan ob'ektga kirishni cheklaydigan ob'ekt.	Ha
Xulq-atvoriga oid shablonlar (Behavioral) ob'yektlar orasidagi o'zaro bog'liqlikni belgilaydi va shu bilan moslashuvchanlikni oshiradi.			
Vazifalar zanjiri	Chain of responsibility	Tizimda mas'uliyat darajasini tashkil qilish uchun mo'ljallangan.	Ha
Action, Transaction buyrug'i	Command	Harakatni anglatadi. Buyruq ob'yekti ishni o'zi va uning	Ha

		parametrlarini qamrab oladi.	
Interpretator	Interpreter	Tez-tez uchraydigan, lekin o'zgarishi mumkin bo'lgan vazifani yechadi.	Ha
Cursor iterator	Iterator	Birlashtirilgan ob'yektlarning har birining tavsiflarini ishlatmasdan, jamlanma ob'jekt elementlariga ketma-ket kirishni ta'minlaydigan ob'jekt.	Ha
Vositachi	Mediator	Zaif aloqani shakllantirishda va ob'yektlar bir-biriga aniq murojaat qilish zarurligini bartaraf etishda ko'plab ob'yektlarning o'zaro ta'sirini ta'minlaydi.	Ha
Saqlovchi	Memento	Bu sizning ichki holatlaringizni ixchamlashtirishni buzmasdan tuzatishga va saqlashga imkon beradi, shunda keyinchalik bu holatlarda tiklanishi mumkin.	Ha
Null Object	Null Object	“Standart”	Yo'q

		ob'yektni taqdim qilib, nol ko'rsatkichlarni oldini oladi.	
Kuzatuvchi yoki yozilgan-Abonent	Observer	Ob'yektlar orasidagi munosabatlardan "bittadan ko'pga" munosabatlarni belgilaydi, ya'ni bitta ob'yektning holati o'zgarganda, unga bog'liq barcha narsalar bu voqea haqida xabardor qilinadi.	Ha
Hizmatchi[en]	Servant	Sinf guruhlarida umumiy funkcionallikni ta'minlash uchun foydalaniladi.	Yo'q
Mutahassislik	Specification	Biznes-mantiqni bog'lash uchun hizmat qiladi.	Yo'q
Holat	State	Dasturni bajarish vaqtida ob'yekt o'z holatiga qarab o'z xatti-harakatlarini o'zgartirishi kerak bo'lgan hollarda qo'llaniladi.	Ha
Strategiya	Strategy	Bu algoritmlar oilasini aniqlash, ulardan har birining ixchamlanishi va o'zaro	Ha

		almashinuvini ta'minlash uchun mo'ljallangan.	
Shablonli usul	Template method	Algoritmning asosini belgilaydi va merosxo'rlar butun tuzilishini o'zgartirmasdan algoritmning ba'zi qadamlarini qayta belgilashga imkon beradi.	Ha
Tashrif buyuruvchi	Visitor	Boshqa sinflar ob'ektlarida bajariladigan operatsiyani ta'riflaydi. Tashrif buyuruvchi sinfi o'zgartirilganda, xizmat qilayotgan sinflarni o'zgartirish shart emas.	Ha
Oddiy siyosat	Simple Policy		Yo'q
Event listener			Yo'q
Bir martalik tashrif buyuruvchi [en]	Single-serving visitor	Ishlatoladigan, keyinchalik foydalaniladigan va undan so'ng o'chiriladigan tashrif buyuruvchi shablonlarini ishlab chiqishni optimallashtirish.	Yo'q
Iyerarhik tashrif buyuruvchi[en]	Hierarchical visitor	Iyerarhik ma'lumotlar strukturasi	Yo'q

		barcha vertikallarini kesib o'tishning yo'lini beradi (masalan, daraxtsimon).	
--	--	---	--

Parallel dasturlash shablonlari. Ko'p ishlaydigan dasturlarni samaraliroq yozish uchun foydalaniladi va sixronlash muammolarida tayyor yechimlarni taqdim etadi.

Nomi	Haqiqiy nomi	Izoh
Active Object	Active object	U chaqirilgan oqimdan usullarni bajarish oqimini ajratib olish uchun xizmat qiladi. Assixron chaqirish usuli va rejalashtiruvchi shablonlaridan foydalanadi.
Balking[en]	Balking	Ob'yektga to'g'ri bo'lgan holatda, ob'yektda biror harakat bajarishga xizmat qiladi.
	Binding properties	Turli xil ob'yektlardagi xususiyatlarni sinxronlashtirishni ta'minlash uchun bir nechta kuzatuvchilarni birlashtiradi.
Xabarlar almashish[en]	Messaging pattern, Messaging design pattern (MDP)	Komponent va ilovalarda ma'lumot almashishga imkon beradi (xabar).
Ikki marta tekshiruvdan so'ng bloklash	Double checked locking	Blokni olish bilan bog'liq xarajatlarni kamaytirish uchun mo'ljallangan.
Event-based asynchronous[en]	Event-Based Asynchronous	Bir nechta oqimlardagi dasturlarda paydo bo'ladigan asinxron naqsh bilan manzil muammosi.
To'xtatib turishni qo'riqlash	Guarded suspension	Ob'yektga tegishli bo'lgan harakatni faqat to'g'ri holatda bajarilishiga bloklash uchun

		ishlatiladi.
Half-Sync/Half-Async		
Leaders/followers		
Bloklash	Lock	Bir oqim kirishni to'sish yoki boshqa oqimlar bilan almashtirish uchun manbani bloklaydi.
Monitor	Monitor	Bir oqimga nisbatan xavfsiz foydalanish uchun mo'ljallangan ob'yekt.
Reactor[en]	Reactor	Bir yoki bir nechta manbaalardan servisga so'rov jo'natish uchun mo'ljallangan.
Read/write lock[en]	Read/write lock	Bir nechta oqimni bir vaqtning o'zida umumiy bazadagi ma'lumotlarni o'qishga ruxsat beradi, faqat ma'lum bir vaqtning o'zida bir nechta o'zgarishga ruxsat beradi.
Rejalashtiruvchi	Scheduler	Har qanday aniq siyosatga bog'liq bo'lmagan holda rejalashtirish siyosatini amalga oshirish mexanizmini ta'minlaydi.
Thread-Specific Storage[en]		Turli oqimlar uchun turli global o'zgaruvchilarni taqdim etish uchun xizmat qiladi.
Bir oqimli bajarish	Single thread execution	Bu usulni raqobatbardosh deb atashga to'sqinlik qiladi, shu bilan birga ushbu usulni parallel bajarilishini taqiqlaydi.
Kooperativ namunasi	Cooperative pattern	Ish oqimlarini tugatish signalini berish uchun umumiy bayroqni ishlatib, ijro etuvchi oqimlarni xavfsiz ravishda to'xtatish uchun

		mexanizmni ta'minlaydi.
--	--	-------------------------

NAZORAT SAVOLLARI

1. Web loyihalashning asosiy tushunchalari?
2. Web loyihalash bosqichlari va ularning vazifalari?
3. Loyihalash shablonlari turlari?
4. Loyihalash shablon asosiy elementlari va kategoriyalari?

3-BOB. HTML TILIGA KIRISH. HTML HUJJATNING STRUKTURASI. SARLAVHA ELEMENTLARI

3.1. HTML gipermatnli hujjatni belgilash tiliga kirish

HTML (Hyper Text Markup Language) – belgili til bo'lib, ya'ni bu tilda yozilgan kod o'z ichiga maxsus ramzlarni mujassamlashtiradi. Bunday ramzlar hujjat ko'rinishini faqatgina boshqarib, o'zi esa ko'rinmaydi. HTMLda bu ramzlarni teg (teg – yorlik, belgi) deb ataladi. HTMLda hamma teglar ramz-chegaralovchilar (< , >) bilan belgilanadi. Ular orasiga teg identifikatori (nomi, masalan V) yoki uning atributlari yoziladi. Yagona istisno bu murakkab chegaralovchilar (<!--va -->) yordamida belgilanuvchi sharxlovchi teglardir. Aksariyat teglar jufti bilan ishlatiladi. Ochuvchi tegning jufti yopuvchi teg. Ikkala juft teg faqatgina yopuvchi teg oldidan «slesh» (“/”) belgisi qo'yilishini hisobga olmaganda, deyarli bir xil yoziladi. Juft teglarning asosiy farqi shundaki, yopuvchi teg parametrlardan foydalanmaydi. Juft teg yana konteyner deb ham ataladi. Juft teglar orasiga kiruvchi barcha elementlar teg konteyneri tarkibi deyiladi. Yopuvchi tegda zarur bo'lmagan bir qator teglar mavjud. Ba'zida yopuvchi teglar tushirib qoldirilsa ham zamonaviy brauzerlar aksariyat hollarda hujjatni to'g'ri formatlaydi, biroq buni amalda qo'llash tavsiya etilmaydi. Masalan, rasm qo'yish tegi , keyingi qatorga o'tish
, baza shriftini ko'rsatish <BASEFONT> va boshqalar o'zining , </BR> va hokazo yopuvchi juftlarisiz yozilishi mumkin. Noto'g'ri yozilgan tegni yoki uning parametri brauzer tomonidan rad qilinadi (bu brauzer tanimaydigan teglarga ham taalluqli). Masalan, <NOFRAME> teg-konteyneri faqatgina freymlarni taniydigan brauzer tomonidan hisobga olinadi. Uni tanimaydigan brauzer <NOFRAME> tegini tushunmaydi. Teglar parametr va atributlarga ega bo'lishi mumkin. Parametrlar yig'indisi har-bir tegda individualdir. Parametrlar quyidagi qoida asosida yoziladi:

- teg nomidan so'ng probellar bilan ajratilgan parametrlar kelishi mumkin;
- parametrlar ixtiyoriy tartibda keladi;
- parametrlar o'zining nomidan keyin keluvchi «=» belgisi orqali beriluvchi qiymatlarga ega bo'lishi mumkin;
- odatda parametrlar qiymati « » - «qo'shtirnoq» ichida beriladi;
- parametr qiymatida ba'zan yozuv registri muhim.

Shuni esda tutish lozimki, hamma teglar o'zining individual

parametriga ega bo'lishiga qaramay, shunday bir qator parametrlar mavjudki, ularni <BODY> bo'limining barcha teglarida ishlatish mumkin. Bu parametrlar CLASS, ID, LANG, LANGUAGE, STYLE va TITLElardir. HTML-hujjatini yozishni boshlashda ishlatiladigan birinchi teg bu <HTML> tegidir. U har doim hujjat yozuvining boshida bo'lishi lozim. Yakunlovchi teg esa </HTML> shakliga ega bo'lishi kerak. Bu teglar, ular orasida joylashgan yozuvning hammasi butun bir HTML-hujjatini anglatishini bildiradi. Aslida esa hujjat oddiy matnli ASCII-faylidir. Bu teglarsiz brauzer hujjati formatini aniqlab, tarjima qila olmaydi. Ko'pincha bu teg parametrga ega emas. HTML 4.0 versiyasiga qadar VERSION parametri mavjud edi. HTML4.0 da esa VERSION o'rniga <!DOCTYPE> parametri paydo bo'ldi.

<HTML> va </HTML> orasida 2 bo'limdan tashkil topishi mumkin bo'lgan hujjatning o'zi joylashadi. Mazkur hujjatning birinchi bo'limi sarlavhalar bo'limi (<HEAD> va </HEAD>), ikkinchi bo'lim esa hujjat tana qismidir (<BODY> va </BODY>), uni hujjat tanasi ham deb yuritamiz.

3.2. Gipermatnli hujjat tuzilishi. Hujjat sarlavhasi (HEAD bo'limi) konteyner elementlari va atributlari

HEAD bo'limi sarlavha hisoblanadi va u majburiy teg emas, biroq mukammal tuzilgan sarlavha juda ham foydali bo'lishi mumkin. Sarlavha qismining maqsadi hujjatni tarjima qilayotgan dastur uchun mos axborotni yetkazib berishdan iborat. Hujjat nomini ko'rsatuvchi <TITLE> tegidan tashqari bu bo'limning qolgan barcha teglari ekranda aks ettirilmaydi. Odatda <HEAD> tegi darhol <HTML> tegidan keyin keladi. <TITLE> tegi sarlavhaning tegidir va hujjatga nom berish uchun xizmat qiladi. Hujjat nomi <TITLE> va </TITLE> teglar orasidagi matn qatoridan iborat. Bu nom brauzer oynasining sarlavhasida paydo bo'ladi (bunda sarlavha nomi 60 belgidan ko'p bo'lmasligi lozim). O'zgartirilmagan holda bu matn hujjatga bookmark berilganda ishlatiladi. Hujjat nomi uning tarkibini qisqacha ta'riflashi lozim. Bunda umumiy ma'noga ega bo'lgan nomlar (masalan, Homepage, Index va boshqalar) ni ishlatmaslik lozim. Hujjat ochilayotganda birinchi bo'lib uning nomi aks ettirilishi, so'ngra esa hujjat asosiy tarkibi ko'p vaqt olib, kengayib ketishi mumkin bo'lgan formatlash bilan birga yuklanishini hisobga olgan holda, foydalanuvchi hech bo'lmaganda ushbu axborot qatorini o'qiy olishi uchun hujjatning nomi berilishi lozim.

<HEAD> sarlavha ichidagi ob'ektlar </HEAD>.

<head> konteyneri ichida quyidagi buyruqlardan foydalanish mumkin (teglardan):

title –xujjat nomi

meta –parametrlarni nom=qiymat

link –boshqa ob'ektlar bilan aloqa

style –stillarni aniqlash

script – skript, masalan JavaScript tilidagi

base – negiz URL ni beradi

Hujjat nomi brauzer darchasidagi sarlavhada sichqon yordamida BACK tugmasi bosib turilganda aks etadi. Hujjatga nom berish shart bo'lmasada, albatta yozish tavsiya etiladi. Chunki BACK tugmasi yordamida yoki jurnalni ko'rib chiqish jarayonida kerakli hujjatni ko'rish imkoni tug'iladi. Asosiy sabab esa, HTML-hujjatlar indeksatsiyasi sarlavha ustida bajarilishidir.

Meta tegi yordamida qidiruv mashinalari uchun tavsiflar, so'zlar va kodlarni berish mumkin.

Link toq tegi hujjatning saytdagi boshqa hujjatlar bilan aloqasini va uning sayt iyerarxik tuzilmasidagi o'rnini ko'rsatadi. Sarlavhada bir necha LINKning bir necha elementlari bo'lishi mumkin.

Uning atributlari:

href – URL ob'ektni aniqlaydi;

rel – joriy hujjatning HREF atributi orqali aniqlangan ob'ekt bilan o'zaro munosabati turini ko'rsatadi. Mumkin bo'lgan qiymatlari:

stylesheet – joriy hujjat uchun stillar jadvali (CSS) faylini ko'rsatadi. Brauzer href atributda ko'rsatilgan adresdan css-faylni yuklaydi va uni joriy hujjatga qo'llaydi (STYLE ga qarang);

home – saytingizning bosh sahifasini ko'rsatadi;

toc, contents – shu hujjatning (mundariyasi) sarlavhalari faylini ko'rsatadi;

index – joriy hujjat bo'yicha indeksga ko'ra izlash uchun zarur ma'lumotlar faylini ko'rsatadi;

glossary – joriy hujjatdagi atamalar ro'yxati faylini ko'rsatadi;

copyright – saytning yaratuvchilari, mualliflik xuquqlari va hokazolar qayd etilgan sahifasini ko'rsatadi;

up, parent – "yaratuvchi" sahifa (sayt iyerarxik tuzilmasida bir pog'ona yuqorida turgan hujjat) ni ko'rsatadi;

child – "yaralgan" sahifa (sayt iyerarxik tuzilmasida bir pog'ona pastda turgan hujjat) ni ko'rsatadi;

next – hujjatlar ketma-ketligidagi navbatdagi sahifani (masalan lugʻat, elektron katalog yoki tashkilot hujjatlarining keyingi sahifasini) koʻrsatadi;

previous – hujjatlar ketma-ketligidagi oldingi sahifani koʻrsatadi;

last, end – hujjatlar ketma-ketligidagi oxirgi sahifani koʻrsatadi;

first – hujjatlar ketma-ketligidagi birinchi sahifani koʻrsatadi;

help – maslahatli yordam sahifasini (masalan saytingizda harakatlanish boʻyicha) koʻrsatadi;

type – obʻekt uchun href atributida koʻrsatilgan MIME-turini aniqlaydi.

Masalan:

```
<HEAD>
```

```
<TITLE>Element DIV</TITLE>
```

```
<LINK rel="HOME" title="HTML-ma'lumotnoma" href="index.html">
```

```
<LINK rel="UP" title="Matn bo'laklari" href="Axborot_tex.html">
```

```
<LINK rel="PREVIOUS" title="Element P" href="magistr.html">
```

```
<LINK rel="NEXT" title="Element ADDRESS" href="test.html">
```

```
</HEAD>
```

Sarlavha elementlari teglari. HTML hujjatida sarlavhaning oʻz teglari mavjud boʻlib, ular oltitadir. Eng yuqori darajasi bu birinchidir. Har bir sarlavhaning oʻz tegi va oʻz tasvirlanish qoidasi mavjud. Eng katta yaʼni birinchi darajali sarlavha `<h1>` va `</h1>` teglari orqali, ikkinchi darajali sarlavha `<h2>` va `</h2>` teglari orqali va oxirgi oltinchi darajali sarlavha `<h6>` va `</h6>` teglari orqali ifodalanadi. Quyidagi misolda biz sarlavhalar tasvirini koʻramiz:

Misol:

```
<html>
```

```
<head>
```

```
<title>Sarlavhalarni tasvirlash</title>
```

```
</head>
```

```
<body>
```

```
<h1>Birinchi web sarlavha</h1>
```

```
<h2>Ikkinchi web sarlavha</h2>
```

```
<h3>Uchinchi web sarlavha</h3>
```

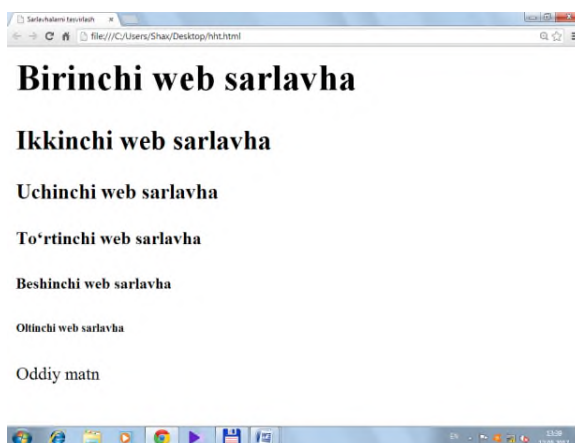
```
<h4>Toʻrtinchi web sarlavha</h4>
```

```
<h5>Beshinchi web sarlavha</h5>
```

```
<h6>Oltinchi web sarlavha</h6>
```

```
<p>Oddiy matn</p>
```

</body>
</html>



3.1-rasm. Sarlavha teglari bilan ishlash

Sarlavha teglari `` va `<div>` teglari kabi *id*, *class*, *style*, *align* parametrlarini o'z ichiga oladi.

3.3. Meta ma'lumotlar

`<META>` tegi nomalarning maxsus guruhi bo'lib, web sahifalarda meta ma'lumotlarini hosil qilish uchun `<meta>` tegi ishlatiladi, uning umumiy ko'rinishi quyidagicha:

```
<meta name=|o'zgaruchi nomi| - content=|o'zgaruvchi qiymati|>
```

Agarda biz Web sahifadagi muallif haqida ma'lumot yozmoqchi bo'lsak uni quyidagi ko'rinishda yozish mumkin:

```
<meta name =|Auther|content=|Bu meniki!!!|>
```

Meta ma'lumotlar asosan Internetda joylashgan qidirish mashinalari uchun zarur. Qidirish mashinalari Web sahifalar haqidagi ma'lumotlarni qaerdan oladi degan savol paydo bo'ladi. Xuddi shu ma'lumotlarini qidirish mashinalari meta o'zgaruvchilardan oladi.

`<meta>` tegida *keymards* va *description* o'zgaruvchilari mavjud.

Keymards o'zgaruvchisi web sahifadagi kalitli so'zlar ro'yxatini o'zida saqlaydi.

description o'zgaruvchisi esa Web sahifaning qisqacha ma'lumotini o'zida saqlaydi.

Masalan, bizning Web sahifamiz kompyuter viruslari haqida tayyorlangan bo'lsin u holda HTML hujjatda meta ma'lumotlarni quyidagicha yozish mumkin:

Misol.

```

<html>
<head>
<title>Web sahifa </title>
<meta name=|keywords| lang=|ru| content=| Web,
sayt, sahifa,...|>
<meta name=|description| content=|Web sahifa web sayt, web server
va web sahifaga bag'ishlangan.|>
</head>
<body>
Web sayt bu ....
</body>
</html>

```

Brauzerlar foydalanuvchi tomonidan ochib ko'rilgan web sahifalarni kesh xotirada saqlab qoladi. Agar foydalanuvchi yana shu sahifalarga murojaat qilsa, web brauzer oldindan kesh xotirada mavjud bo'lgan (yana yangi sahifani internetdan olmasdan) nusxasini olib ko'rsatadi. Bu jarayon foydalanuvchining vaqtini va iqtisodini tejaydi. Endi Web sahifa qachon yangilanadi degan savol paydo bo'ladi. Bu savolga meta ma'lumotlardagi expres o'zgaruchilardan javob olish mumkin. Bu o'zgaruvchida web sahifaning yaroqlilik muddati ko'rsatiladi.

Agar kesh xotiradagi web sahifa yaroqlilik muddati o'tgan bo'lsa, brauzer tarmoqdan web sahifani qaytadan o'qib oladi.

Misol:

```

<meta http-equiv=|Expres| content=|Tue,uf Aug 2002 14:56:27
Gmt|>

```

Web sahifalarda ma'lumotlar tez o'zgarishi mumkin, masalan chatlarda va birja sahifalarida ma'lumotlar tez o'zgaradi. Bunday hollarda refresh o'zgaruvchisidan foydalaniladi va qiymatlari sekundlarda beriladi.

Masalan:

```

<meta http-equiv=|Refresh| content=10>

```

Bu yozuvdan keyin web sahifa har 10 sekunddan keyin avtomatik tarzda o'zi qayta yuklanadi.

Hujjatning BODY bo'limi. Ushbu bo'linma hujjatning tarkibiy qismini o'z ichiga oladi. Bo'linma <BODY> tegidan boshlanib </BODY> tegida tugaydi. Biroq ushbu teglar qat'iy mavjud bo'lishi shart emas, chunki brauzerlar matnga qarab hujjat tarkibiy qismining ibtidosini aniqlashi mumkin. <BODY> tegining bir qator parametrlari mavjud bo'lib, ularning birortasi ham majburiy emas.

<**BODY**> tegi parametrlari:

ALINK – faol murojaat (Xavola (ssilka))ning rangini belgilaydi.

BACKGROUND – fondagi tasvir sifatida foydalaniluvchi tasvirning URL-manzilini belgilaydi.

BOTTOMMARGIN – hujjatning quyi chegaralarini piksellarda belgilaydi.

BGCOLOR – hujjat fonining ranglarini belgilaydi.

BGPROPERTIES – agar FIXED qiymati oʻrnatilmagan boʻlsa, fon tasviri aylantirilmaydi.

LEFTMARGIN – chap chegaralarni piksellarda belgilaydi.

LINK – xali koʻrib chiqilmagan Xavola (ssilka)ning rangini belgilaydi.

RIGHTMARGIN – hujjat oʻng chegarasini piksellarda oʻrnatadi.

SCROLL – brauzer darchalari harakatlantirish yoʻlaklarini oʻrnatadi.

TEXT – matn rangini aniqlaydi.

TOPMARGIN – yuqori chegarasini piksellarda oʻrnatadi.

VLINK – ishlatilgan murojaat rangini belgilaydi.

BOTTOMMARGIN, LEFTMARGIN, RIGHTMARGIN va TOPMARGIN pametrlari matn chegarasi va darcha chetlari orasidagi masofani piksellarda belgilaydi. (Faqat HTML 4.0 versiyasidan boshlab IE brauzerlari bu parametrlarni taniy oladi) BGPROPERTIES parametri faqatgina bitta FIXED qiymatiga ega. HTML dagi ranglar oʻn oltilik sanoq tizimida (RGB), yoki ranglar nomi yordamida berilishi mumkin. Ranglar bazasi 3 ta rangga – qizil (R), yashil (G) va koʻk (B) ranglarga asoslangan boʻlib, u RGB deb belgilanadi. Har-bir rang uchun 00 dan FF gacha boʻlgan oʻn oltilik sanoq tizimidagi qiymat beriladi, bu esa 0 dan 255 gacha boʻlgan diapazonga toʻgʻri keladi. Soʻngra bu qiymatlar bir songa birlashtiriladi va ularning oldiga “#” belgisi qoʻyiladi. Masalan, #800080 siyohrangni bildiradi.

Misollar:

<BODY TEXT = “#000000”> yoki <BODY TEXT = black>

<BODY BGCOLOR = “#C0C0C0”>yoki<BODY BGCOLOR = WHITE>

<BODY LINK = “#333”>yoki<BODY LINK = RED>

<BODY LINK = “#efefef” ALINK = “#800080”>yoki<BODY VLINK = Aqua ALINK = PURPLE>

Hamma brauzerlar oʻn oltilik sanoq tizimidagi standart ranglarni taniydi. Bular quyidagilardir:

3.1-jadval. Standart ranglar

Black	#000000	Maroon	#800000
-------	---------	--------	---------

Silver	#C0C0C0	Red	#FF0000
Grey	#808080	Purple	#800080
White	#FFFFFF	Green	#008000
Fuchsia	#FF00FF	Navy	#000080
Lime	#00FF00	Blue	#0000FF
Olive	#808000	Teal	#008080
Yellow	#FFFF00	Aqua	#00FFFF

Misol:

<BODY BGCOLOR = AQUA TEXT = “#848484” LINK = RED VLINK = PURPLE ALINK = GREEN>

Agar BGCOLOR parametri rangni nomi yoki uning tarkibiy qismlarini o'n oltilik sanoq tizimidagi kodda keltirish vazifasi yordamida fon rangini chiqarish uchun ishlatilsa, BACKGROUND tasvir yordamida sahifaga fon berishda foydalaniladi. Tasvir sifatida GIF yoki JPG formatidagi grafik fayllar ishlatiladi. HTML-hujjat fonidagi tasvir doimo butun sahifani to'ldirib turadi. Agar tasvir o'lchami darcha o'lchamidan kichik bo'lsa, u mozayka tamoyiliga asosan ko'paytiriladi. Odatda fon tasviri sifatida tarmoq orqali yusinh uchun uncha ko'p vaqt ketmaydigan kichik tasvir tanlab olinadi, yoki fon sifatida shaffof relef logotipi tasviridan foydalaniladi.

Misol:

<BODY BACKGROUND = texture.gif BGCOLOR = gray>.

Sahifa yaratilishida doimo fon rangini berish tavsiya qilinadi. Agar fon tasviri ham berilayotgan bo'lsa, fon va tasvir ranglari bir-biriga yaqin bo'lgani ma'qul.

Misol:

<BODY TEXT = BLUE LINK = RED VLINK = BLUE ALINK = PINK BACKGROUND= “ ” HYPERLINK <http://www.foo.com/jkorpela/HTML3.2/wave.gif> >.

Misol:

<HTML><HEAD><TITLE> - sahifa fonini berish misoli </TITLE></HEAD>

<BODY BGCOLOR = YELLOW TEXT = BLACK LINK = RED VLINK = PURPLE ALINK = GREEN>

</BODY>

</HTML>

3.4. Gipermatnli hujjat. Giper murojaat

Gipermatnli hujjat—bu boshqa hujjatlarga o'tish uchun aloqa bog'lovchi (Xavola (ssilka))ni o'zida saqlaydigan hujjat. U sichqoncha tugmasini bosish orqali bir hujjatdan ikkinchi hujjatga tezlik bilan o'tishni ta'minlaydi. Bunday aloqalarni zamonaviy mahsulotlar dasturlari fayllarida ko'plab uchratish mumkin.

Web—sahifalarga joylashtirish mumkin bo'lgan ko'plab multimedia ob'yektlari mavjud. Zamonaviy gipermatn hujjatlarida matnning o'ziga qo'shimcha qilib ko'pincha har xil grafika, video va audio ob'yektlardan foydalaniladi.

Murojaat (Xavola (ssilka)) sifatida esa, tasvir ishlatiladi.

Giper murojaat (Xavola (ssilka)) — bu maxsus HTML tegi bo'lib, unda parametr sifatida zaruriy hujjat manzili saqlanadi. Giper murojaat faollashganda (zamonaviy grafik brauzerlarda bu ish sichqonchanning chap tugmasi orqali amalga oshiriladi) zaruriy Web—serverga HTTP—talab (kerakli hujjat nomi bilan) yuboriladi.

Giper murojaat (giper xavola (ssilka)) ikki qismdan iborat. Uning birinchi qismini xavola (ssilka) ko'rsatkichi (anchor), brauzerga instruksiya beradigan ikkinchi qismi esa, xavola (ssilka)ning manzil qismi (URL—manzil) deyiladi. Agar foydalanuvchi sichqoncha kursoriga Xavola (ssilka) ko'rsatkichiga olib kelsa, u holda u ko'rsatuvchi belgiga aylanadi, (giper aloqaning birinchi belgisi).

Foydalanuvchi xavola (ssilka) ko'rsatkichi bo'yicha sichqonchanning chap tugmasini bosgan holda xavola (ssilka) qilinayotgan hujjatni oladi.

Xavola (ssilka) ko'rsatkichi so'z, so'zlar guruhi yoki tasvir bo'lishi mumkin. Xavola (ssilka) tashqi ko'rinishi uning turiga, yaratish usuliga va brauzer ko'rib chiqish dasturining qo'yilishiga bog'liq bo'ladi. Ko'rsatkichlarning matnli va grafikli turlari mavjud.

Matn ko'rsatkichlari bir yoki bir necha so'zlardan iborat bo'lib, so'z ostiga chizish bilan ajralib turadi. Xavola (ssilka) matn ko'rsatkichining yozilishi quyidagi misolda ko'rsatilgan:

< A HREF= «sample. html»> bu matn xavola (ssilka) ko'rsatkichi hisoblanadi.

Xavola (ssilka) sifatida grafik tasvirlardan ham foydalanish mumkin. Grafik xavola (ssilka)lar harakat qilish prinsipi bo'yicha matn xavola (ssilka)laridan hech qanday farqlanmaydi. Ular tagiga chizilmaydi va ranglari ajralib turmaydi.

Brauzerning ajralib ko'rinishi uchun, odatda tasvir atrofiga ramka rasmi chiziladi. Xavola (ssilka) grafik ko'rsatkichi uchun misol:

```
< AHREF= «sample. html»> <IMGSRC = «picture. gif»> </A>
```

Gipermurojaat ikkinchi qismi – bu URL–manzil. Web–sahifaning bu manzili ko'rsatkichda sichqoncha tugmachasi bosilganda yuklanadi. Manzil ko'rsatish nisbiy va absolyut bo'lishi mumkin.

Agar URL–manzilda faylga berish yo'li to'la ko'rsatilmasa, bunday xavola (ssilka) nisbiy deyiladi. Masalan, agar [http: //www.google.uz/page](http://www.google.uz/page) manzil bo'yicha brauzer sahifani yuklagan bo'lsa, u holda nisbiy ko'rsatkich [http: //www.google.uz/page/ picture](http://www.google.uz/page/picture) manzili deb tushunadi, ya'ni shu mashinada joylashgan katalog (qism katalog).

Nisbiy ko'rsatkichlar ishlatishda juda qulaydir. Kompyuter, katalog va faylni to'la–to'kis aniqlovchi URL–manzil absolyut ko'rsatkich deyiladi. Absolyut ko'rsatkichlar nisbiylardan farqli ravishda boshqa kompyuterlardagi fayllarga xavola (ssilka) qila oladi.

Xavola (ssilka)ni amalga oshirish uchun brauzerga uning xavola (ssilka) ko'rsatkichi hisoblanganligi va siz xavola (ssilka) qilayotgan hujjat manzilini ko'rsatish zarurligi to'g'risida xabar berish kerak bo'ladi. Ikkala harakat ham <A> tegi yordamida bajariladi. Teg <A> mohiyati (ma'nosi) URL–manzil hisoblangan yagona HREF parametriga ega. Ko'rsatkich nisbiy yoki absolyut bo'lishi mumkin. Bu teg konteyner hisoblanganligi bois, yopuvchi teg ni, albatta o'rnatish kerak bo'ladi.

Masalan, < AHREF= “[http: //www. Ictcouncil. google.uz](http://www. Ictcouncil. google.uz) ”> xavola (ssilka) matn ko'rsatkichi .

Nisbiy ko'rsatkichlar bilan ishlashni osonlashtirish maqsadida teg <BASE> kiritilgan. U hujjat boshidagi HEAD bo'limida joylashgan va URL–manziliga egadir, (unga nisbatan hujjatda hamma manzillar mavjud). Bu ko'rsatma nisbiy manzillar ishlatiladigan hujjatdagi istagan tegga ta'sir etadi. Agar teg <BASE> bo'lmasa, u holda manzil joriy hujjatlar manziliga nisbatan quriladi.

Endi–nisbiy manzilga mo'ljallangan an'anaviy misol. Quyidagi holatni ko'rib chiqamiz:

Biz, yaratmoqchi bo'lgan joriy fayldan ilgari yaratilgan amaldagi first.html fayliga xavola (ssilka) (Xavola (ssilka)) qilamiz, ya'ni;

```
<HTML>  
<HEAD> ‘  
<TITLE> Giperaloqa </TITLE>  
</HEAD>  
<BODY>
```


</H><CENTER><FONTCOLOR=#FF000> saytimizning hurmatli tashrif buyuruvchilari! </CENTER></H1>

<H4> Sizga xizmat qilayotgan bizning kompaniyamiz servislarini iloji boricha yaxshilash to'g'risidagi sizning istaklaringizni ko'proq bilishni hohlaymiz. Buning uchun, marhamat, <AHREF= "first.html" > anketamizni to'ldiring minnatdorchiligimizning cheki bo'lmaydi </H4>

</BODY>

</HTML>

So'ngra, faylni saqlang (second.html kabi) va brauzerda oching.

Ba'zida boshqa hujjatga emas, balki joriy hujjatning boshqa qismiga ko'chib o'tish kerak bo'lib qoladi. Masalan, katta hujjat yaxshi o'qiladi, agar unda bo'lishlarga mos kelgan xavola (ssilka)li mundarija bo'lsa.

Ichki xavola (ssilka)ni qurish uchun, avvalo tayinlangan joyni aniqlovchi ko'rsatkichni yaratish kerak. Buning uchun maxsus giperbog'lanish («yakor»), joriy hujjatdagi qandaydir joyni ko'rsatib beruvchi («anchors») lar ishlatiladi.

-

Yakor odatiy giperbog'lanish singari juft teg <A> bilan belgilanadi.

Atribut NAME esa keyinchalik boshqa giperbog'lanishlar xavola (ssilka) etiladigan «yakor» nomini beradi.

<A NAME ="chapter_1" Bob 1

Yuqorida ko'rsatilgan «yakor»ga misol keltirish uchun quyidagi giperaloqani yozamiz:

<AHREF= «#chapter_1» > 1–Bobga o'tish

Demak, hujjat manzilining HREF parametrada ko'rsatkich o'rniga, xuddi yuqoridagi kabi, ichki xavola (ssilka) ekanligini anglatib turuvchi prefiks # bilan birga xavola (ssilka) nomi joylashtirildi.

Endi, agar foydalanuvchi sichqoncha tugmasini bossa, brauzer joriy hujjatda chapter–1 nomli «yakor»ni qidiradi. Agar topa olsa, u holda o'zi aniqlangan hujjatning o'sha qismiga o'tib oladi. Ma'lum bo'ldiki, agar har bir sahifaning boshiga mundarija yozilsa bu juda kerak bo'lishi mumkin. U holda mundarijaning har bir punkti kerakli «yakor»ga xavola (ssilka) qiluvchi giperbog'lanish ekanligini bildiradi.

<HTML>

<HEAD>

<TITLE> sahifa–misollar ro'yxati </TITLE>

</HEAD>

<BODY>

```

</H> maqola </H1>
<P> mundarija </P>
<P> </P>
<P> < AHREF ="# chapter-1"> Bob 1 </A></P>
<P> < AHREF ="# chapter-2"> Bob 2 </A></P>
<P> < AHREF ="# chapter-3"> Bob 3 </A></P>
<HR>
<H2>< A NAME = "chapter-1"> Bob 1 </A></P>
<H2>< A NAME = "chapter-2"> Bob 2 </A></P>
<H2>< A NAME = "chapter-3"> Bob 3 </A></P>
</BODY>
</HTML>

```

Yuqoridagi misolni mundarijali sahifa yaratishda shablon sifatida qo'llashingiz mumkin. Biz, hujjatning bo'sh satriga joylashtirish uchun xat boshi <P></P> ning bo'sh tegidan foydalandik.

«Yakordan» foydalanib, boshqa hujjatning kerakli qismiga o'tish mumkin.

Bunday holatda giperbog'lanish quyidagi ko'rinishga ega bo'ladi:

```

<AHREF = " Second. html # chapter-2"> ikkinchi bo'limning 2-
bobiga o'tish </A>

```

Biz, boshqa Web-sahifalarda giperbog'lanishni ko'rib chiqdik. Lekin, <A> tegining HREF atributida internetning boshqa servislariga xavola (ssilka) (Xavola (ssilka)) qiluvchi manzillar bo'lishi mumkin. Shular qatoriga FTP-server yoki elektron pochta manzilini kiritish mumkin.

Ma'lumki, Web-sahifa HTTP protokoli yordamida uzatiladi.

Elektron pochtdan foydalanishga misol keltiramiz:

```

<HTML>
<HEAD>
<TITLE> Mening shaxsiy sahifam </TITLE>
</HEAD>
<BODY>
<H1 ALIGN = CENTER> Salom, do'stlar! </H1>
<HR>
Sahifaning asosiy mazmuni.
<HR>
<ADDRESS>
Sahifaning oxirgi yangilanish vaqti 10 iyun 2017 y.
</ADDRESS>

```

 Taklif va mulohazalaringizni yuboring:
 kozimjonyoqubov@mail.ru
</BODY>
</HTML>

Salom, do'stlar!

Sahifaning asosiy mazmuni.

Sahifaning oxirgi yangilanish vaqti 10 iyun 2017 y.

Taklif va mulohazalaringizni yuboring: kozimjonyoqubov@mail.ru

3.2-rasm. Giper murojaatni amalga oshiruvchi sahifa

Ma'lum bo'ldiki, HTML–kodidagi URL–manzil o'rniga oldindan mailto so'zi bilan ogohlantirilgach, elektron pochta manzilini ko'rsatdik. Sichqoncha kursorini berilgan xavola (ssilka) ko'rsatkichiga keltirsak, bu manzil brauzer darchasining pastki qismida ko'rinadi.

Boshqa internet zaxiralariga xavola (ssilka) etish xuddi shunday bajariladi.

Siz xavola (ssilka) matni rangini World Wide Webning boshqa joyida ham o'zgartirishingiz mumkin. Ilgari, sahifa matnining standart rangini <BODY> tegi yordami bilan bajarilar edi. Masalan, sahifangizdagi standart qizil matnning rangi quyidagicha o'rnatiladi:

</BODY TEXT = RED>

Giperbog'lanish matn rangini xuddi shunday boshqarishi mumkin, 3.1 – jadvalda Web–sahifadagi gipermatn xavola (ssilka)lar rangini boshqaruvchi 3 ta yangi kalit so'zlar keltirilgan.

3.2 – jadval. Xavola (ssilka) matn rangini boshqaruvchi kalit so'zlar

Kalit so'zlar	Izoh
LINK	LINK kalit so'zi TEXT so'zi kabi ishlatiladi. LINK=BLUE o'rnatilgan bo'lsa, tashrif buyuruvchilar tomonidan foydalanilmagan barcha xavola (ssilka)lar ekranda ko'k rang bilan aks etadi. <BODY TEXT = RED LINK =BLUE>
VLINK	Balki, siz sezgandirsiz, ba'zi xavola (ssilka)lar matni rangi ba'zan qolganlarnikidan ozgina bo'lsada farq qiladi. Bu mazkur xavola (ssilka)ning siz tomoningizdan faollashtirilganligini anglatadi. Tashrif

	<p>buyuruvchi boxabar bo'lgan (visited Link) xavola (ssilka) rangini VLINK kalit so'zi bilan o'zgartirishi mumkin. Shu usul bilan xavola (ssilka) rangini ochiq kul rang qilish mumkin.</p> <p><BODY TEXT = LIME LINK = AQUA VLINK =SILVER></p>
ALINK	<p>Siz yana o'zgartira oladigan rang–bu sichqoncha chap (yoki o'ng) tugmachasi bosilsa, paydo bo'ladigan xavola (ssilka) matni rangidir. Aktiv (faol) xavola (ssilka) rangi (Active Link) ALINK kalit so'zi orqali beriladi. Agar sizning sahifangizda kimdir ishtirok etgan bo'lsa, u holda xavola (ssilka) matni LINK so'zi bilan beriladigan rangga aylanadi. Agar unda sichqoncha chap (yoki o'ng) tugmachasi bosilsa, u holda qisqa vaqt ichida ALINK so'zi bilan beriladigan rangga ega bo'ladi. Shu usulda ishimizni davom ettirib, faol xavola (ssilka) matni rangini och sariqqa aylantiramiz:</p> <p><BODY TEXT LIME LINK =AQUA VLINK =SILVER ALINK = YELLOW></p>

NAZORAT SAVOLLARI

1. HTML gipermatnli hujjat strukturasi?
2. HEAD tegi va uning ichidagi teglarning vazifalari?
3. BODY tegi va uning parametrlari?
4. Izohlardan foydalanishni izohlang.
5. Gipermatnli hujjat deganda nimani tushunasiz?
6. Giperaloqa deganda nimani tushunasiz?
7. Giperaloqa nechta qismdan iborat?
8. Xavola (ssilka) matni rangini boshqaruvchi kalit so'zlar.

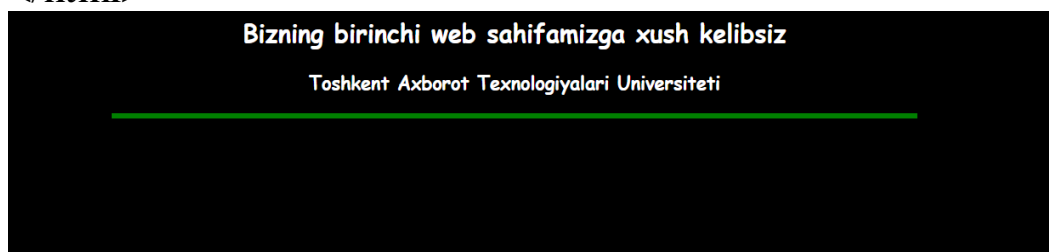
4-BOB. HTML ASOSIY ELEMENT TEGLARI. JADVAL YORDAMIDA SAHIFALASH

4.1. HTML gipermatn tilining asosiy teglari: center, hr, br, h1, p, font

Ba'zi bir teglarning atributlari, boshqa teglarda ham qaytarilishi mumkin, shunday bo'lsa ham iloji boricha har birini alohida ko'rib o'tamiz.

Ba'zi bir teglarning yopiluvchi teglari mavjud emas (</>), bu holda uni yopish shart emas, lekin yopiluvchilari bo'lgan teglarni albatta yoping, agar yopmasangiz bu teg boshqa kodlarga ta'sir qilib, sahifangizni o'zgartirib yuborishi mumkin.

```
<!--Asosiy oyna-->
<html>
<head>
<title>Asosiy oyna</title>
</head>
<body bgcolor=black text=white >
<marquee behavior="alternate" direction="left">
<h1 align="center">
Bizning birinchi web sahifamizga xush kelibsiz
</h1>
</marquee>
<b>
<h2><P align=center>Toshkent Axborot Texnologiyalari
Universiteti </P> </h2>
</b>
<hr align="center" color=green size=6 width=70%>
</body>
</html>
```



4.1-rasm. Sahifa va ekranda gorizontaal chiziq chiqarish

1. **<center>...</center>** - bu teg ichidagi barcha yozuvlar (teglar ham bo'lishi mumkin) sahifaning markazidan boshlab yoziladi, shunda umumiy yozuv o'rtada namoyon bo'ladi.

2. **<hr>** - bu tegning yopiluvchi sherigi yo'q. Teg oddiy chiziq chizib beradi. Uning quyidagi atributlari mavjud:

align - chiziqni tekisligini o'rnatadi va quyidagi qiymatlarni qabul qilishi mumkin:

*left - chap tomondan tekislash

*right - o'ng tomondan tekislash

*center - markazga tekislash.

color - chiziq rangini berish (rang nomi yoki 16 lik tizimdagi raqami).

noshade - effektlarsiz chizish.

size - chiziq qalinligi (size= "20").

width - chiziq uzunligi, piksel yoki foizda beriladi (width= "20%").

3. **
** - kursorni bitta pastga tushirish, undan keyin keladigan barcha narsalar (jadval, yozuv,...) bitta pastdan yoziladi (ya'ni Enter). Yopiluvchi sherigi yo'q.

4. **<h1>...</h1>** - Matnga nom qo'yish. Bu sinf 6 xil ko'rinishda yozilishi mumkin (h1, h2, h3, h4, h5, h6). Raqamlar kattalashgan sari yozuvlar o'lchami kichisinfhib boradi. Bu teg har xil o'lchamdagi nomlar (bo'lim nomi, band nomi, bob nomi,...) berishga qulay. Uning bitta atributi mavjud:

align - yozuvni tekislash, qiymatlari quyidagilar:

*left - chap tomonga to'g'irlash (odatiy holi).

*center - markazga to'g'irlash.

*right - o'ng tomonga to'g'irlash.

*justify - butun qator bo'ylab to'g'irlash.

5. **<p>...</p>** - paragraflar qo'shish uchun ishlatiladi. Har bir ochilgan «p» teg yozuvni, yangi qatordan boshlab ekranga chiqaradi. Bu tegning quyidagi atributi mavjud:

align - yozuvni tekislash, u quyidagi qiymatlarni qabul qiladi:

*left - chap tomonga tekislash;

*right - o'ng tomonga tekisash;

*center - markazga tekislash;

*justify - butun qator bo'ylab tekislash;

6. **...** - yozuvni formatlangan holda chiqarib beradi. Quyidagi atributlari mavjud:

color - yozuv rangini o'rnatish;

face - yozuv shriftini o'rnatish (times new roman, verdana, arial);
size - yozuv o'lchamini o'rnatish.

4.2. Ro'yxatlar va ularning turlari

Ma'lumki, ro'yxatlarni har bir Web-sahifada uchratish mumkin. HTML tilida ma'lumotlarni ro'yxat ko'rinishida ifodalash uchun maxsus teglar mo'ljallangan.

Ro'yxatlar ma'lumotlarni elektron va nashr hujjatlarida ifodalashning eng ko'p qo'llaniladigan formalardan biri hisoblanadi.

Ro'yxatlardan foydalanish quyidagi hollarda juda qulaydir:

- o'qish paytida qulaylik tug'dirish uchun axborot fragmentlarini yagona strukturaga birlashtirish;
- murakkab qidiruv jarayonlarini yozib chiqish;
- punktlari (joylari) mos hujjat bo'linishlarini ko'rsatib turuvchi axborot mundarijasining joylashuvi.

HTML tilida ro'yxatlarning quyidagi asosiy turlari mavjud markerlangan, raqamlangan va aniqlash ro'yxati. Yana boshqa xillar ham bor, lekin yuqoridagi uchta ro'yxat turlari eng ko'p ishlatiladi. Ularning o'zaro o'xshashligi shundaki, ularning barchasi xat boshining har bir punktiga joylashgan bo'ladi va siz bimalol, har bir matnni ajratishingiz mumkin. Yuqorida keltirilgan ro'yxat turlarining asosiy farqi raqamlashtirish usulida va uning strukturasiadir.

Markerlangan ro'yxat. Markerlangan (aks holda bu kabi ro'yxatlar raqamlangan yoki tartibga keltirilmagan ro'yxatlar deyiladi) ro'yxatni yaratish uchun ichida ro'yxatning hamma elementlari mavjud teg-konteyner `` `` (UL–Unordered List–tartibga solinmagan ro'yxat)dan foydalanish zarurdir. Ro'yxatning ochuvchi va yopuvchi teglari satrni ro'yxat boshiga va oxiriga o'tkazishni ta'minlab turadi.

Shu bilan birga, ro'yxatni hujjatdagi asosiy ma'lumotlardan ajratib turadi.

Shuning uchun, bu yerda `<p>` xat boshi tegini qo'llash yoki `
` satrini majburiy o'tkazish zaruriyati yo'q.

Ro'yxatning har bir elementi teg `` (/LI–List Item–ro'yxati elementi) bilan boshlanishi kerak. Teg `` mos yopuvchi tegga muhtoj emas, lekin uning mavjudligi xato hisoblanmaydi. Brauzerlar hujjatni aks ettirishda har bir yangi ro'yxat elementini xat boshidan boshlaydi.

Misol:

`<HTML>`

```
<HEAD>
<TITLE> Burj yulduzlari </TITLE>
</HEAD>
<BODY>
<UL>
<H1> B Burj yulduzlari </B></H1>
<LI> Hamal
<LI> Savr
<LI> Javzo
<LI> Saraton
<LI> Asad
<LI> Sumbula
<LI> Mezon
<LI> Akrab
<LI> Qavs
<LI> Jadiy
<LI> Dalv
<LI> Xut
</UL>
</BODY>
</HTML>
```

Web ilova:

B Burj yulduzlari

- Hamal
- Savr
- Javzo
- Saraton
- Asad
- Sumbula
- Mezon
- Akrab
- Qavs
- Jadiy
- Dalv
- Xut

4.2-rasm. Markerlangan ro'yxat ko'rinishi.

 tegi bilan belgilangan ro'yxat elementlaridan tashqari boshqa HTML–elementlari ham ishtirok etish mumkinligini aytib o'tish zarurdir. Yuqoridagi misolda xuddi shunday elementlardan biri ro'yxat punkti (joyi) hisoblanmaydigan, lekin uning sarlavhasida asosiy rolni o'ynaydigan oddiy matn ko'rsatilgan.

HTML tili bo'yicha ba'zi darsliklarda ro'yxat sarlavhasi uchun teg–konteyner <LH> (LH–List Header ro'yxat sarlavhasini) ishlab chiqish kerakligi to'g'risidagi ko'rsatmalar uchrab turadi. Hozirgi paytda nomi chiqqan brauzerlardan birortasi ham bu tegni tan olmaydi va u HTML spetsifikasiga kirmaydi. Shunday qilib bu teg bitta – yarimta xatoliklarni keltirib chiqarmasa ham, uni qo'llash hech qanaqa ma'no kasb etmaydi.

 tegida ikkita parametr ko'rsatilishi mumkin, COMPACT va TYPE.

COMPACT parametri ma'nosiz yoziladi va mazkur ro'yxatni kompakt ko'rinishida chiqarish zarurligini brauzerga biriktirib qo'yish uchun ishlatiladi.

Masalan, shriftni yoki ro'yxat satrlari orasidagi masofani kichraytirish mumkin va b.q.

TYPE parametri ro'yxat satri belgilanadigan nishonni berish (yusinh) imkonini yaratadi; uchtasidan bittasiga ruxsat berilgan: diss, circle va square. Bu parametr ro'yxat markerlar xilini majburiy topshirish uchun ishlatiladi. Aniq marker xili foydalaniladigan brauzerga bog'liq bo'ladi. Aks ettirishning tipik variantlari quyidagicha:

TYPE = diss–markerlar bo'yalgan aylanalar bilan aks ettiriladi;

TYPE = circle–markerlar bo'yalmagan aylanalar bilan aks ettiriladi.

TYPE = square–markerlar bo'yalgan kvadratchalar bilan aks ettiriladi.

Misol: <UL TYPE = circle >.

TYPE = diss standart bo'yicha foydalaniladigan qiymat (ma'no, mohiyat, qiymat) hisoblanadi. Kiritilgan markerlangan ro'yxatlar uchun standart bo'yicha berilgan darajada diss ko'rsatkichi ishlatilsa, ikkinchida circle, uchinchi va undan keyingi darajalarda square ishlatiladi.

Masalan, HTML 4.0 spetsifikatsiyasida TYPE = square qiymatida aks etadigan marker xili uchun bo'yalmagan to'rtburchak ko'rsatiladi, (square outline).

Parametr TYPE shu ko'rsatkichlari bilan alohida ro'yxat elementlari markerlari xillarining ko'rsatmasi uchun qo'llanishi mumkin. Buning

uchun parametr TYPE o'ziga mos kelgan qiymatlari bilan elementlari ro'yxati tegida ko'rsatma berishiga ruxsat etilgan.

Misol: <LI TYPE = circle >

Grafik ro'yxat markerlari. Ro'yxat markerlari sifatida grafik tasvirlardan foydalanish mumkin, chunki ular chiroyli, o'ziga jalb qiluvchi darajasida rasmiylashtirilgan HTML hujjatlarni yaratishda ko'plab ishlatiladi. Haqiqatan, bunday imkoniyat to'g'ridan-to'g'ri HTML tomonidan taqdim etilmaydi, lekin biroz suniy ravishda amalga oshiriladi. Maqsad shuki, ro'yxat tegi (quyida ko'riladigan barcha teglar singari) yagona topshiriqni bajaradi, ya'ni berilgan tegdan keyin joylashgan barcha axborotlar qandaydir kattalikka o'ng tomonga siljish bilan aks etishi zarurligi to'g'risida brauzerga ko'rsatma beradi. Alohida ro'yxat elementlariga ko'rsatma beruvchi teglari standart ro'yxat elementlari markerlarini olib ketishni ta'minlaydi.

Agar grafik markerlar bilan ro'yxat tuzish talab qilinsa, u holda, umuman teglarisiz ishni hal qilish mumkin. Faqatgina har bir ro'yxat elementi oldiga qo'yilgan grafik tasvirni joylashtirish kerak bo'ladi. Bunda hal qilinishi lozim bo'lgan yagona masala bu ro'yxat elementlarini bir-biridan ajratishdir. Buning uchun <p > xat boshi teglaridan yoki
 satrini majburiy o'tkazishdan foydalansa bo'ladi. Grafik markerli ro'yxatni amalga oshirish misoli quyida keltirildi:

```
<HTML>
<HEAD>
<TITLE> ro'yxat grafik markerlari </TITLE>
</HEAD>
<BODY>
<UL>
<H1> <B> materiklar ro'yxati </B></H1>
< IMG SRC = "Tips.gif" > Yevropa <BR>
< IMG SRC = "Tips.gif" > Osiyo <BR>
< IMG SRC = "Tips.gif" > Afrika <BR>
< IMG SRC = "Tips.gif" > Amerika <BR>
< IMG SRC = "Tips.gif" > Avstraliya <BR>
</UL>
</BODY>
</HTML>
```

Mazkur misolda ro'yxat elementi markeri sifatida Tips.gif grafik fayli ishlatildi. HTML-sahifalarda grafiklarni qo'llash uzatilgan axborotlar hajmi ancha oshib ketishiga olib kelishini aytib o'tish maqsadga

muvofiqdir. Lekin, bu holatda hajm kengayishini nazarga olmasa ham bo'ladi, chunki u juda kam o'zgargan.

Bu yerda barcha markerlar uchun faqat bir marta beriladigan fayl ishlatiladi.

Kichik tasvirli fayl o'lchami ham juda kichik bo'ladi.

Raqamlangan ro'yxat. Raqamlangan (OL–Ordered List–tartibga keltirilgan ro'yxat) markerlangan ro'yxatga juda o'xshab ketadi. Lekin, ularda birgina farq mavjud, u ham bo'lsa raqamlangan ro'yxatda har bir punkt oldiga grafik merkerlar o'rniga avtomatik ravishda raqamlar yoki harflarning ketma-ket joylashtirishidir.

Raqamlashtirish brauzerga bog'liq bo'ladi va ro'yxat teg parametrlari orqali beriladi.

Raqamlangan ro'yxatni yaratish uchun ichida barcha ro'yxat elementlari joylashgan teg konteyner dan foydalanish zarur hisoblanadi.

Ochuvchi va yopuvchi ro'yxat teglari satrni ro'yxatning old va orqa tomoniga o'tkazishni ta'minlaydi. Shunday usul bilan ro'yxat hujjat ichidagi asosiy narsalardan ajratiladi. Markerlangan ro'yxat singari har bir raqamlangan ro'yxat elementi ham teg bilan boshlanishi kerak. Misol.

```
<HTML>
<HEAD>
<TITLE> raqamlangan ro'yxat misoli </TITLE>
</HEAD>
<BODY>
<OL>
<H1> <B> Yerdan ko'rinadigan yorug'roq yulduzlar: </B></H1>
<LI> Sirnus
<LI> Kanopus
<LI> Arktur
<LI> Alfa Sentavra
<LI> Vega
<LI> Kapella
<LI> Rigel
<LI> Protsion
<LI> Axernar
<LI> Beta Sentavra
<LI> Betelgeyze
<LI> Aldebarak
```

```
<BR> ....  
<LI Valuye = 58> Mitsar  
<BR>  
<LI Valuye = 75 > Qutb  
</OL>  
</BODY>  
</HTML>
```

Web ilova:

Yerdan ko'rinadigan yorug'roq yulduzlar:

1. Sirnus
2. Kanopus
3. Arktur
4. Alfa Sentavra
5. Vega
6. Kapella
7. Rigel
8. Protsion
9. Axernar
10. Beta Sentavra
11. Betelgeyze
12. Aldebarak
-
13. Mitsar
14. Qutb

4.3-rasm. Raqamlangan ro'yhat ko'rinishi

 tegi quyidagi parametrlarga ega: COMPACT, TYPE va START.

Parametr COMPACT markerlangan ro'yxatlardagi kabi ma'noga ega. Parametr TYPE esa, ro'yxatni raqamlash xillarining topshirig'i (vazifasi) uchun ishlatiladi.

Quyidagi qiymatlarni qabul qilishi mumkin:

TYPE = 1. Standart raqamlash–1, 2, 3, 4, 5 ...

TYPE = A. Bosh harflar–A, B, C, D, E, ...

TYPE = a. Kichik harflar–a, b, c, d, e, ...

TYPE = I. Rim raqamlari–I, II, III, IV, V, ...

TYPE = i. Kichik rim raqamlari–I, II, III, IV, V, ...

Hamisha standart bo'yicha TYPE = 1 ishlatiladi, ya'ni arab raqamlari yordamida raqamlansa. Element ro'yxati raqamidan so'ng doim qo'shimcha «nuqta» belgisi chiqariladi.

Parametr TYPE o'sha qiymatlar alohida ro'yxat elementlarining

raqamlash xillariga ko'rsatma berish uchun qo'llanilishi mumkin. Buning uchun TYPE parametri mos qiymatlar bilan ro'yxat elementi tegida ko'rsatma berishi mumkin.

Misol. <LI TYPE = A>

 tegining START parametri ro'yxat raqamlashini birdan boshlamaslikka imkon beradi. START parametri qiymati sifatida doim natural son ko'rsatilishi kerak.

Misol: <OL TYPE = A START = 5>

Bunday yozuv ro'yxatni raqamlashda lotincha bosh harf «Ye»dan boshlashni aniqlaydi. Raqamlashning boshqa xillari uchun START5 = yozuvi mos ravishda raqamlashni «5» sonidan, rim raqami «V» (va sh.k.) dan boshlash kerakligini bildiradi.

Ro'yxat raqamlash turini va raqamlar qiymatini o'zgartirish istagan ro'yxat elementi uchun ham mumkin.

Teg raqamlangan ro'yxatlar uchun TYPE va VALUE parametrlaridan foydalanishga ruxsat beradi. TYPE parametri tegining qiymatlarini qabul qilib olishi mumkin.

Misol: <LI TYPE = A>

Teg VALUE parametrining qiymati berilgan ro'yxat elementi raqamini o'zgartirishga yordam beradi. Bunda barcha navbatdagi elementlarning raqamlanishi o'zgarib ketadi. Bunday ro'yxatga misol yuqorida keltirib o'tildi. Bu misolda eng yorug' yulduzlarning tartibli ro'yxati mavjud bo'lib, unda 58 va 75 ta joylarda joylashgan, bizning kengliklarda yaxshi ko'rinadigan yo'lduzlar bor. (Mitsar–katta ayiq yulduzlar turkumidagi eng yorug' yulduz; qutb yulduzi esa, kichik ayiq turkumiga tegishli).

Aniqlash ro'yxati. Aniqlash ro'yxatlari qandaydir terminlar ro'yxatini va ularning ta'rifini yaratish uchun juda mos tushadi. Ideal holatda aniqlash ro'yxatidan foydalanish–bu lug'at. HTML yordami bilan siz har bir termin va uning aniqlanishini osongina topasiz.

Aniqlash ro'yxatlari teg konteyner <DT> (Definition List) yordami bilan beriladi. Konteyner ichki qismida <DT> teg (Definition Term) bilan aniqlanadigan termin belgilansa, <DD> teg (Definition Description) bilan esa, xat boshi o'z aniqlanishi bilan belgilanadi. <DT> va <DD> teglari uchun mos yopuvchi teglarni yozmasa ham bo'ladi.

Umuman, aniqlash ro'yxatlari quyidagicha yoziladi:

<DL>

<DT> termin

<DD> terminini aniqlash

</DL>

<DL> tegida ishlashda boshqa ro'yxatlarga o'xshash COMPACT parametri ko'rsatilishi mumkin. Aniqlash ro'yxati qo'llangan HTML–hujjat misolini keltiramiz:

<HTML>

<HEAD>

<TITLE> aniqlash ro'yxati misoli </TITLE>

</HEAD>

<BODY>

<H1> aniqlash ro'yxati misoli </H1>

<DL>

<DT> Internet

<DD>. Butun dunyo kompyuter tarmog'i.

<DT> HTML

<DD> internet tarmog'ida nashr uchun mo'ljallangan hujjatlar yaratish tili.

<DT> www

<DD> Internet tarmog'iga joylashtirilgan o'zaro bog'langan elektron hujjatlar massivining umumiy nomi. Eng ommabop internet–servis.

</DL>

</BODY>

</HTML>

Taxlangan ro'yxatlar. Ro'yxatlarning ajoyib xususiyatlaridan biri–bu ularni bir-biriga joylashtirish (taxlash) imkoniyati borligidir. Bu holda ko'p darajali yoki taxlangan ro'yxatlar tashkil etiladi.

Ro'yxatga ro'yxat qo'shish ro'yxatni yaratish kabi amalga oshiriladi. Buning uchun HTML maxsus teglari bo'lmaydi. Brauzerni adashtirib yubormaslik uchun, siz har bir ichki ro'yxatni tegi yordami bilan yopganingizga ishonch hosil qiling. Hattoki har xil turdagi ro'yxatlarni ham bir-biriga joylash mumkin. Quyida taxlangan ro'yxatli HTML–hujjat kodi keltirilgan. Bu misolda har bir markerlangan ro'yxat elementiga o'zining raqamlangan ro'yxati taxlangan.

<HTML>

<HEAD>

<TITLE> taxlangan ro'yxat misoli </TITLE>

</HEAD>

<BODY>

<H1> ba'zi planetalar yo'ldoshlari </H1>

```
<LI> Yer
<OL>
<LI> Oy
</OL>
<LI> Mars
<OL>
<LI> Fobos
<LI> Deymas
</OL>
<LI> Uran
<OL>
<LI> Ariyel
<LI> Umbriyel
<LI> Titaniya
<LI> Oberon
<LI> Miranda
</OL>
<LI> Neptun
<OL>
<LI> Triton
<LI> Nerenda
</OL>
</UL>
<BODY>
<HTML>
```

4.3. HTML da jadvallarni hosil qilish. Sahifalarni jadvallar yordamida verstka qilish

Jadvallar HTMLda keng miqyosda qo'llaniladigan va eng kuchli vositalardan biri hisoblanadi.

Hozirda jadvallar ma'lumotlarni ifodalash bilan birga, Web-sahifani rasmiylashtirishdagi element sifatida ham, ishlatilmoqda. Jadvallar tufayli, matn fragmentlari va grafiklarning pozitsion o'rnini aniq almashtirishni, bezash elementlarini, chizg'ich va bo'yashlarni, chekinish va bo'rttirishlarni xuddi bosma nashrdagidek bajarish mumkin.

Avvalo, oddiy jadvallarni yaratish uchun zarur bo'lgan teglar va ularning parametrlarini ko'rib chiqamiz, so'ngra ularni batafsil ta'riflashga o'tamiz.

Jadval ta'rifi <BODY> hujjati bo'limining ichki qismida joylashgan bo'lishi kerak. Hujjat ixtiyoriy miqdordagi jadvallarni o'zida sig'dirish mumkin, shu bilan birga ularni bir-biriga taxlash uchun ham ruxsat beriladi. Har bir jadval, <TABLE> tegi bilan boshlanishi va </TABLE> tegi bilan tugallanishi shart. Bu juft teglar ichida jadvaldagi barcha ta'rif (opisaniye)lar joylashtirilgan. Har bir jadval yoki bir necha satrlardan iborat va ularning har birida alohida yacheykalar uchun ma'lumotlar berilgan.

Har bir satr <TR> tegi (Table Row) bilan boshlanadi va </TR> tegi bilan tugallanadi. Satrdagi alohida yacheykalar <TD> va </TD> (Table Data) yoki <TH> va </TH> (Table Header) teglar bilan o'raladi. <TH> tegi odatda yacheyka-sarlavhalari uchun, <TD> tegi esa, yacheyka-berilgan ma'lumotlar uchun qo'llaniladi. Foydalanishdagi ular orasidagi farq shrift turiga bog'liq bo'ladi, chunki ular standart bo'yicha yacheyka ichidagilarni aks ettirish va berilgan ma'lumotlar qanday joylashganligini bilish uchun ishlatiladi. <TH> turli yacheykada ma'lumotlarni yarimquyuq (Bold) shrift bilan ko'rsatiladi va markaz bo'yicha joylashadi, (ALIGN=CENTER, VALIGN=MIDDLE).

<TD> tegi bilan aniqlangan yacheykalar chapga (ALIGN=LEFT) va o'rtaga (VALIGN=MIDDLE) vertikal yo'nalishga to'g'rilangan ma'lumotlarni standart bo'yicha aks ettiradi.

<TD> va <TH> teglari <TR> jadval satri ta'rifidan tashqarida paydo bo'la olmaydi.

Yakunlovchi </TR>, </TD> va </TH> kodlari tushirib qoldirilishi mumkin.

Bunday holatda satr ta'rifi yoki yacheykaning oxiri navbatdagi satr yoki yacheyka, yoki jadval oxirining boshlanishi hisoblanadi. Yakunlovchi jadval tegi </TABLE> esa, tushirib qoldirilishi mumkin emas.

Jadvaldagi satrlar miqdori ochuvchi teg <TR>lar soni bilan, ustunlar miqdori esa, <TD> yoki <TH>larning maksimal miqdori bilan aniqlanadi.

Yacheykalar bir qismi hech qanday ma'lumotlarga ega bo'lmasliklari mumkin. Bu kabi yacheykalar ketma-ket keladigan juft teglar-<TD>, </TD> bilan ta'riflanadi. Agar qandaydir satr oxiriga joylashgan bitta yoki bir necha yacheykalar ma'lumotlarga ega bo'lmasalar, u holda ularning ta'rifini tushirib qoldirish mumkin bo'ladi. Brauzer esa, avtomatik ravishda talab darajada bo'sh yacheykalarni qo'shib qo'yadi. Demak, har xil satrlarda joylashgan bir xil o'lchamli turli xil miqdordagi ustunlarni o'z ichiga olgan jadvallarni qurish man etiladi.

Jadvalda juft teglar-<CAPTION> va </CAPTION>ga joylashtirilgan

sarlavha bo'lishi mumkin. Jadval sarlavhasining ta'rifi <TABLE> va </TABLE> teglari ichki qismining istagan joyida bo'lishi mumkin. Lekin, istagan <TD> va <TH> yoki <TR> teglari ta'rifidan tashqarida joylashishi kerak.

HTML tili spetsifikatsiyasiga asosan sarlavha ta'rifining joylashuvi qat'iy tarzda reklama qilingan: u <TABLE> tegidan keyin va birinchi teg <TR>gacha shu zahotiyoq joylashishi zarur.

Standart bo'yicha jadval sarlavha matni uning ustiga (ALIGNqTOP) joylashadi va gorizontol holatda markazlashtiriladi.

Sanab o'tilgan teglar son va moxiyati jihatdan turli xil parametrlarga ega bo'lishi mumkin. Lekin eng oddiy holatda ular standart bo'yicha qiymat qabul qiladigan parametrlarsiz ishlatiladi.

Endi misol keltiramiz. Birinchi jadvalimiz juda oddiy va ko'paytirish jadvali bo'ladi. Bu sahifani table1. html, deb ataymiz.

```

<HTML>
<HEAD>
<TITLE> ko'paytirish jadvali </TITLE>
</HEAD>
<BODY>
<H1> Ko'paytirish jadvali (1 dan 5 gacha) </H1>
<TABLE BORDER="2">
<TR><TD>№</TD><TD>2x</TD><TD>3x</TD><TD>4x</TD><
TD>
5x</TD></TR>
<TR><TD>1</TD><TD>2</TD><TD>3</TD><TD>4</TD><TD>
5
</TD></TR>
<TR><TD>2</TD><TD>4</TD><TD>6</TD><TD>8</TD><TD>
10
</TD></TR>
<TR><TD>3</TD><TD>6</TD><TD>9</TD><TD>12</TD><TD>
>15
</TD></TR>
<TR><TD>4</TD><TD>8</TD><TD>12</TD><TD>16</TD><T
D>20
</TD></TR>
<TR><TD>5</TD><TD>10</TD><TD>15</TD><TD>20</TD><T
D>25
</TD></TR>

```

```

<TR><TD>6</TD><TD>12</TD><TD>18</TD><TD>24</TD><T
D>30
</TD></TR>
<TR><TD>7</TD><TD>14</TD><TD>21</TD><TD>28</TD><T
D>35
</TD></TR>
<TR><TD>8</TD><TD>16</TD><TD>24</TD><TD>32</TD><T
D>40
</TD></TR>
<TR><TD>9</TD><TD>18</TD><TD>27</TD><TD>36</TD><T
D>45
</TD></TR>
<TR><TD>10</TD>><TD>20</TD><TD>30</TD><TD>40</TD>
<TD>
50</TD></TR>
</TABLE>
</BODY>
</HTML>

```

Oddiy jadval murakkab ko'p darajali strukturani tashkil etuvchi bir-biriga taxlangan ko'plab teglardan iborat. Endi jadval qanday teglar bilan shakllanishini ko'rib chiqamiz.

Web ilova:

Ko'paytirish jadvali (1 dan 5 gacha)

No	2x	3x	4x	5x
1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25
6	12	18	24	30
7	14	21	28	35
8	16	24	32	40
9	18	27	36	45
10	20	30	40	50

4.4-rasm. Ko'paytirish jadvali

Jadval tegi quyidagi format (shakl)ga ega:

```

<TABLE [BORDER= “{ramka qalinligi}”]>
  [CELLPADDING= “{yacheyka ichidagi yozuvlar va chegarasi
orasidagi masofa}”]
  [CELLSPACING= “{yacheyka orasidagi masofa}”]
  [WIDTH= “{jadvaldan kengligi}”] [HEIGHT= “{jadval
balandligi}”]
  [ALIGN =“left /Center/ right”] [BGCOLOR=“{jadval foni rangi}”]
  [BORDERCOLOR= “{ramka rangi}”] [BACKGROUND= “{fonli
rasm manzili}”]
  [FRAME= “none/above/below/hsides/lhs/rhs/vsides/vsides/bat”]
  [RULES= “none/rows/cols/all”]>
<!--satrni aniqlash-->
<TABLE>

```

Endi bu tegning atribut (parametr)larini ko'rib chiqamiz.

Atribut BORDER jadval ramkasi qalinligini piksellarda beradi. Agar u ishtirok etmasa, ramka ko'rinmaydi, lekin brauzer (Web-sharhlovchi) aks ettirishda ramka mavjud deb hisoblaydi. Agar bu atributga 0 qiymatni bersak, u holda ramka jadvalda butunlay bo'lmaydi.

Atribut CELLPADDING yacheyka ichidagi narsalar bilan ramka orasidagi bo'sh oraliq o'lchamini piksellarda beradi. Standart bo'yicha u 1 ga teng.

Atribut CELLSPACING jadval yacheykalari orasidagi bo'sh oraliqni piksellarda ko'rsatadi. Standart bo'yicha u 2 ga teng.

Atributlar WIDTH va HEIGHT jadval o'lchamini o'rnatishga imkon beradi.

O'lchamni brauzer darchasi o'lchamlariga nisbatan absolyut kattalik ko'rinishida piksel va foizda ko'rsatishga ruxsat beriladi, (foiz belgisi % ni qo'yish esdan chiqmasin). Agar bu atributlar tushirib qoldirilgan bo'lsa, u holda brauzerning o'zi jadval o'lchamini qo'yadi.

Atribut ALIGN jadvalni matn bilan beradi. Left qiymatni brauzer darchasining chap chekkasiga o'tishga majbur etadi. Matn esa uni o'ng tomondan qarshiliksiz aylanib o'tadi. Right qiymati teskari joylashtirsa, Center esa jadvalni sahifa markaziga joylashtiradi.

Atribut BGCOLOR va BACKGROUNDlar mos ravishda fon rangini va jadval uchun fonli rasm manzilini beradi.

Atribut BORDERCOLOR jadval ramka rangini beradi—u W3C tomonidan standartlashtirilgan va uni faqat Internet Explorer quvvatlab turadi.

Atribut FRAME jadval ramkasining qaysidir tashqi qismi chizmasini

aniqlaydi. Quyidagi qiymatgacha ruxsat berilgan:

- void–tashqi ramka butunlay yo'q;
- above–tashqi ramkaning faqat yuqori chizig'i chiziladi;
- below–tashqi ramkaning faqat pastki chizig'i chiziladi;
- hsides–tashqi ramkasining faqat gorizontal chizig'i chiziladi, ya'ni yuqori va quyi chiziqlari;
- lhs–tashqi ramkaning faqat chap chizig'i chiziladi;
- rhs–tashqi ramkaning faqat o'ng chizig'i chiziladi;
- vsides–tashqi ramkasining faqat vertikal chizig'i chiziladi;
- box–tashqi ramkaning faqat hamma chiziqlari chiziladi.

Bu atribut W3C tomonidan standartlashtirilgan bo'lsada, uni faqat Internet Explorer quvvatlaydi.

Atribut RULES jadval ramkasi ichki qismining qanday chizilishini ko'rsatadi.

Bunda quyidagi qiymatlarga ruxsat beriladi:

- none–hech qanday ichki ramkalar bo'lmaydi;
- rows–faqat gorizontal chiziqlar chiziladi, (satrlar orasidagi);
- cols–faqat vertikal chiziqlar chiziladi, (ustunlar orasidagi);
- ale–barcha ichki ramkalar chiziladi. Bu atribut ham W3C

tomonidan standartlashtirilgan bo'lsada, uni Internet Explorer quvvatlab turadi.

Jadval ichidagi ma'lumotlarni formatlash. Jadval ichidagi har bir alohida yacheykani mustaqil formatlash uchun ajratilgan o'rin deb hisoblash mumkin. Matnni aks ettirishni boshqarishdagi barcha qoidalar yacheyka ichidagi matnni formatlashda ishlatilishi mumkin. Yacheyka ichida <BODY> hujjati tegi ichida paydo bo'lib qoladigan HTMLning hamma elementlarini ishlatish mumkin. Shu jumladan, <p>,
, <HR> matnlar joylashuvini boshqaradigan teglar; <H1> dan <H6>gacha bo'lgan sarlavha kodlari;

, <I>, , <BIG>, , , simvollarni formatlash teglari; grafik tasvirni; <A> va shu kabi gipermatn aloqalarni o'rnatish teglari.

Alohida yacheykalar ichidagi teglarning ta'sir etish joyi yakunlovchi tegning mavjudligiga qaramay, shu yacheyka imkoniyatlari bilan chegaralanadi.

Masalan, yacheyka ichida matn rangi aniqlangan bo'lsa, u holda xattoki yakunlovchi kod </ FONT> yoki uning bir nechta yacheyka yoki jadval satri orqali joylashuvi bo'lmasa-da, navbatdagi yacheyka matni standart rangi bilan aks ettiriladi.

Jadval yacheykalari ichidagi ma'lumotlarni formatlash uchun quyidagi parametr (atribut)lar mo'ljallangan:

Yacheykadagi qiymatlarni rostdash parametrlar–ALIGN va VALIGN. <TR>, <TD> va <TH> kodlarida qo'llanishi mumkin. Gorizontaal bo'yicha rostdash parametrlari–ALIGN. LEFT, RIGHT va CENTER esa, qiymatlarini qabul qilishi mumkin, (standart bo'yicha LEFT <TD> uchun CENTER <TH> uchun). Vertikal rostdash parametri–VALIGN TOP (yuqori qirra bo'yicha), BOTTOM (pastki qirra bo'yicha), MIDDLE (markaz bo'yicha), BASELINE (baza chizig'i bo'yicha), MIDDLE (standart bo'yicha) qiymatlarini qabul qilib olishi mumkin. Baza chizig'i bo'yicha rostdash hamma yacheykalardagi alohida satr matnlarini bitta chiziqqa birlashtirishni ta'minlaydi. <TR> kodi darajasida rostdash parametrlarining berilishi berilgan satrlarning barcha yacheykalari uchun rostdashni aniqlaydi.

Bunda har bir alohida satr yacheykasida o'zining parametri aniqlanishi (<TR> da berilgan parametrlar harakatini qaytadan aniqlash) mumkin.

Misol keltiramiz. Bu misolda berilgan jadval ma'lumotlari birinchi ustun yacheykalarida o'ng tomonga, ikkinchi ustunda markazga, uchinchi ustunda esa, chap tomonga to'g'rilangan, (standart qiymati bo'yicha):

```
<HTML>
<HEAD> <TITLE> jadval elementlarini rostdash </TITLE>
<BODY>
<TABLE BORDER WIDTH=100%>
<TR>
<TD ALIGN=RIGHT> yacheyka 1 </TD>
<TD ALIGN=CENTER> yacheyka 2 </TD>
<TD> yacheyka 3 </TD>
</TR>
<TR>
<TD ALIGN=RIGHT> yacheyka 4 </TD>
<TD FLIGN=CENTER> yacheyka 5 </TD>
<TD> yacheyka 6 </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Murakkab jadvallar uchun bir nechta o'xshash yacheykalarni gorizontaal va vertikal bo'yicha bittaga birlashtirishga zaruriyat mavjudligi

ma'lum. Bu berilgan imkoniyat <TD> yoki <TH> beriladigan COLSPAN (COLUMN SPANNING) va ROWSPAN (ROW SPANNING) parametrlari yordami bilan amalga oshiriladi. Yozish formasi (shakli): COLSPAN=num; bu yerda num–joriy yacheykani gorizonta bo'yicha nechta ustunga kengaytirishni ko'rsatuvchi sonli qiymat.

ROWSPAN parametrining qo'llanilishi shu kabi bajariladi, faqat bu yerda vertikal bo'yicha joriy yacheykalarni o'z ichiga olishi kerak bo'lgan satrlar miqdorini ko'rsatish kerak bo'ladi. Standart bo'yicha bu parametrlar uchun birga teng bo'lgan qiymat o'rnatiladi. Bitta yacheyka uchun ikkala parametrlar qiymatini bir vaqtda berilishi mumkin.

Yacheykalarni birlashtirishga misol. Bu sahifani tbl2.html nomi bilan saqlab qo'yamiz:

```
<HTML>
<HEAD>
<TITLE> yacheykalarni birlashtirish </TITLE>
</HEAD>
<BODY>
<TABLE BORDER>
<TR>
<TD ROWSPAN =2> ikki satrni egallovchi yacheyka </TD>
<TD COLSPAN=2> ikki ustunni egallovchi yacheyka </TD>
</TR> <TR>
<TD> yacheyka 3 </TD>
<TD> yacheyka 4 </TD>
</TR>
<TR>
<TD> yacheyka 5 </TD>
<TD> yacheyka 6 </TD>
<TD> yacheyka 7 </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Ikki satrni egallovchi yacheyka	Ikki ustunni egallovchi yacheyka	
Yacheyka 5	yach. 3	yach. 4
	yach. 6	yach. 7

4.5– rasm. Bir nechta satr va ustunlarga taqsimlanuvchi yacheykalar jadvali

MS Internet Explorerga xos yana bir qancha parametrlar BORDERCOLOR, BORDERCOLORLIGHT va BORDERCOLORDARK jadvallar ramkasi rangini tanlashga imkon beradi. Bu parametrlar <TABLE>, <TD>, <TH> va <TR> teglarida berilishi mumkin. Bu parametrlarning qiymati sifatida rang nomi yoki o'noltilik sonlari ishlatilishi mumkin. BORDERCOLOR parametri jadval ramkasi elementlarining rangini aniqlasa, qolgan ikkita parametrlar esa, BORDERCOLOR qiymatini qaytadan aniqlab, ramkaning qolgan alohida qismlari uchun xizmat qiladi. Parametr BORDERCOLORLIGHT hamma jadvallarning chap va yuqori qirralarini mos ravishda har bir yacheykaning o'ng va pastki qirralarini ko'rsatilgan rangga bo'yaydi. Ikkinchi parametr BORDERCOLORDARK esa, qarama-qarshi qirralarga rang beradi. Bu ikki parametr harakatining birlashuvi oqibatida jadval sahifa satridan yuqoriroq (yoki quyiroq)da ko'rinadi. Bularning barchasi tanlangan ranglar birikishiga bog'liq.

Brauzer MS Internet Explorer (shu jumladan, 4.x versiyali Netscape brauzeri ham) barcha HTML hujjatlar uchun qo'llaniladigan ishlar kabi jadval rasmi fonini aniqlovchi BACKGROUND parametridan foydalanishga ruxsat beradi. Bu parametr <TABLE>, <TD>, <TH> va <TR> teglarida berilishi mumkin.

Yana bitta jadval misolini ko'rib chiqamiz: (uni tbl 3. html nomi bilan saqlaymiz)

```
<HTML>
<HEAD>
<TITLE> Fifth document </TITLE>
</HEAD>
<BODY>
<H1> Jadval misoli </H1>
<!--jadval asosi uchun chap teg parametrlari bilan-->
<TABLE BORDER= 2 CELLSPACING=3 CELLPADDING=3>
<!--jadval sarlavhasi va sarlavha ramkasi-->
<CAPTION ALIGN=TOP> <H3> Talabalar ro'yxatini </H3>
</CAPTION>
<TR>
<TD></TD>
<TH> bo'yi </TD>
<TH> yoshi </TH>
</TR>
```

```

<TR>
<TH> Rahimov </TN>
<TD> 175 </TD>
<TD> 21 </TD>
</TR>
<TR>
<TH> Nasriddinov </TN>
<TD> 170 </TD>
<TD> 23 </TD>
</TR>
<TR>
<TH> Hamidov </TH>
<TD> 169 </TD>
<TD> 19 </TD>
</TR>
<TR>
<TH> Karimova </TN>
<TD> 171 </TD>
<TD> 20 </TD>
</TR>
<TR>
<TH> Hasanova </TN>
<TD> 169 </TD>
<TD> 22 </TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Natijada ekranda quyidagi ko'rinish paydo bo'ladi:
 Jadval misoli: talabalar ro'yxati:

Jadval misoli

Talabalar ro'yxatini

	bo'yi	yoshi
Rahimov	175	21
Nasriddinov	170	23
Hamidov	169	19
Karimova	171	20
Hasanova	169	22

4.6-rasm. Talabalar ro'yhati jadvali

NAZORAT SAVOLLARI

1. Html tilida matnlar bilan ishlash?
2. Taxlangan ro'yxatlar va ular bilan ishlovchi teglar?
3. Aniqlash ro'yxati va ular bilan ishlovchi teglar?
4. Raqamlangan ro'yxat deganda nimani tushunasiz?
5. Html da jadval tegi va uning atributlari?
6. Jadvaldagi yacheykalarni birlashtirish?
7. Jadval ichidagi ma'lumotlarni formatlash?

5-BOB. HTMLDA FREYM SAHIFALASH, FORMA VA OB'YEKTLAR

5.1. HTML formalar va ularning atributlari. HTML forma ob'yektlari va atributlari

Formalar Web dasturlashda foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni tartibga solish uchun qo'llaniladi. Forma elementlari to'ldirilgandan keyin undagi ma'lumotlar serverdagi ma'lumotlarni qayta ishlovchi dasturga yuboriladi. Ko'p sonli jo'natilayotgan ma'lumotlar jo'natish tugmasi bosilgandan so'ng serverda joylashgan Common Gateway Interface (CGI) yordamida yoki maxsus fayl orqali qayta ishlanadi. Shu tariqa foydalanuvchi Internet orqali Web-server bilan birgalikda ishlaydi.

Formaning berilishi - FORM elementi. FORM elementi hujjatni ma'lum bir *formaga* soladi va *forma* elementlari teglarini boshqa teglardan ajratib turadi. <FORM> bir nechta <INPUT> yoki shu kabi boshqa teglar ketma ketligidan tashkil topadi. Ular <FORM> va </FORM> teglari orasiga joylashtiriladi. Formada usul (method), *formaga* kiritilgan ma'lumotlarni qayta ishlash uchun holatlar (action) mavjud. Usul (GET yoki POST) formaga kiritilgan ma'lumotlar qay tarzda serverga jo'natilish usulini belgilasa, holat esa serverdagi qaysi dasturga yuborilishi URL (Uniform Resource Location) manzilini ifodalaydi.

<FORM METHOD="post" ACTION="<URL>">

Formaning boshqaruv elementlari - <INPUT>tegi. Ushbu teg *formaning* qaysi nuqtasiga ma'lumot kiritilishini belgilaydi. U foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni formaga keltiradi. Bular matn kiritish maydoni, ro'yxatlar, rasmlar yoki tugmalar bo'lishi mumkin. Maydon turi TYPE atributi yordamida aniqlanadi.

Formaning TYPE=text atributi. Agar foydalanuvchi uncha katta bo'lmagan matn kiritsa (bir yoki bir nechta satr), <INPUT> tegidan foydalanadi va TYPE atributiga text qiymati o'zlashtiriladi. Standart holat uchun bu qiymatni berish muhim emas. Bundan tashqari *maydonni* nomlash va unga murojaat qilish uchun NAME atributi ham beriladi.

Sizning ismingiz <INPUT NAME=Name SIZE=35>.

Foydalanish mumkin bo'lgan yana uchta qo'shimcha atributlar mavjud. Birinchisi MAX LENGTH deb ataladi, u foydalanuvchi kiritayotgan matn maydoni maksimum uzunligini belgilaydi. Standart bo'yicha bu qiymat chegaralanmagan. Ikkinchi atribut SIZE hisoblanadi, u

esa matn maydonini ko'rinib turuvchi qismini belgilaydi. Standart bo'yicha uning qiymati brauzerga bog'liq bo'ladi. Agar MAX LENGTH qiymati SIZE qiymatidan katta bo'lsa, brauzer ma'lumotni oynaga moslashtiradi. So'ng qo'shimcha atribut matn maydonini boshlang'ich qiymatini belgilovchi VALUEdir.

Formaning TYPE=checkbox atributi. HTML *formada* mustaqil belgilagich (bayroqcha) dan foydalanish uchun `<INPUT>` tegining atributiga TYPE=checkbox ni o'zlashtirish kerak. Formaga bog'liq ravishda foydalanuvchi bir yoki bir nechta belgilagichlarni belgilashi mumkin. Agar `<INPUT>` tegi atributi bilan CHECKBOX qiymati qo'llanilsa, u bilan birga NAME va VALUE atributlari ham qo'llanilishi kerak. NAME atributi ushbu ma'lumot kiritish ob'yektining nomini ifodalaydi. VALUE atributida ushbu *maydonning* qiymati ko'rsatiladi.

```
<BR>O'zbekiston<INPUT NAME="Davlat" TYPE=checkbox  
VALUE="O'zbekiston">
```

```
Mamlakatlar SNG<INPUT NAME="Davlat" TYPE=checkbox  
VALUE="SNG">
```

Ba'zi hollarda ushbu maydon belgilangan holda qo'llanilishi ham mumkin. Bunday hollarda `<INPUT>` tegida CHECKED atributi qo'llanilishi kerak.

Formaning TYPE=radio atributi. Ba'zan bir nechta qiymatlar orasidan birini tanlashga to'g'ri keladi. Bunday hollarda formada `<INPUT>` tegi bilan birga TYPE=radio atributi qo'llaniladi. Agar `<INPUT>` tegi atributi bilan ushbu qiymat qo'llanilsa, u bilan birga NAME va VALUE atributlari ham qo'llanilishi kerak. NAME atributi ushbu ma'lumot kiritish ob'yektining (*tugma*) nomini ifodalaydi. VALUE atributida ushbu *maydonning* qiymati ko'rsatiladi..

```
<BR>Erkak jinsi<INPUT NAME="Jins" TYPE=radio  
VALUE="Erkak">
```

```
Ayol jinsi<INPUT NAME="Jins" TYPE=radio  
VALUE="Ayol">
```

Formaning TYPE=image atributi. Formaning tarkibiga qarab ba'zan unda joylashgan rasmning ustiga sichqonchani bosish bilan undagi ma'lumotlarni jo'natishga to'g'ri kelib qoladi. Buning uchun `<INPUT>` tegi TYPE=image atributi bilan qo'llaniladi. Foydalanuvchi rasm ustiga sichqoncha kursorini bossa, aynan shu yerdagi ekran koordinatalarini brauzer saqlab qoladi. So'ng formaga kiritilgan ma'lumotlarni "qayta ishlaydi". Agar `<INPUT>` tegi image atributi bilan qo'llanilsa, u bilan birga NAME va SRC atributlari ham qo'llanilishi kerak. NAME

maydonning nomini belgilaydi. SRC atributi esa rasm joylashgan manbaning URL manzilini beradi. ALIGN atributi qo'shimcha hisoblanadi va u ham ba'zan tegi bilan qo'llaniladi.

```
<BR>Nuqtani tanlang<INPUT TYPE=image NAME=point  
SRC=image.gif>
```

Formaning TYPE=password atributi. Agar *formada* parollardan foydalanish kerak bo'lib qolsa, TYPE atributi qiymatiga password (TYPE=password) ni o'zlashtiriladi. Ushbu turdan foydalanish kiritilayotgan ma'lumotni oshkor bo'lmagan holda ko'rsatishni ta'minlayadi. Shu sabab, kiritilgan ma'lumot ochiq kanal orqali jo'natiladi va ushbu ma'lumot tutib olinishi mumkin.

```
<BR> Nomingiz <INPUT NAME=login> Parol  
<INPUT TYPE=password NAME="So'z">
```

Formaning TYPE=reset atributi. Baza foydalanuvchi *formani* to'ldirish vaqtida, ularni boshdan to'ldirishga to'g'ri keladi. Ushbu holda Reset tugmasi mavjud bo'lib, bu tugmaning bosilishi formani dastlabki, kirish holatiga olib keladi (formani "tozalaydi"). Reset tugmasini tashkil qilish uchun <INPUT> tegi atributiga TYPE=reset o'zlashtiriladi. Agar *formada* reset atributi qo'llanilsa, <INPUT> tegiga VALUE atributini qo'shimcha qilish mumkin. Ushbu atribut tugmadagi yozuvni ifodalaydi.

```
<INPUT TYPE=reset VALUE="Formani tozalash">
```

Formaning TYPE=submit atributi. HTML *forma* da foydalanuvchi ma'lumot kiritish jarayonini yakunlash jarayoni mavjud. Buning uchun <INPUT> tegining atributiga TYPE=submit qiymat o'zlashtiriladi. Agar *formada* <INPUT> tegi submit atributi bilan qo'llanilsa, unga qo'shimcha ravishda ikkita atributdan foydalanish mumkin: NAME va VALUE. NAME atributi *maydonning* nomini ifodalaydi. VALUE atributi — Submit tugmasi matnini ko'rsatadi.

```
<INPUT TYPE=submit VALUE="Xabarni jo'natish "/>
```

Formaning TYPE=hidden atributi. Yashirin *maydon*. INPUT tegini TYPE=hidden atributi bilan qo'llanilishi foydalanuvchiga ma'lum bo'lmagan NAME va VALUE atributlaridagi qiymatlarni jo'natishga imkon beradi.

Formaning <TEXTAREA> – ko'p satrli matn kiritishni tashkil etish tegi.

Ba'zan *formada* ko'p satrli matnlarni kiritish talab etiladi. Buning uchun <TEXTAREA> tegi yordamida bir necha satrdan iborat bo'lgan matn maydoni tashkil etish mumkin. Ushbu teg uchta atribut bilan ishlatiladi: COLS, NAME va ROWS.

Atribut COLS. Maydonning ustunlari (belgilar soni) sonini belgilaydi.

Atribut NAME. Maydonning nomini belgilaydi.

Atribut ROWS. Maydonning ko'rinuvchi satrlari sonini belgilaydi.

```
<TEXTAREA NAME=mavzu COLS=38
ROWS=3></TEXTAREA>
```

<SELECT>- formada ro'yxatdan foydalanish tegi. Agarda *forma* mukammal bo'lsa, gohida unda harakatlanuvchi ro'yxat ham qo'llaniladi. Buning uchun *SELECT* tegidan foydalaniladi. Ro'yxat bo'limlarini aniqlash uchun *<OPTION>* tegidan foydalaniladi. *<SELECT>* tegi muhim bo'lmagan uchta atributni qo'llab quvvatlaydi: *MULTIPLE*, *NAME* va *SIZE*.

MULTIPLE atributi. Bir vaqtning o'zida bir nechta variantni tanlash imkonini beradi.

NAME atributi. Ob'yekt nomini ifodalaydi.

SIZE atributi. Ro'yxatni ko'rinuvchi satrlari sonini ifodalaydi. *<SIZE>* 1 bo'lgan holda brauzer oddiy ro'yxatni ko'rsatadi. *Formada <OPTION>* tegi faqat *<SELECT>* teglari orasida qo'llaniladi. Parametrlari: *SELECTED* va *VALUE*.

SELECTED atributi. Dastlabki holatda ushbu element tanlangan ekanligini bildiradi.

VALUE atributi. Ro'yxatga o'zlashtirilishi mumkin bo'lgan qiymatni ifodalaydi.

```
<BR>Tanlash
```

```
<SELECT NAME="Tanlash">
```

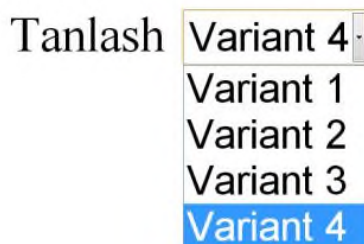
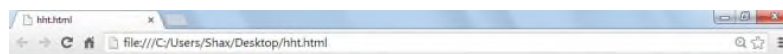
```
<OPTION>Variant 1
```

```
<OPTION>Variant 2
```

```
<OPTION VALUE="Variant 3">Variant 3
```

```
<OPTION SELECTED>Variant 4
```

```
</SELECT>
```



5.1-rasm. Qalqib chiquvchi menyu.

5.2. Freymlar. Freym tegi sintaksisi va atributlari. Sahifalarni freym yordamida hosil qilish

Freymlar brauzerni kuzatuv oynasini yonma - yon joylashgan bir nechta to'g'ri burchakli sohalarga bo'lish imkonini beradi.

Mazkur bo'laklardan har biriga alohida HTML-fayl, ya'ni boshqalardan mustaqil ravishda ko'zdan kechiriluvchi fayllarni yusinfh mumkin. Zaruriyat tug'ilganda freymlar orasida o'zaro bog'liqlikni tashkil etish mumkin. O'zaro bog'liqlik tashkil etilganda freymlardan birida xavola (ssilka) tanlansa, boshqa freym oynasida kerakli hujjatning yuklanishiga olib keladi.

Garchi HTML-hujjatlarda foydalanuvchiga axborot aks ettirilishining turli usullari xavola (ssilka) etilsada, axborotni ifodalashning freym tizimi ham o'zining afzalliklariga ega. Quyidagi hollarda aynan freym tizimi qo'l keladi:

- Bir soxada ishlayotganda boshqa bir soxaga hujjatlarni yusinfh orqali boshqarishni tashkil etish zarurati tug'ilganda;
- Ekraning boshqa hududlarida nima bo'lishidan qat'iy nazar ekranda doimo ko'rinib turishi kerak bo'lgan axborotni ko'zdan kechirish darchasining ma'lum qismiga joylashtirish lozim bo'lganda;
- Darchaning har biri mustaqil ravishda ko'rib chiqilishi mumkin bo'lgan yonma-yon bir necha soxalarda joylashtirish qulay bo'lgan axborotni taqdim etish zarurati tug'ilganda.

Freymlar tizimini tasvirlash uchun <FRAMESET>, <FRAME> yoki <NOFRAME> teglaridan foydalaniladi. <FRAMESET> tegi freymlarni belgilaydi. Freymlardan tashkil topgan Web-sahifalar <BODY> bo'linmasiga ega bo'lishi mumkin emas. <FRAMESET> va </FRAMESET> konteynerlari har bir freymni belgilash blokini o'rab turadi. Bunday konteynerning ichida faqat <FRAME> teglari yoki kiritilgan <FRAMESET> teglari mavjud bo'ladi. <FRAMESET> tegining atributlari:

ROWS, COLS. Ushbu parametrlar qiymatlari piksellarda, foizlarda yoki nisbiy birliklarda berilishi mumkin. Qator yoki ustunlar soni mos ro'yxatdagi qiymatlar soni bilan aniqlanadi.

Masalan:

<FRAMESET ROWS = "100, 240, 140"> - uchta freymdan iborat to'plamni belgilaydi. Qiymatlar piksellarda berilgan. Birinchi freym 100 piksel, ikkinchisi 240 piksel va nihoyat so'nggi freym 140 piksel balandlikka ega.

<FRAMESET ROWS = "25%, 50%, 25%"> - ekranning maqbul balandligidan yuqori qatorning qiymati 25 foiz, o'rta qatorniki 50 foiz, quyi qatorniki 25 foiz ekanligini bildiradi.

<FRAMESET COLS = "*, 2*, 3*"> - qiymatlar nisbiy birliklarda. "Yo'lduzcha" – "*" fazoni proporsional taqsimlash uchun ishlatiladi. Har bir yo'lduzcha butunning bir qismini bildiradi. Hisoblab topish uchun yo'lduzchalar oldidagi sonlarni qo'shish va hosil bo'lgan sondan kasrning maxraji sifatida foydalaniladi. Yuqoridagi misolda birinchi ustun darcha umumiy kengligining 1/6, ikkinchi ustun 2/6, uchinchi ustun 3/6 qismini egallaydi.

<FRAMESET COLS = "100, 25%, *, 2*">.

<FRAME> tegi aloxida fayllarni belgilaydi, bu teg <FRAMESET> va </FRAMESET> teglari juftligining ichida joylashishi lozim.

Masalan:

<FRAMESET ROWS = "*, 2*">

<FRAME>

</FRAME>

</FRAMESET>

<FRAMESET> tegi berilganida qancha alohida freymlar belgilangan bo'lsa, shuncha freym teglarini yozish lozim.

<FRAME> tegi atributlari:

SRC

NAME

MARGINWIDTH
MARGINHEIGHT
SCROLLING
NORESIZE

FRAMEBORDER=YES/NO (Faqat IE lar uchun)

SRC atributi boshidan boshlab Mazkur freymga yuklanuvchi hujjatning URL-manzilini belgilaydi. Odatda bunday manzil sifatida asosiy hujjat qaysi katalogda bo'lsa, o'sha yerda joylashgan HTML-faylning nomidan foydalaniladi.

Masalan:

```
<FRAMESET SRC="sample.html">
```

Zero, freymni tasvirlashda berilgan HTML-fayl to'liq HTML-hujjat bo'lishi kerak, ya'ni u HTML, HEAD, BODY va boshqalarga ega bo'lishi lozim. Agar freymdan tasvirni aks ettirishda foydalanilsa, unda:

```
<FRAME SRC="http://www.bhv.ru/ehampl.gif">
```

NAME parametri berilgan freymga xavola (ssilka) sifatida ishlatish mumkin bo'lgan freymning nomini belgilaydi.

Masalan:

```
<FRAME SRC="sample.html" NAME="frame1">
```

frame1 deb nomlangan ushbu freymga xavola (ssilka) qilinishi mumkin.

Masalan:

```
<A HREF="other.html" TARGET="frame1">
```

frame1 freymiga other.html faylini yusinh uchun shu yerga sichqoncha kursori bosiladi.

MARGINWIDTH va MARGINHEIGHT atributi freym xoshiya (chegara) kengligini belgilaydi. Atributlar qiymatlari piksellarda beriladi.

Masalan:

```
<FRAME MARGINWIDTH = "5" MARGINHEIGHT = "7">
```

Bu yerda freym yuqori va pastda 5 piksel, o'ng va chap tomonlaridan esa 7 piksel chegaraga ega. Ishlatilishi mumkin bo'lgan eng kichik qiymat 1 pikseldir.

SCROLLING atributidan prokrutka yo'laklarini aks ettirishni boshqarishda foydalaniladi.

Uning sintaksisi

```
<FRAME SCROLLING= "YES/ NO/ AVTO"> ko'rinishga ega.
```

NORESIZE atributi foydalanuvchi tomonidan freym o'lchami o'zgartirilishining oldini olishda ishlatiladi.

Masalan:

<FRAME NORESIZE>.

Tabiiyki NORESIZE atributining bitta freymga nisbatan qo'llanilishi boshqa freymlar o'lchami o'zgartirilishining ham oldi olinishiga sabab bo'ladi. Garchi freymlar tizimi HTML 4.0 da standart bilan mustaxkamlangan bo'lsada, <NOFRAMES> tegi freymnlarni qo'llab-quvvatlamaydigan brauzerlar yordamida ko'zdan kechirishda asqotadi. Demak, freymlarga bog'lanmagan brauzerlar uchun <NOFRAMES> va </NOFRAMES> teglari juftligidan foydalaniladi.

Masalan:

<NOFRAMES> butun HTML-hujjat </NOFRAMES> mazkur teglar orasiga joylashtirilgan barcha ma'lumotlar freymnlarni qo'llab-quvvatlash imkoniyatiga ega bo'lmagan brauzerlar yordamida aks ettiriladi. Freymlarga bog'langan brauzerlar esa <NOFRAMES> va </NOFRAMES> orasidagi barcha axborotga bog'liq emas.

Freymlar bilan ishlayotganda foydalanuvchi uchun qulay bo'lgan hujjat yusinfh sxemasini yaratish mumkin. Freymlar orasidagi o'zaro aloqa hujjatlarni boshqa freymdagi buyruqlar yordamida aynan tanlangan freymga yusinfh imkonini berishidadir. Bu maqsadda <A> tegining TARGET atributidan foydalaniladi. TARGET atributi ushbu xavola (ssilka) ko'rsatayotgan hujjat yuklanuvchi freym yoki brauzer oynasi nomini belgilaydi.

O'zgartirilmagan holda ushbu parametr yo'q bo'lganda hujjat joriy freym yoki oynada yuklanadi. Freym nomi sifatida mavjud darcha yoki freym nomi berilishi yoki bo'lmasa yangi oyna ochish uchun yangi nom berilishi mumkin. 4 ta zaxiradagi nomlar bor. Ular: _blank, _self, _top, _parent. Bulardan tashqari “_” belgisi bilan boshlanuvchi har qanday nomdan foydalanish maqsadga muvofiq emas. TARGET=“_blank” – hujjatning yangi oynaga yuklanishini ta'minlaydi. Bu oyna nomga ega bo'lmasligi tufayli unga boshqa hujjatni yusinfhning iloji bo'lmaydi. TARGET=“_self” dan foydalanilganda hujjat joriy freymga yuklanadi. TARGET=“_top” hujjatning butun darchaga yuklanishiga sabab bo'ladi. TARGET=“_parent” hujjatning joriy freymning freym-otasonasi tomonidan egallangan soxasiga yuklanishiga olib keladi.

NAZORAT SAVOLLARI

1. Html da formalar bilan ishlovchi teglar va ularning atributlari?
2. TYPE=radio atributining vazifasi?

3. TEXTAREA tegining vazifasi?
4. Html da freymlarni tashkil etish?
5. Html da freym tegi atributlari va ularning vazifalari?

6-BOB. HTML5 YANGI STANDARTLAR. TEG VA ATRIBUTLAR. GRAFIKA, AUDIO VA VIDEO BILAN ISHLASH. VIZUAL EFFEKTLAR

6.1. Yangi HTML5 razmetkani qo'llash. Yangi strukturali teg va atributlar

Hozirgi kunga kelib web texnologiya juda katta tezlikda rivojlanib bormoqda. Har kuni yangi saytlar butun olam o'rgimchak to'riga qo'shib borilmoqda. Saytlarning asosini html gipermatn tili tashkil etadi. Ko'p yillar davomida sayt yaratish uchun html 4 tilidan foydalanib kelingan edi. 2004 yilda esa yirik kompaniyalar (Apple, Opera, Mozilla) birlashib yangi guruh hosil qilishdi va bu guruhni WHAT WG (Web Hypertext Application Technology Working Group) deb nomlashdi. So'ng bu guruhga W3C (World Wide Web Consortium) ham kirib keldi va yangi html 5 ni ishlab chiqarish boshlandi. HTML 5 da flash pleyerlar o'rnini teglar egalladi va sayt strukturasi uchun alohida teglar qo'shildi.

HTML5 tilida sahifalarni yaratishda eng birinchi elementi bo'lgan `<!DOCTYPE HTML>` kodini yozish zarur. Yana eng muhim elementlaridan biri belgilarni `<meta charset= "UTF-8">` orqali **UTF-8** kodlariga o'girish zarur. Agar **UTF-8** ni yozib ketilmasa web-sahifada kirilcha ma'lumotlar kiritilsa, ular tushunib bo'lmas belgilarga aylanib qoladi. Shuning uchun doimo `<meta charset= "UTF-8">` kodini yozish maqsadga muvofiq bo'ladi.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
</head>
<body>
Content of the document.....
</body>
</html>
```

HTML5 ning yangi elementlari:

`<article>`, `<aside>`, `<audio>`, `<canvas>`, `<datalist>`, `<details>`,
`<embed>`, `<footer>`, `<header>`, `<nav>`, `<output>`, `<progress>`, `<section>`,
`<video>` va boshqalar.

HTML5da HTML4 ga nisbatan formalar bilan ishlashda foydalanuvchilar uchun ancha qulayliklar kiritilgan. Misol uchun: Formalarni yaratish va foydalanuvchilar ma'lumotlarini olish juda osonlashtirildi. Sanalarni tanlash, rang tanlash va shu kabi bir qancha qo'shimchalar qo'shilgan. Kiritish maydonlariga pochталarni qidirish va URL olish imkoniyatlari qo'shilgan. Hamda PUT va DELETE formalar uchun yangi funksiyalar yaratilgan.

Quyida HTML5 ning yangi teglari keltirilgan:

article tegi-saytning tarkibi haqida ma'lumot (yangiliklar, maqolalar, bloglar).

aside tegi-kontentning yonboshdan joylashuvchi ma'lumot, arxivga gipermurojaat, belgilash.

audio tegi-web sahifadagi audioyozuvlarni qo'shish, eshittirish va sozlash.

bdi tegi-mattni chiqarishda yo'nalishning o'zgarishini oldini oladi.

canvas tegi-JavaScript orqali turli xildagi ob'yektlarni chizish, rasm chiqarish, transformasiya va xususiyatlarni o'zgartirish imkoniyatiga ega maydon yaratadi.

command tegi-o'chirib–yoqqich, bayroqcha yoki oddiy tugmalarni qo'yish uchun buyruq.

datalist tegi-matnli maydonda kiritish sharoitida ochiluvchi ro'yhat yaratadi.

details tegi-foydalanuvchi talabiga ko'ra yashirish yoki ko'rsatishga mo'ljallangan.

embed tegi-ob'yektlarni ko'rsatish va yusinfh uchun foydalanadi.

figcaption tegi-chiqarish maydonini aniqlaydi <figure> tegida foydalanish.

figure tegi-har qanday elementlarni guruhlarga birlashtirish uchun foydalaniladi masalan tasvirlarni.

footer tegi-saytning “quyi” qismi bo'limini aniqlaydi.

header tegi-saytning “yuqori” qismi bo'limini aniqlaydi.

hgroup tegi-web sahifaning yoki bo'limni guruhlarga birlashtirish uchun foydalanadi.

keygen tegi-ochiq va yopiq juft kalitlarni yaratish uchun foydalaniladi.

main tegi-hujjatning asosiy tarkibi uchun mo'ljallangan <main> elementi.

Ichki qismi qaytarilmas bo'lishi kerak.

mark tegi-mattni belgilanganday qilib ko'rsatadi.

menu tegi- menyuning ro'yhatini ko'rsatish.

meter tegi-ma'lum oraliqda qiymatlarni chiqarish uchun foydalanadi.

nav tegi-sayt bo'yicha navigatsiya xavola (ssilka)larni (gipermurojaatni) guruhlaydi.

output tegi-chiqarish maydonini belgilaydi. (iloji bo'lsa skriptlar orqali).

progress tegi-vazifaning tugaganlik sharoitini ko'rsatish uchun foydalaniladi.

rp tegi-`<ruby>` tegini qo'llay olmaydigan brauzerlar uchun matn chiqarish.

rt tegi-`<ruby>` tegi ichiga joylashtirilgan annotatsiyani yuqori va pastdan qo'yish.

ruby tegi-belgilangan matndan yuqorida yoki pastdan uncha katta bo'lmagan annotatsiyani qo'shish uchun mo'ljallangan.

section tegi-hujjatning qismini ko'rsatadi, yangiliklar bloki, qayta aloqa, matnning bo'linishlari uchun va h.k.

source tegi-`<audio>` va `<video>`teglari uchun tovushli yoki video fayl qo'yish.

summary tegi- `<details>` tegi uchun sarlavhalar ko'rsatish uning ustiga bosilganda axborot ochilib yopiladi.

time tegi-belgilangan matnni teg ichiga sana yoki vaqtni birdaniga ko'rsatadi.

video tegi-web sahifaga videolarni qo'shish, ko'rsatish va boshqarishlarni sozlash.

wbr tegi-Agar yuqorida turuvchi elementlarni talab qilsa matnning qayerida satrdan-satrga o'tish belgisini qo'yish kerakligini ko'rsatadi.

6.2. HTML5 standartlari yordamida takomillashgan forma interfeysini ishlab chiqish

Web saytlarda formalar juda ko'p joyda ishlatiladi. Misol uchun, registratsiyada, so'rovlarda, avtorizatsiyada, kommentariyalarda va hokazo.

```
<html>
<head> <title>Form of Form</title><head>
<body>
<form name="forma" action="index.php" method="get">
<table border=1 bgcolor= "blue" align="center">
<tr>
```

```

<td align="left">
  <input type="text" name="search" size="80" value="Qidirish
tizimi...">
  <input type="submit" value="Qidirish">
  <br>
  <input type="checkbox" name="site">Saytdan qidirish
  <input type="checkbox" name="google">Googledan qidirish
</td>
</tr>
</table>
</form>
</body>
</html>

```

Bu kodlarda quyidagi teglar ishlatilgan:

1. `<form>...</form>` - web sahifaga forma joylashtirish, forma foydalanuvchi va web server bilan aloqani ta'minlab beradi. Agar forma elementlari (button, checkbox, radiobutton,...) kiritilmasa, bu teg brauzerga hech nima chiqarmaydi, uning atributlari:

`accept-charset` - serverga jo'natilayotgan axborot kodirovkasini o'rnatish.

```
<form accept-charset= "windows-1251">...</form>
```

`action` - forma orqali jo'natilayotgan ma'lumotlar qaysi sahifa orqali qayta ishlanishi kerakligi beriladi, biror manzil ko'rsatiladi.

```
<form action=«index.php»>...</form>
```

formadagi barcha axborotlar «index.php» sahifasiga olib o'tilib, u yerda tahrirlanadi.

`autocomplete` - formani avtomat to'ldirishni boshqarish, brauzer keshida qandaydir ma'lumotlar bo'lsa, shular orqali forma to'latilishi mumkin, uning qiymati: on (yopiq), off (o'chiq).

`method` - axborotlarni jo'natish turi.

`name` - formaga nom berish, bu nom skriptlar yozishda ishlatiladi.

`novalidate` - formaga kiritilgan ma'lumotlarni brauzer tomonidan tekshirish. Brauzer pochta manzillarni, url manzillarni tekshirishi mumkin.

`target` - jo'natilayotgan axborotni qayta ishlovchi web sahifani ochilish turi, uning qiymatlari: `_blank`, `_self`, `_parent`, `_top`. «a» tegidagi «target» atributi bilan bir hil.

2. `<input>...</input>` - bu teg orqali forma elementlarini yaratish mumkin, ya'ni formani asosi hisoblanadi. Bu elementlar yuqoridagi kodda keltirilgan.

Bu tegning atributlari:

`type` - forma elementi turi, bu atribut formada har xil ko'rinishdagi elementlar hosil qilib beradi, ularni ba'zilarini misollarda ko'rish mumkin, ularning qiymatlari quyidagilar:

`button` – tugma (knopka) hosil qilish.

`checkbox` - bir necha tanlovni amalga oshirib beruvchi bayroqcha hosil qilish.

`file` - serverga jo'natiladigan fayl manzilini kiritish, fayl elementi.

`hidden` - ko'rinmas element, sahifada ko'rinmaydi, lekin ko'p hollarda ishlatiladi.

`image` - rasmi qator, rasmga bosilganda, axborotlar serverga jo'natiladi.

`password` - oddiy matnli element, lekin yozuvlari ko'rinmaydi, parol yozish uchun.

`radio` - bir necha tanlovalardan faqat bittasini tanlash imkonini beruvchi element.

`reset` - forma qiymatlarini dastlabki holatga qaytarish.

`submit` - formadagi axborotlarni serverga jo'natish.

`text` - oddiy matn kiritish elementi.

`align` - tip rasm (`image`) bo'lganda, rasmni tekislash, qiymatlari: `bottom`, `left`, `middle`, `right`, `top`.

`alt` - rasmi tugmada, rasm ochilmasa, uning o'rnida paydo bo'ladigan yozuv.

`autocomplate` - elementni avtomat to'ldirishni o'rnatish.

`border` - rasm atrofidagi ramka qalinligini o'rnatish.

`checked` - `checkbox` yoki `radio` uchun dastlabki holatni «belgilash» ko'rinishida namoyon qilish.

`<input type=«checkbox»> Messi`

`<input type=«checkbox» checked=«checked»> Rooney`

`<input type=«checkbox» > Neymar`

`disabled` - elementni blokirovka qilish, ya'ni dastlabki holatni o'zgartirib bo'lmaydi.

`<input type=«text» disabled value=«Yozolmaysan»>`

`form` - boshqa forma bilan identifikator orqali bog'lanish.

`formaction` - axborotlarni qayta ishlaydigan sahifa adresini berish (form tegi `action` atributi bilan bir hil).

formmethod - axborotlarni qaysi metod orqali serverga jo'natishni belgilash (Get, Post. Form tegi method atributi bilan bir hil).

maxlength - foydalanuvchi tomonidan maksimal kiritilishi mumkin bo'lgan axborot o'lchami (kiritilgan raqamgacha bo'lgan simvollar).

smultiple - element turi fayl bo'lganda, bir paytda bir necha fayllar kiritilishi mumkinligini bildiradi.

name - forma elementiga takrorlanmas nom berish, bu nom axborot serverga jo'natilayotganda ishlatiladi.

readonly - yozuvli elementda ishlatilsa, faqat yozuvni ko'rish imkonini yaratadi, o'zgartirib bo'lmaydi.

size - yozuvli elementlarda (text, password) yozuv yozish mumkin bo'lgan joyning uzunligini o'rnatish.

src - tur rasm (type = image) bo'lgan elementni adresini ko'rsatish.

tabindex - klaviaturadan nechi marotaba «tab» tugmasi bosilsa, shu elementga o'tish kerakligini o'rnatish. Qiymat raqamlardan iborat bo'ladi.

value - forma elementiga dastlabki qiymatni berish.

1. <fieldset>...</fieldset> - formadagi bir necha elementlarni guruhlash uchun ishlatiladi. Guruhlangan elementlar chiziq bilan chegaralanadi. Uning atributlari:

disabled - guruhning barcha elementlarini bloklash.

form - boshqa formalar bilan bog'lanish (identifikator orqali).

title - formadagi guruhga yordam sifatida yozuv chiqarish.

<fieldset>

<input type=«text»> Name
<input type=«Password» value=«1234»> Parol

<input type=«checkbox» checked=«checked»> Belgilash

</fieldset>

2. <legend>...</legend> - «fieldset» orqali hosil qilingan guruhga nom berish, bu nom chizilgan chegara chizig'ining tepa qismida paydo bo'ladi (misolda «shaxsiy ma'lumotlar» degan nom).

Atributlari quyidagilar:

accesskey - klaviatura tugmalari orqali tez o'tishni amalga oshiradi.

align - guruh nomini tekislash: left, right, center.

title - forma nomiga sichqoncha oborilsa, yordam sifatida yozuv chiqarish.

<fieldset title=«guruh»><legend align=«center»>Umumiy guruh</legend>

<input type=«text»> Name

<input type=«Password» value=«1234»> Parol

</fieldset>

3. <select>...</select> — foydalanuvchi tanlashi uchun bir necha variantlarga ega bo'lgan ro'yxat hosil qilish. Bu teg «option» tegi bilan birga ishlatiladi. Bir yoki bir nechta variantni tanlash imkonini beradi (size atributi orqali o'rnatiladi). Uning atributlari:

accesskey - klaviatura tugmalari orqali tez o'tishni amalga oshiradi.

disabled - o'zgartirib bo'lmaydigan holga olib keladi, ya'ni blokirovka.

form - boshqa forma bilan identifikator orqali bog'lanishni amalga oshirib beradi.

multiple - bir vaqtning o'zida bir necha qiymatni tanlash imkonini beradi.

name - serverga jo'natish uchun element nomini o'rnatish.

size - qiymatlarni nechki qator bo'lib ekranga chiqarishini o'rnatish, qolgan qiymatlar ko'rinmas holda bo'lib turadi va menyu ochilsa ko'rinadi.

4. <option>...</option> — «select» orqali hosil qilingan ro'yxatdagi elementlarni yaratish.

Nechta «option» bo'lsa, ro'yxatda shuncha element bo'ladi. Uning atributlari:

disabled - ma'lum bir ro'yxat elementlarini bloklash.

label - ro'yxat elementiga metka qo'yish.

selected - brauzer yuklanganda shu atribut o'rnatilgan element tanlangan bo'lib chiqadi.

value - serverga shu nom orqali element qiymati ketadi.

```
<select name="club" size="1">
```

```
<option value="mufc">ManchesterUnited FC
```

```
<option value="fcb" selected>Barcelona FC
```

```
<option value="pfc" disabled>Pakhtakor FC
```

```
</select>
```

5. <textarea>...</textarea> - forma elementi bo'lib, katta hajmdagi matnlarni kiritish uchun ishlatiladi. Uning atributlari:

accesskey - klaviatura tugmalari orqali tez o'tishni amalga oshiradi.

cols - matn yozish uchun ajratilgan joy uzunligi (raqamda beriladi, nechta simvol kiritilishi).

disabled - elementni blokirovka qilish.

form - boshqa forma bilan identifikator orqali bog'lanishni amalga oshirib beradi.

maxlength - foydalanuvchi tomonidan kiritilishi mumkin bo'lgan maksimal simvollar soni.

name - elementning nomi.

readonly - element ichidagi yozuvni faqat o'qish mumkin bo'lgan holga keltirish (o'zgartirib bo'lmaydi).

rows - matn yozish uchun ajratilgan joy balandligi (qatorlar soni), shu sonidan so'ng pastga tushish uchun aylantirgich hosil bo'ladi (prokrutka).

tabindex - klaviaturadan nechinchi marotaba «tab» tugmasi bosilsa, shu elementga o'tishni.

wrap - yozilgan matnni keyingi qatorga o'tkazish qoidasini o'rnatish, uning qiymatlari.

soft - matn uzunligi bitta qatorga sig'masa, avtomat keyingi qatorga o'tadi.

hard - «cols» da berilgan qiymatdan so'ng uzun matnlar keyingi qatorga o'tadi.

off - matn uzunligi tugamaguncha birinchi qator ham tugamaydi (surilgich chiqadi).

```
<textarea rows= "3" cols= "50" readonly wrap= "off">  
</textarea>
```

6.3. Web sahifaga audio, video va vektor grafikalarini qo'shish

HTML 5 audio kontentlar bilan ishlash uchun keng imkoniyatlar beradi. HTML 5 audio kontentlarni quyidagi brauzerlar versiyalarida ishga tushirish mumkin.

1. IE: 9.0
2. Firefox: 3.5 dan boshlangan, 15.0 - to'liq
3. Chrome: 3.0
4. Safari: 3.1
5. Opera: 10.5
6. iOS Safari: 7.1
7. Opera Mini: -
8. Android Browser: 4.1
9. Chrome for Android: 44

Ayni paytda barcha brauzerlarda birdek ishlaydigan audio format mavjud emas. Shuning uchun imkon qadar, maksimal ishlashni

tashkilashtirish uchun bir necha formatlardan foydalaniladi. Bularga mp3 va ogg formatlari misol bo'la oladi (jadval ko'rinishida).

```
<!DOCTYPE html>
<html>
<head>
<title>Adding HTML5 Audio</title>
</head>
<br>
<body>
<h1 align="center"><font color="blue">MP3 Musics </font>
</h1>
<table cellpadding="5" cellspacing="1" align="center"
border="0"><tr><th bgcolor=""><h2>Sardor_Rahimxon -
Onam</h2></th><th bgcolor=""><h2>Talabalik
yillarim</h2></th></tr>
<tr bgcolor="pink">
<td><audio controls play><source
src="D:/MUSIQA(g'aroyib)/Sardor_Rahimxon - Onam.mp3">
</audio></td>
<td><audio controls play><source
src="D:/MUSIQA(g'aroyib)/talabalik yillarim.mp3"> </audio></td>
</tr>
</table>
<br>
</body>
```

MP3 Musics

Sardor_Rahimxon - Onam

Talabalik yillarim



6.1-rasm. HTML5 da audio tegidan foydalanish

CSS uslublar va JQuery qo'shimchasi bilan audio playerni o'zgacha ko'rinishga keltirish mumkin. Agarda audio playerning ustiga sichqoncha olib borilsa, audio player ishga tushadi.

HTML 5 texnologiyasida video fayllar bilan ishlash. HTML 5 texnologiyasi video va multimedia fayllari bilan ishlashda o'ziga hos dasturiy interfeys va yangi standartlarni taqdim etadi.

<Video> elementi bilan web sahifaga video fayllarni qo'shish va CSS uslublari yordamida video pleyerni tashqi ko'rinishiga turli hil uslublarni berish imkonini paydo bo'ldi.

Video faylning sahifaga qo'yish uchun quyidagi teglardan foydalaniladi.

```
<h1>Video</h1>
<video controls="controls" width="500" height="280" loop="">
  <source src="video.mp4" type="video/mp4">
  <source src="video.ogg" type="video/ogg">
  <source src="video.webm" type="video/webm">
</video>
```

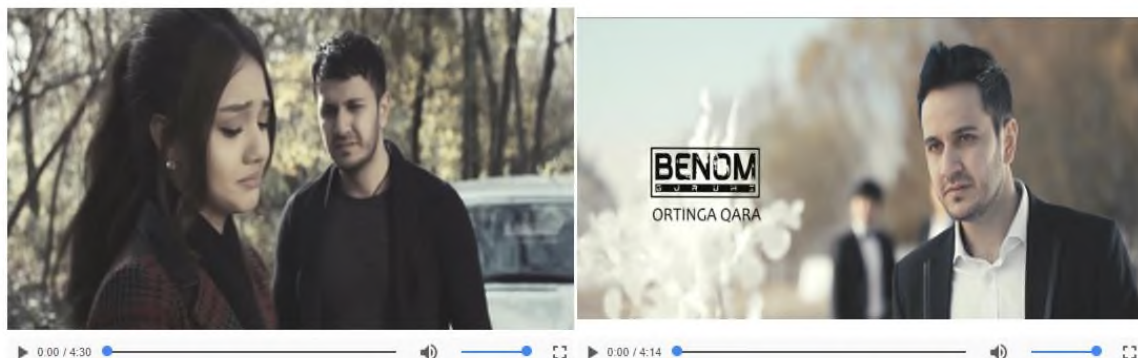
Masalan jadval sifatida web saytga video qo'shishni ko'rib chiqamiz:

```
<!DOCTYPE html>
<html>
<head>
<title>Adding HTML5 Audio</title>
</head>
<body>
<h1 align="center"><font color="blue"> MP4 Videos </font></h1>
<table align="center" width="30" bgcolor="" border="0"><tr><th
bgcolor=""><h2 > Sarvar va Komil - Ayt (1-QISM)</h2></th><th
bgcolor=""><h2>Sarvar va Komil - Ortinga qara (2-
QISM)</h2></th></tr>
<tr bgcolor=""><td><video poster="C:\Users\Acer\Desktop\1.png"
width="600" height="420" controls>
<source src="D:\zagruzki\Sarvar va Komil - Ayt (1-QISM).mp4">
</video></td>
<td><video poster="C:\Users\Acer\Desktop\2.png" width="600"
height="420" controls>
<source src="D:\zagruzki\Sarvar va Komil - Ortinga qara (2-
QISM).mp4">
</video> </td></tr>
</table>
<br>
<br>
</body>
```

MP4 Videos

Sarvar va Komil - Ayt (1-QISM)

Sarvar va Komil - Ortinga qara (2-QISM)



6.2-rasm. HTML5 da video tegidan foydalanish

Video fayillarni imkon qadar barcha brauzerlarda ishlashini ta'minlash uchun uch xil formatda tayyorlab olish kerak. Barcha brauzerlar bitta formatda ishlamaydi. Shuning oldini olish uchun video fayillarni mp4, ogg va web formatlarida tayyorlab olinadi.

HTML5 da vizual effektlarni hosil qilish. Dastlab Apple kompaniyasi ishlab chiqqan vizuallashtiruvchi dastur WebKit (Safari brauzeri) uchun amalga ohirilar edi, endi qo'shimcha modullarsiz ko'pgina iOS, Android, Kindle, Chrome, BlackBerry, Opera i Tizen brauzerlar uchun web sahifaga grafik elementlar chizish imkonini beradi.

Canvas standartlashtirilgandan keyin barcha brauzerlar uchun mashxur web dasturlash vositasiga aylandi. Boshqa HTML-elementlar kabi canvas – bu elementi o'lchami oldindan aniqlangan oddiy web sahifa bo'lib, ichida JavaScript dan foydalanib grafika bilan ishlash mumkin.

Holst <canvas> tegi yordamida hosil qilinadi, uni identifikator bilan aniqlash kerak.

Misol: HTML5 da <canvas> tegidan foydalanish.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<canvas id="myCanvas" width="200" height="100"  
style="border:1px solid #d3d3d3;">
```

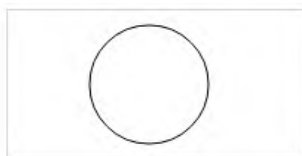
```
Your browser does not support the canvas element.
```

```
</canvas>
```

```
<script>
```

```
var canvas = document.getElementById("myCanvas");
```

```
var ctx = canvas.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
</script>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<body>
<canvas id="myCanvas" width="450" height="100"
style="border:5px solid #d3d3d3;">
Your browser does not support the canvas element.
</canvas>
<script>
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");
ctx.font = "30px Arial";
ctx.strokeText("Hello TUIT of Student Kozimjon",10,60);
</script>
</body>
</html>
```



Hello TUIT of Student Kozimjon

NAZORAT SAVOLLARI

1. HTML5 tilida audio tegi va uning vazifasi?
2. HTML5 tilida video tegi va uning vazifasi?
3. HTML5 da vizual effektlar va canvas tegidan foydalanish?

7-BOB. CSS GA KIRISH. HTMLGA ULANISH USULLARI. SELEKTOR TURLARI. ELEMENTLAR VA SINFLAR. O'LCHOV BIRLIKLARI

7.1. CSS ga kirish. HTML ga ulanish usullari

CSS - bu stillar bilan ishlay oladigan kaskadli stillar majmuasidir. Web sahifalarga har xil stillar berish uchun ishlatiladi. HTML hujjat ichida teglar bilan birga foydalaniladi. CSS kodlarini kompilyasiya qilish uchun qandaydir kompilyatorlar, qandaydir dasturlar, va bu kod tushuna oladigan qandaydir redaktorlar kerak bo'lmaydi. Bu kodlarni HTML singari web brauzerning o'zi kompilyasiya qiladi va natijani chiqarib beradi. Bu stillar jadvali (CSS) to'liq holatda 1997 yilda tashkil topib, "WWWCorsorium" ida qo'llab quvvatlangan va foydalanishga taqdim etilgan. O'sha paytlarda HTML ning 3 versiyasi ommabop edi. CSS ni dastlab Netscape Navigator 4.0 va Internet Explorer 4.0 brauzerlarida ko'rish mumkin bo'lgan. Hozirda barcha brauzerlar CSS da yozilgan kodlarni tushuna oladi. Bu stillarga oid dasturlash tilini o'rganish sodda bo'lib, yangi stillar qo'shilishi natijasida kengayib bormoda.

CSS ni HTML hujjati ichida ishlatilishini birin ketin ko'rib o'tamiz:

1. Web sahifada ishlatiladigan CSS kodlari orqali hosil qilinadigan stillarni hammasini bitta faylga yozib, uni serverda saqlaymiz. So'ng, sahifadan, serverdagi saqlab qo'yilgan faylga murojaat qilib, kerakli stilni olamiz. Bu usul katta hajmdagi web sahifalar yaratishda foydali hisoblanadi. Faylga murojaat, <head> tegi ichida amalga oshiriladi va quyidagi ko'rinishga ega bo'ladi.

<LINK REL=STYLEShEET TYPE="text/css"
HREF="URL"></link> tegi sahifa biror faylga murojaatni amalga oshirmoqda degani, REL – fayl qandaydir stillardan iborat degani, TYPE - stillar CSS kodlaridan iborat va fayl joylashgan manzil yoziladi.

2. CSS kodlar web sahifa ichidagi <head> tegi orasida beriladi. Bunda yaratilgan stillar shu sahifaga tegishli bo'ladi, bu degani faqat shu sahifada ishlatilishi mumkindir.

3. Navbatdagi turi, bu har bir element uchun alohida teg ichida stil berib chiqishdan iborat. Misol uchun, <p> tegi ichida biror matn yozilsa, shu matn uchun teg ichida alohida stil beriladi va bu stil shu teg uchun xususiy bo'ladi. Bu usuldan foydalanishni maslahat bermaymiz, chunki bu usul kodlarni chalkash va tushunarsiz bo'lishiga olib keladi. Ayrim hollarda ya'ni, shoshilinch vaziyatlarda yoki stil qay tartibda

namoyon bo'lishni ko'rib olish uchun bu usuldan foydalanish mumkin. Undan tashari <style> tegi orqali CSS faylni import qilish mumkin bo'ladi.

```
@import: url(mystyles.CSS);
```

CSS stillarini ishlatishda, quyidagi guruhlash usullaridan foydalanish, tartibli web sahifa tuzishga olib keladi:

- kodlarni ixchamlash uchun, turli elementlar uchun bir xil turdagi stillarni guruhlab ishlatish maqsadga muvofiqdir.

```
H1 {font-family: Verdana}
```

```
H2 {font-family: Verdana}
```

Yuqoridagi stilni guruhlab quyidagicha yozamiz.

```
H1, H2 {font-family: Verdana}
```

- Guruhlashda bir xil element uchun bir necha stillarni birlashtirish mumkin.

```
H2 {font-weight: bold}
```

```
H2 {font-size: 14pt}
```

```
H2 {font-family: Verdana}
```

Barcha stillarni jamlaymiz.

```
H2 {font-weight: bold; font-size: 14pt; font-family: Verdana;}
```

- Ba'zi stillar bitta stil ichida berilishi ham mumkin.

```
H2 {font: bold 14pt Verdana}
```

CSS stillari bilan ishlaganda, element ichidagi elementlar ham o'zidan yuqori turgan element stilini qabul qiladi, ya'ni o'zidan oldingi element "ota" vazifasini bajaradi va o'z "farzandlari"ga ham o'zida borini beradi. Misol uchun, <p> tegi ichidagi matn ko'k rangda yozilishi lozim bo'lsa (P {color: blue}), <p> tegi ichidagi tegiga tegishli bo'lgan matn ham ko'k rangda yoziladi. Ba'zi stillar faqat yakka tartibda ishlaydi, ya'ni ichki teglarga stillari o'tmaydi (misol uchun, background), shuning uchun barcha stillarni birma bir ishlatib, o'rganib olish lozim. <body> tegiga stil berib, dastlabki o'rnatilish jarayonini hosil qilish mumkin. Shunda biror elementga stil berish esdan chiqib qolsa, <body> tegidagi stil esdan chiqqan elementga o'rnatiladi.

```
BODY {color: greyen;
```

```
font-family: "Verdana";
```

```
background: url(joke.gif) white; }
```

7.2. CSS kaskadli jadvallar stillarida selektorlar

CSS qoidalar, selektor va stilni e'lon qilish blokidan tashkil topadi. Selektor har doim chap tomonda joylashadi va undan keyin figurali qavsni

ichida stilni e'lon qilish bloki joylashadi. Har bir e'lon, hususiyat va uni qiymatidan tashkil topadi. E'lonlar bir – biridan nuqtali vergul bilan ajratiladi. Aynan hususiyat stilni tashqi ko'rinishini aniqlaydi.

Stil qo'llash mumkin bo'lgan elementning konstruktsiyasini selektor deb ataymiz.

Selektorlarning turlari.

- Element bo'yicha selektor. Selektor sifatida elementning nomidan ham foydalanish mumkin (p, td, body). Masalan quyidagicha konstruktsiya yaratamiz:

```
P { color: #CCC; font-size: 14px }
```

Bu abzats ichidagi matn kulrang rangda, 14 piksel o'lchamli shriftda yozilgani ko'rsatilgan. HTML tilidagi ixtiyoriy elementdan shu tarzda selektor sifatida foydalanish mumkin. Selektorga qo'llanilgan stil, butun sahifa uchun o'rinli bo'ladi. Bazan shunday holat yuzaga keladiki, H1 sarlavha ba'zi joyda ko'k rangli, 18 px o'lchamli bo'lsin, ba'zi joyda esa H1 sarlavha yashil rangda, 16 px o'lchamli bo'lsin. Agar biz selektor sifatida H1 ni olsak, bunday differentsiallashtirishga erisha olmaymiz. Bu muammoni hal qilish uchun CSS1 da ikkita mexanizm, sinfs va unikal identifikatorlar (ID) tushunchasi kiritilgan.

- Sinf bo'yicha selektor. Nomidan ko'rinib turibdiki, bu mexanizm, berilgan elementlarni sinflarga ajratadi. Bu ish quyidagicha amalga oshiriladi: Bu hat boshida biror matn keltirilgan, bu hat boshidagi matn esa kichraytirilgan shrift bilan yoziladi. Ko'rsatilgan sinfsga stillar quyidagi tartibda yoziladi:

```
P { color: #CCC; font-size: 14px }
```

```
P.smaller { font-size: 12px }
```

Bu holda birinchi hat boshi 14 piksel o'lchamli shrift bilan yoziladi. Ikkinchi hat boshi esa 12 piksel o'lchamli shrift bilan yoziladi. Sintaksis ikkala holda ham o'xshash, lekin sinfsga murojaat nuqta orqali amalga oshiriladi, nuqtadan keyin sinfsni nomi yoziladi. Agar ko'rsatilgan sinfsni faqatgina hat boshi uchun emas, balki ixtiyoriy element uchun qo'llasak, stil quyidagicha yoziladi:

```
.smaller { color: #CCC; font-size: 14px }
```

Agar ikkita <P> va <PRE> element va bir xil sinfs bilan berilsa, ularni differentsiallashtirish mumkin:

```
P.smaller { color: #CCC; font-size: 14px }
```

```
PRE.smaller { color: #F00; font-size: 12px }
```

Bu holda <PRE> element bilan ko'rsatilgan matn qizil rangda, shrift 12 piksel boladi. <P> element bilan ko'rsatilgan matn kulrang rangda, shrift 14 piksel o'lchamda bo'ladi.

- ID bo'yicha selektor:

Unikal identifikatorlarni qisqa dilib, ID deb ataymiz. Berilgan hujjatda faqat bir martagina ID selektoridan foydalanish mumkin.

```
#first { font-style:italic; }
```

```
<H1 ID = "first"> Unikal sarlavha</H1>
```

ID unikal sarlavha uchun CSS stil yozganda ajratuvchi sifatida (#) belgisidan foydalaniladi:

```
H1#first {  
color: #00F;  
font-family: Verdana, sans-serif;  
font-weight: bold;  
text-align: center }
```

- KonMatnselektor. Web-sahifalar yaratishda ba'zan, bir tegni ichiga boshqa tegni joylashtirishga extiyoj paydo bo'ladi. Bu holda stildan to'g'ri foydalanish uchun, konMatnselektorlar yordam beradi. Masalan. uchun stil belgilaylik, qachonki u <P> konteynerni ichida joylashgan bo'lsa. Sintaksis quyidagicha bo'ladi:

```
EF { Stil qoidalarini ifodalash }
```

Bu yerda E konteyner teg, F esa - <E> konteynerni ichidagi teg. Bu holda stil <F> tegi uchun qo'llaniladi va quyidagi kodga amal qilinadi <E><F> ...</F></E>. Faqatgina ikkita teg emas, balki undan yuqori qatlamlar ishlatish mumkin. Bu holda konstruktsiya quyidagicha bo'lishi mumkin: div div ul li {...}.

Misol:

```
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html;  
charset=windows-1251">  
<title>KonMatnselektor</title>  
<style type="text/css">  
P B {  
font-family: Times, serif; /* shriftlar oilasi */  
font-weight: bold; /* qalin chiziqli */  
color: red; /* Matn qizil rangda */  
}
```

```

</style>
</head>
<body>
  <div><b>Qalin chiziqli matn</b></div>
  <p><b>Bir vaqtni o'zida ham qalin chiziqli va qizil rangli
matn</b></p>
</body>
</html>

```

- Universal selektor. Ba'zi hollarda web-sahifaning hamma elementlari uchun bir vaqtda bitta stil o'rnatishga to'g'ri keladi. Bu holda universal selektordan foydalaniladi. Sintaksis quyidagicha bo'ladi:

```
*{Stil qoidalarini ifodalash}
```

Universal selektordan foydalanish uchun yulduzcha (*) belgisi ishlatiladi. Bazi hollarda universal selektorni ko'rsatish shart emas. Masalan *.class va .class yozuvlar bir xil natijani chiqaradi. Universal selektordan foydalanish uchun yulduzcha (*) belgisi ishlatiladi. Masalan.

```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
  <title>Universal selektor</title>
  <style type="text/css">
  *{
  margin: 0;
  padding: 0; /*Hamma elementlar uchun hoshiya va otstup nolga teng
*/
  font-family:Arial;
  font-size:14px;
  }
  </style>
</head>
<body>
  <p> Universal selektordan foydalanish uchun yulduzcha (*) belgisi
ishlatiladi. Bazi hollarda universal selektorni ko'rsatish shart emas.
Masalan *.class va .class yozuvlar bir xil natijani chiqaradi.</p>
  </body>
</html>

```

O'lchov birliklari. CSS da hususiyatlarni qiymatini berish uchun turli o'lchov birliklari ishlatiladi. Bular uzunlik o'lchov birligi, ranglarni berish birligi va URL.

Uzunlik birliklari

- em - ems, foydalanilayotgan shriftning balandligi
- ex - x-height, foydalanilayotgan yozma harfning balandligi ("x")
- px - pixels, piksellar (monitor ekranidagi bitta nuqta)
- in - inches, dyumlar (1in=2,54 sm)
- cm - centimeters, santimetrlar
- mm - millimeters, millimetrlar
- pt - points, punktlar (1pt = 1/72in = 0,35mm)
- pc - picas, piklar (1pc = 12pt)

7.1-jadval. CSS kaskadli jadvallar stillarida ranglarning berilishi.

Rang	RGB	Kod	Qisqacha
Oq	255,255,255	#ffffff	#fff
Qora	0,0,0	#000000	#000
Qizil	255,0,0	#ff0000	#f00
Yashil	0,255,0	#00ff00	#0f0
Ko'k	0,0,255	#0000ff	#00f

% - nisbiy qiymat (masalan +20%).

Ranglarni belgilash. CSS da ranglar uch xil usulda beriladi: kalitli so'z orqali (blue, red va hokazo), o'n oltilik sistemadagi kodi va RGB modelidagi kod orqali.

7.3. Pseudosinflar va psevdoelementlar

Pseudosinf selektorlar. Pseudosinf selektorlarni ko'pchilik brauzerlar o'qiydi va shuning uchun amalda keng qo'llaniladi.

CSS 1 da xavola (ssilka) lar bilan ishlash uchun to'rt xil Pseudosinf mavjud:

- link - oddiy xavola (ssilka) ni belgilash uchun ishlatiladi;
- active - aktiv (ko'rib chiqilayotgan) xavola (ssilka) ni belgilash;
- visited - ko'rib chiqib bo'lingan xavola (ssilka) ni belgilash;
- hover - sichqoncha ko'rsatkichi bilan aktivlashtirilgan (ko'rib chiqilmoqchi bo'lgan) xavola (ssilka) ni belgilash.

CSS yordamida xavola (ssilka) larni tashqi ko'rinishini berish uchun quyidagicha qoidalar yozish mumkin:

A: link{ ... oddiy (ko'rib chiqilmagan) xavola (ssilka) ni bezash stili... }

A:active{ ... aktiv xavola (ssilka) ni bezash stili ... }

A:visited{ ...ko'rib chiqilgan xavola (ssilka) ni bezash stili... }

va nihoyat HTML vositasi bilan bajarish mumkin bo'lmagan ish, sichqoncha ko'rsatkichi bilan aktivlashtirilgan xavola (ssilka) ni berish.

A:hover{ ... ko'rib chiqilmoqchi bo'lgan xavola (ssilka) ni bezash stili... }

Sinfs selektori orqali xavola (ssilka) yaratsak, u quyidagi ko'rinishda bo'ladi:

a.visited {color: red;}

W3C Web site

Xuddi shu ishni Pseudosinf selektori orqali ko'rsatamiz:

a:visited {color: red;}

Bu holda ko'rib chiqilgan har qanday xavola (ssilka) qizil rangda bo'ladi, va unga sinfs atributini qo'shish shart emas .a va visited ni ajratuvchi ikki nuqta (:) aynan Pseudosinfdan foydalanilayotganini bildiradi. Pseudosinf va psevdoelementni hamma kalit so'zlari ikki niqtadan boshlanadi.

Pseudosinflardan, oddiy sinf va identifikatorlar bilan birgalikda turli kombinatsiya tuzib ham foydalanish mumkin.

... tekst...

A.pagelink:hover { ... stilni ifodalash ... }

Yoki identifikator bilan birgalikda foydalanish:

...tekst...

A#mylink:visited { ...stilni ifodalash ... }

Ko'pchilik web sahifalardagi stil quyidagi ko'rinishda bo'ladi:

a:link {color: navy;}	}	Bu ikkita qoidada statik pseudosinfdan foydalanilgan
a:visited		
{color:gray;}		
a:hover {color: red;}	}	Bu ikkita qoidada dinamik pseudosinfdan foydalanilgan
a:active		
{color:yellow;}		

Dinamik Pseudosinflar foydalanuvchining ta'sirida hujjatning tashqi ko'rinishini o'zgartirish imkonini beradi. CSS2.1. uchta dinamik Pseudosinfni aniqlaydi:

7.2-jadval. Dinamik psevdosinflar turi.

Nomi	Tushuntirish
:focus	Joriy vaqtda diqqat-e'tiborni tortayotgan har qanday elementga qo'llash mumkin (klaviaturadan axborot kiritish jarayoni)
:hover	Sichqoncha ko'rsatkichi bilan ustiga ta'sir etilayotgan har qanday elementga qo'llash mumkin
:active	Foydalanuvchi tomonidan aktivlashtirilgan har qanday elementga qo'llash mumkin

Misol:

```
input:focus {background: silver; font-weight: bold;}
```

Bu misol formaga axborot kiritish jarayonida, axborot kiritilayotgan oynaning fon rangi va shriftni qalinligi o'zgarishi effektini beradi:

Misol:

```
body *:hover {background: yellow;}
```

Bu qoidaga ko'ra, body bo'limida joylashgan har qanday elementlar, sichqoncha ko'rsatkichini ustiga qoysak, sariq fonli rangni oladi.

Agar hujjatning tiliga e'tibor bermoqchi bolsak, :lang() Psevdosinfiga murojaat qilamiz. Masalan frantsuz tilidagi ixtiyoriy elementni og'ma shriftida ko'rsatish uchun quyidagicha qoida yozamiz:

```
*:lang(fr) {font-style: italic;}
```

CSS2.1 da Psevdosinflarni kombinatsiyasidan foydalanish mumkin. Masalan hemis tilida yozilgan hamda ko'rib chiqilmagan va ko'rib chiqilgan xavola (ssilka) larni ifodasi quyidagi ko'rinishda bo'ladi:

```
a:link:hover:lang(de) {color: gray;}
```

```
a:visited:hover:lang(de) {color: silver;}
```

Psevdoelement. HTML tilida hat boshining birinchi qatori yoki birinchi harfini ifodalash uchun element yo'q. Demak bu ishni bajarish uchun bezash stilini qo'llab bo'lmaydi. Shuning uchun ham CSS da psevdoelement tushunchasi kiritilgan.

CSS2.1 da to'rtta psevdoelement mavjud:

Birinchi harf – :first-letter

Birinchi qator - :first-line

Elementdan oldin - :before

Elementdan keyin - :after

Sinfs yordamida katta harf yaratish usulini ko'ramiz:

```
<P><SPAN class="firstletter">K</SPAN>Katta harf yaratish</P>
```

stil qoidalarini yozamiz:

SPAN.firstletter { color: blue; font: bold large Verdana, sansserif }

Bu holda har bir hat boshini birinchi harfini

..., teglar orasiga yozish kerak.

Bu ish noqulaylik yaratadi.

Pseudosinfidan foydalansak ish ancha soddalashadi, quyidagicha kod yozish yetarli:

<P> Psevdoelement first-letter yordamida katta harf yaratamiz</P>

stil yozish qoidasi:

P:first-letter{ color: blue; font: bold large Verdana, sansserif }

Agar matn birinchi harfdan ajralgan holda aks etsin desak, qo'shimcha hususiyat float:left beramiz. U holda katta harf chap tomonga tekislanadi, Matn katta harfdan ajralib, uni atrofida joylashadi.

P:firstletter{ color: blue; font: bold 24pt Verdana. sansserif; float:left }

p:first-letter {color: red;} – hat boshining birinchi harfi qizil rangda aks etadi.

h2:first-letter {font-size: 200%;} - sarlavhaning birinchi harfi ikki baravar kattalashgan holda aks etadi.

Matnning birinchi qatorini, boshqa qatorlardan farqli tarzda ifodalash uchun quyidagicha qoida yozamiz:

P:first-line{ font-style: italic }

bu holda ixtiyoriy hat boshining birinchi qatoridagi matn kursiv shrift bilan aks etadi.

p:first-line {color: purple;} – hat boshining birinchi qatori siyoh rangda aks etadi.

:first-letter va :first-line psevdoelementlarni faqatgina blok darajasidagi elementlarga qo'llash mumkin. Psevdoelementlarda qo'llash mumkin bo'lgan hususiyatlarni quyidagi jadvalda keltiramiz:

7.3-jadval. Psevdoelementlarda qo'llash mumkin bo'lgan hususiyatlar.

:first-letter	:first-line
Hamma hususiyatlar:	Hamma hususiyatlar:
font	font
color	color
background	background
margin	word-spacing
padding	letter-spacing
border	vertical-align

text-decoration text-transform line-height float	text-decoration text-transform line-height
---	--

Elementdan oldin qo'yiladigan maxsus stilni qo'llash.

Masalan har bir h2 elementning oldiga ikkita qizil kvadrat qavs qo'yish kerak bo'lsin:

```
h2:before {content: "[ ]"; color: red;}
```

Elementdan keyin qo'yiladigan maxsus stilni qo'llash.

Masalan hujjatni yakunlashda quyidagicha qoidadan foydalanish mumkin:

```
body:after {content: " The End.";}
```

Psevdoelementlardan foydalanishga misollar keltiramiz:

```
<html>
<head>
  <meta http-equiv="content-type" content="text/html;
charset=windows-1251">
```

```
  <title>Harfning rangi</title>
```

```
  <style type="text/css">
```

```
    P:first-letter { color: red; /* Har bir hat boshining birinchi harfi qizil
rangda */ }
```

```
    P:first-line{ font-style: italic }
```

```
    h2:first-letter{ color:blue;
```

```
    font:30pt; font-family:Verdana. san-serif;}
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
<h2>Hikmatlar</h2>
```

```
<p>Abu Rayhon Beruniy hikmatlari</p>
```

```
<p>Har bir insonning qadr-qiymati o'z ishini qoyil qilib
bajarishida.</p>
```

```
</body>
```

```
</html>
```

2 - misol

```
<head>
```

```
  <meta http-equiv="content-type" content="text/html;
charset=windows-1251">
```

```
  <title>sarlavha ostidagi fon rangi</title>
```



```
<style type="text/css">
H2 {
background: #D9FFAD; /* sarlavha ostidagi fon */}
p:first-letter{font-size:200% }
</style>
</head>
<body>
<h2>Beruniy hikmatlari</h2>
<p>Rozilik ko'zi hech bir aybni ko'rmaydi,</p>
<p>Lekin norozilik ko'zi barcha ayblarni oshkor qiladi...</p>
</body>
</html>
```

NAZORAT SAVOLLARI

1. HTML hujjatga CSS stillar majmuasini ulash va uning usullari?
2. CSS stillar majmuasida selektorlar va ularning vazifasi?
3. Psevdo-sinf nima va uning qanday turlari mavjud?
4. Psevdo-element nima va uning qanday turlari mavjud?

8-BOB. CSS HUSUSIYATLARI. CSS SAHIFALASH

8.1. CSS hususiyatlari

Color. Bu hususiyat elementning ichidagi Matni rangini berish vositasidir. Bu hususiyatning qiymati uch xil ko'rinishdan birida beriladi:

1. kalit so'zlar bo'yicha (red, yellow, silver,...);
2. o'n oltilik kod bo'yicha (#ff0099, #006688,...);
3. rgb modelidagi o'nlik kod bo'yicha (color:rgb(25, 205, 172)).

Background-color. Bu hususiyatdan web sahifaning yoki ixtiyoriy elementning fon rangini berishda foydalanlladi.

Background – image. Bu hususiyatdan web sahifaning yoki ixtiyoriy elementning fon rasmini berishda foydalanlladi. Rasmi fayl aloxida papkada saqlanadi.

Background-repeat. Bu hususiyat sahifaning fon rasmini takrorlanish stilini beradi. U quyidagi qiymatlardan birini qabul qiladi:

1. repeat - tasvir vertikal va gorizontal yo'nalish bo'yicha takrorlanadi va butun sahifani qoplaydi.
2. repeat-x - tasvir gorizontal yo'nalish bo'yicha bir qator bo'lib takrorlanadi.
3. repeat-y - tasvir vertikal yo'nalish bo'yicha bir ustun bo'lib takrorlanadi.
4. no-repeat - tasvir takrorlanmaydi.

Misol :

```
body {  
background-image:url (images/1.gif);  
background-repeat:no-repeat;  
background-attachment:fixed;  
background-position:50%,50%.}
```

bu misolda fon rasmining nomi, "1.gif", u images papkasida joylashgan, birinchi holat bo'yicha fon rasmi takrorlanmaydi, ikkinchi holatda fon rasmi matn bilan birga harakatlanmaydi, uchinchi holatda esa fon rasmi ekranning o'rtasiga joylashgan.

CSS yordamida fon rasmini, ekranning ixtiyoriy joyiga qo'yish, vertikal va gorizontal yo'nalish bo'yicha takrorlanishini ta'minlash mumkin.

Shrift hususiyati. CSS da shriftlarni boshqarish imkoniyati HTML ga qaraganda keng. Matnli bloklar uchun CSS qoidalaridan foydalanish HTML kodni hajmini anchagina kamaytiradi.

font-family hususiyati - shriftni garniturasini berish uchun ushlatiladi. Qabul qilishi mumkin bo'lgan qiymatlar:

- Serif
- Sans-serif
- Monospace
- Cursive
- Fantasy dekorativ shriftlar.

8.1-jadval. Shriftning hususiyatlari.

Hususiyat	Tushuntirish	Misol
Font-family	Shriftlar oilasini belgilash	<code>p{ fontfamily:Helvetica,sansserif; }</code>
Font-style	Kursiv yoki og'ma shriftni o'rnatadi <code>italic oblique normal</code>	<code>em{ font-style:normal; }</code>
Font-variant	Registrni o'rnatadi <code>smal-caps normal</code>	<code>h3{ font-variant:smallcaps; }</code>
Font-weight	Shriftni qalinlik darajasini belgilaydi. <code>Bold normal 100 ...900</code>	<code>B{ font-weight:700; }</code>
Font -size	Shriftni o'lchamini o'rnatadi,bunda absolyut, nisbiy o'lchamlar,% ,xizmatchi so'zlar ishlatilishi mumkin. <code>Xx-smal, x-smal smal medium large x-large xx-large</code>	<code>h2 { font-size:200%; }</code> <code>h3 { font-size:36 pt; }</code>

Matn va uning hususiyatlari. Matnni hususiyati so'zlar va jumalarning o'zari joylashuvini aniqlaydi. Ya'ni matn protsessori bilan taqqoslasak, hat boshilar va uni ichini formatlash jarayonidagi ishlar bajariladi.

Text – decoration. Bu quyidagi qiymatlardan birini qabul qilishi mumkin:

- none – qo'shimcha bezash ishlatilmaydi;
- underline - matnni har bir qatorini tagiga chiziladi;
- overline - matnni har bir qatorini ustiga chiziladi;
- line-through - matnni har bir qatori chizib tashlangan holda aks etadi;
- blink - ekranda yonib, o'chib turadigan matn aks etadi.

Text-align. Bu hususiyat matnning tekislash usulini o'rnatish imkonini beradi. Mumkin bo'lgan qiymatlari:

left - chap hoshiyaga tekislash;

center – o'rtaga tekislash;

right – o'ng hoshiyaga tekislash;

justify - butun sahifa bo'ylab tekislash.

Text-indent. Bu hususiyat orqali hat boshining birinchi qatoridagi otstup beriladi. Matnli blokni ko'rsatishda ishlatiladi. Birinchi galda hat boshi va sarlavhalar uchun qo'llaniladi.

Text-transform. Bu hususiyat matnni registrini almashtirishda qo'llaniladi. Quyidagi qiymatlardan birini qabul qiladi:

capitalize - har bir so'zning birinchi harfini registri o'zgaradi, ya'ni bosh harfda yoziladi.

uppercase - hamma harflarning registri bosh harfga almashadi.

lowercase - hamma harflarning registri yozma harfga almashadi

none - harflarning registri almashmaydi.

HTML va CSS da bloklar. HTML ning hamma elementlarini ikkita turga bo'lish mumkin: blok va qator turlari.

Blok turidagi elementlar mustaqil ko'rgazmali strukturaga ega bo'ladi. Blok turiga misol sifatida h1-h6, p, div kabi elementlarni olish mumkin. Qator turidagi elementlar, chiziqli qator sifatida ko'rinadi. Masalan: I, B, U, S, EM va boshqalar. Ikkita blokli element bitta qatorda joylasha olmaydi.

CSS da hujjatni formatlashda ham blok tushunchasidan foydalaniladi. CSS da bloklar to'g'ri to'rtburchak ko'rinishiga ega bo'lgan mustaqil strukturaviy birlik hisoblanadi. CSSda bloklar struktura jixatidan olganda, bir-biridan ajratilgan. Qator ko'rinishidagi bloklar, strukturali blokni ichida joylashadi. Har bir qator ko'rinishidagi blok, axborotli qismga ega bo'ladi. bular matn, rasm, yoki boshqa axborot bo'lishi mumkin. Blokning bu qismi content yoki hujjatning mazmuni deyiladi. Kontentning atrofidagi bo'sh soha hoshiya (padding) deyiladi. Dizayn nuqta'iy nazaridan hoshiya kontent uchun estetik ko'rinish beradi. Hoshiya esa chegara chizig'i (border) bilan qoplanadi. Bu chiziq turli stil va qalinlikga ega bo'ladi. Chegara chiziq atrofidagi qo'himcha bo'h joy (margin) bo'lishi mumkin. CSS spetsifikatsiyasiga ko'ra, padding, border, margin, blokni kengligiga kirmaydi. Shunday qilib, blokni kengligini berganda, uni axborotli qismi nazarda tutiladi.

Hamma hoshiya, chegaralar va hat boshilarga mos ravishda yuqori, quyi, chap va o'ng bo'laklarga bo'linadi. Har bir bo'lakga aloxida qiymat berish mumkin.

Chegara chiziqlari. Blokning chegara chiziqlari ham to'rtta segmentga bo'linadi. Har bir aloxida segment o'zining xarakteristikasiga ega: chiziqning rangi, qalinligi, turi.

Border-top-width, border-right-width, border-bottom-width, border-left-width hususiyatlar chiziqning qalinligini aniqlaydi.

Qiymatlar quyidagi xizmatchi so'zlar orqali berilishi ham mumkin yoki uzunlik o'lchov birliklarida beriladi.

- thin - ingichka chizikli chegara;
- medium – o'rtacha qalinlikdagi chegara;
- thick - qalin chizikli chegara.

Blok atrofidagi hamma chiziqlarni qalinligi bir xil bo'lsa:

```
P {  
border width: 1px }
```

Bu hat boshi atrofi 1 px qalinlikdagi ramka bilan ajratilgan.

```
DIV {  
border width: thin mediun }
```

Bu misolda DIV konteynerning yuqori va quyi chegarasi ingichka (thin) va chap va o'ng chegarasi o'rtacha (medium) qalinlikdagi chiziq bilan qoplanadi.

Border-color. Border-color hususiyati chegara chiziqning rangini o'rnatadi:

- border-top-color - yuqori segmentning rangi;
- border-right-color - o'ng segmentning rangi;
- border-bottom-color - quyi segmentning rangi;
- border-left-color - chap segmentning rangi.

Bu hususiyatning qiymatlari quyida ko'rsatilgan uch xil usuldan biri orqali berilishi mumkin:

- xizmatchi so'zlar (red, yellow, silver);
- o'n oltilik kod (#00FFCC, #EEDDAA);
- RGB modelidagi o'nlik kod (color: rgb (25, 205, 172)).

Border-style. Bu hususiyat chiziqning turini o'rnatadi. Agar to'rt tomondagi ramka uchun individual stil bermoqchi bo'lsak:

- border top style - yuqori chegaradagi chiziqning stili;
- border right style – o'ng chegaradagi chiziqning stili;
- border bottom style - quyi chegaradagi chiziqning stili;
- border left style - chap chegaradagi chiziqning stili.

Bu hususiyatlar quyidagi qiymatlarning birini qabul qiladi.

- none - chiziqning qalinligi nolga teng.
- Hidden - chiziq ekranda aks etmaydi.
- Dotted - chegara chizig'i nuqtalardan iborat;
- Dashed - chegara chizig'i punktir chiziq.
- Solid - chegara chizig'i uzluksiz chiziq .
- Double - chegara chizig'i ikkilangan uzluksiz chiziq.

Hat boshi (padding). Hat boshilar blokning mazmunini chegara chiziqlardan ajratish imkonini beradi.

Bu hususiyatni to'rt tomondan berish quyidagicha tarzda beriladi.

- padding top
- padding right
- padding bottom
- padding left

Bu hususiyatning qiymatlarini ixtiyoriy uzunlik o'lchov birligi va protsentlarda berish mumkin.

Misol :

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
  <title>margin-bottom</title>
  <style type="text/css">
.layer1 {
background: #008B66; /* fon rangi */
color: white; /* Matnni rangi */
padding: 10px; /* Matn atrofidagi hoshiya */
margin-bottom: -7px; /* quyi hoshiya */
}
.layer2 {
margin-left: 60px; /* chap hoshiya */
background: #ccc; /* fon rangi */
padding: 20px; /* Matn atrofidagi hoshiya */
font:large ; border-bottom-style:dashed; border-bottom-
color:#FF0000;
border-bottom-width:thick; text-align:justify;
}
</style>
</head>
```

```
<body>
  <div class="layer1">
    <h2>Abu Rayhon Beruniy hikmatlari</h2>
  </div>
  <div class="layer2">
    Inson hayotidagi zarurat, ehtiyojlar ularning ilmlarga
    bo'lgan talablarini keltirib chiqardi. Shu ehtiyojlarga ko'ra
    ilmlar har xil tarmoqlarga bo'lindi. Ilmning foydasi ochko'zlik
    bilan oltin-kumush to'plash uchun bo'lmay, balki u orqali inson
    uchun zarur narsalarga ega bo'lishdir.
  </div> </body>
</html>
```

8.2. Bloklarni joylashtirish usullari

Blok yaratilib, uning hususiyati berilgach, brauzer oynasida joylashish o'rnini aniqlash kerak. Umumiy holda blokning o'rnini bir nechta parametrlarga bog'liq. Asosiy rolni esa blokni joylashtirish sxemasi o'ynaydi. Blokni joylashtirish sxemasining to'rt turi mavjud.

1. Normal oqim
2. Nisbiy pozitsiya
3. Absolyut pozitsiya
4. Suzuvchi bloklar modeli

Bundan tashqari hujjatning kodida blokning joylashgan o'rnini ham muhim rol o'ynaydi. Monitor ekranining o'lchami va ko'rsatish qobiliyati ham ahamiyatli. Ko'pchilik web dizaynerlar minimal ko'rsatish qobiliyatni 800x600 deb olib, ishlarini shunga moslaydilar.

Normal oqim. Normal oqim bo'yicha blokni joylashtirish sxemasi oddiy. Hamma struktura yoki qator turidagi bloklar vertikal yo'nalish bo'yicha yuqoridan pastga qarab ketma-ket joylashaveradi va shu tarzda monitor ekranida aks etadi. Bunda ikkita strukturali blok hech qachon bir gorizontall chiziqda joylasha olmaydi. Qator turli bloklarni brauzer gorizontall yo'nalish bo'yicha ketma-ket joylashtiradi. Bunday qator chizikli blok deyiladi, va bir nechta qator turli bloklar bitta chizikli blokda joylashishi mumkin. Agar joy yetmasa vertikal bo'yicha keyingi qatorga o'tadi. Normal oqim o'z hususiyatiga ko'ra HTML kod qanday tartibda yozilsa, shi tartibda ish bajaradi. Agar jadvaldan foydalanilmasa, normal oqim yordamida, murakkab sahifalar tayyorlash mumkin emas.

Kengligi 80%, qalinligi 1 px va yashil ramka chiziq bilan o'ralgan, hamda sahifaning o'rtasida joylashadigan asosiy blok yaratamiz.

```
#main {  
border:1px green solid;  
margin:0 auto;  
width:80%; }
```

HTML kodida esa quyidagicha konstruktsiya yoziladi:

```
<body>  
<div id= "main">  
</div>  
</body>
```

Hamma qolgan bloklar asosiy blok "main" ning ichiga joylashadi. Birinchi shunday blok sarlavha ya'ni saytning nomi bo'ladi. Bu blokni head, deb nomlaymiz. Bu yerdagi matn kattaroq o'lchamli shriftda yoziladi. Bundan tashqari sarlavha uchun to'q kulrang fon, matnning rangini oq qilib belgilaymiz. Sarlavhani o'rtaga joylashtiramiz.

```
#head {  
background:#A5A5A5;  
color: #FFF;  
font size: 2,5em;  
text-align: center; }
```

Endi sahifaning HTML kodi quyidagi ko'rinishni oladi:

```
<body>  
<div id="main">  
<div id="head">
```

Ivan Kuprinning On-line kundaligi.

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

Keyingi blok links ochroq fonda beriladi. Bu yerda o'rtada saytning boshqa bo'limlariga xavola (ssilka) lar joylashadi.

```
#links {  
background:#EDEDED;  
padding: 2px 0;  
text-align:center }
```

Xavola (ssilka) larni bir biridan ajratish uchun uzluksiz probel va vertikal chiziqlar qo'yilgan. Bundan tashqari xavola (ssilka) lar matni qalin shriftda va tagiga chizilmagan holda aks etsin. Buning uchun konMatnselektorga quyidagicha qoida yozamiz.


```
#links A{
color:#000;
font weight: bold;
text decoration: none; }
```

Blokning HTML kodi quyidagi ko'rinishni oladi:

```
<body>
<div id="main">
<div id="head">
On-line kundalik
</div>
<div id="links">
<a href="#">kundalik</A>&nbsp;||&nbsp;
<a href="#">fotolar</A>&nbsp;||&nbsp;
<a href="#">dil izhorlari kitobi</a>
</div>
</div>
</body>
```

Pozitsion. Pozitsion hususiyati quyidagi to'rtta qiymatlardan birini qabul qilishi mumkin.

1. static - bu oddiy blok bo'lib, normal oqim qoidalari bo'yicha joylashadi.

2. relative - normal oqimga nisbatan nisbiy joylashish.

3. absolute - absolyut joylashish.

4. fixed - fiksirlangan joylashish. Blok avval absolyut joylashadi, ya'ni normal oqimdan ajralib chiqib, alohida joylashadi, keyin uning shu joylashish holati fiksirlanadi. Fiksirlangan blok sahifa ko'rib chiqish jarayonida surilganida, o'z holatini o'zgartirmaydi.

Nisbiy pozitsiyalash(relative). Bu usuldan foydalanganda, avval blokning holati normal oqimga nisbatan aniqlanadi, so'ngra blok o'z joyiga nisbatan suriladi. Surilish quyidagi hususiyatlar yordamida beriladi:

left - blokni chap chegarasi, konteynerni chap chegarasiga nisbatan suriladi;

right - blokni o'ng chegarasi, konteynerni o'ng chegarasiga nisbatan suriladi;

top - blokni yuqori chegarasi, konteynerni yuqori chegarasiga nisbatan suriladi;

bottom - blokni quyi chegarasi, konteynerni quyi chegarasiga nisbatan suriladi.

Qiymatlar uzunlik o'lchov birligida yoki protsentlarda beriladi.

Oddiy misol ko'ramiz. Sahifaga uchta blokni joylashtiramiz.

```
div{ border: 1px dotted #000; }
```

```
<div>Birinchi blok</div><br>
```

```
<div>Ikkinchi blok</div> <br>
```

```
<div>Uchinchi blok</div>
```

Bloklardan birini chapga 20 px, yuqoridan pastga 10 px suramiz.

Buning uchun "move" nomli sinf yaratamiz:

```
.move{ position: relative; left: 20px; top: 10px; }
```

```
<div> Birinchi blok</div><br />
```

```
<div class="move"> Ikkinchi blok</div><br />
```

```
<div> Uchinchi blok</div>
```

Absolyut pozitsiyalash. Absolyut pozitsiyalshda blok normal oqimdan butunlay ajralib chiqadi. Bunda ham bloklarni surish quyidagi qiymatlardan birini oladi:

left

right

top

bottom

Bizning misolimizda ikkinchi blok uchun konteyner vazifasini brauzer oynasi o'taydi.

Shuning uchun surishni oynaga nisbatan beramiz. Suriladigan blok uchun "abs" deb identifikator belgilaymiz.

```
#abs{ position: absolute; left: 60px; top: 120px; }
```

```
<div> Birinchi blok</div><br />
```

```
<div id="abs"> Ikkinchi blok</div><br />
```

```
<div> Uchinchi blok</div>
```

Fiksirlangan bloklar. Fiksirlangan blok hech qayerga surilmaydi va har doim o'rnatilgan sohada o'zgarishsiz turadi.

Aniq misol ko'ramiz, ikkita blok yaratamiz. Birinchi blokni fiksirlab, brauzerning chap yuqori oynasiga joylashtiramiz. Bu blokga menu punktlarini joylashtiramiz. Blokning kengligini 100 piksel bilan cheklaymiz.

```
#fix{ position: fixed; left: 0px; top: 0px; width:100px; }
```

Ikkinchi blok absolyut joylashgan, u birinchi blok bilan ustma-ust tushib qolmasligi uchun, o'ng tomonga 150 piksel surilgan.

```
#move{ position: absolute; left: 150px; top: 0px; }
```

Endi bu bloklardan sahifa yaratamiz.

```
<div id="fix">
```

```
1 punktl<br>
```

```
2 punkt<br>
3 punkt<br>
4 punkt </div>
<div id="move">
Sahifaning asosiy matni<br>
Sahifaning asosiy matni<br>
Sahifaning asosiy matni<br>
</div>
```

Suzuvchi blok. Float hususiyati elementni sahifaning chap yoki o'ng chetiga siqib olib keladi. Matn esa uning atrofidan aylanib o'tadi. Matnning bunday harakati toshni yonidan oqib o'tuvchi suvni eslatadi, shuning uchun ham bunday hususiyatli bloklar, suzuvchi blok deyiladi. Absolyut pozitsiyadan farqli o'laroq, suzuvchi bloklar aniq koordinataga ega bo'lmaydi. Ularni aniq berilgan nuqtaga joylashtirish mumkin emas, lekin ular o'ziga xos muhim xarakterga ega.

Float qo'llaniladigan ishlarni sanab o'tamiz:

- Fotografiyalar galereyasi;
- Ikki va uch ustunli (kolonka) maketlar;
- Gorizontaal menyu;
- Matnni mazmunini yorituvchi rasmlar;
- Ko'p ustunli matn.

Float quyidagi qiymatlardan birini qabul qilishi mumkin.

left strukturali blok chapga suriladi, hujjatning qolgan qismi esa blokning o'ng tomoniga o'tadi.

right strukturali blok o'ng tomonga suriladi, hujjatning qolgan qismi esa blokning chap tomoniga o'tadi.

none blok surilmaydi, ya'ni positionda ko'rsatilgan algoritim bo'yicha joylashadi. Agar position hususiyati ko'rsatilmasa, normal oqim holiday bo'ladi.

inherit o'zi joylashgan konteynerni qiymatini oladi.

Misol keltiramiz:

```
<html>
<head>
<title>float</title>
<style type="text/css">
.comment { float: left; background: #fd0; border: 1px solid black;
padding: 10px;
margin-right: 20px; /* ong hoshiya */
width: 30%; /* blokni kengligi */
```

```
}  
</style>  
</head>  
<body>  
<div class="comment">  
Word Matn protsessorida ishlash tartibi - quyidagilarni o'z ichiga  
oladi:
```

```
</div>
```

```
<div>
```

Hujjat yaratish; Matn kiritish; Matnni tahrirlash; Matnni taqrizlash;
Matnni formatlash; Hujjatni saqlash; Hujjatni chop etish.

```
</div>
```

```
</body>
```

```
</html>
```

Natija brauzer oynasida quyidagi ko'rinishda aks etadi:

NAZORAT SAVOLLARI

1. Blok nima va uning qanday hususiyatlari bor?
2. Blok uchun hoshiya qanday o'rnatiladi?
3. Blok uchun chegara chiziqlari qanday o'rnatiladi?
4. Padding nima va qanday o'rnatiladi?
5. CSS yordamida ro'yxat yaratishni qanday usullari mavjud?
6. Bloklarni joylashtirish usullari?

9-BOB. CSS3 KASKADLI JADVALLAR STILI HUSUSIYATLARI

9.1. HTML5 va CSS3 standart. CSS3 kategoriyalar bo'yicha yangi hususiyatlari

CSS – bu stillar bilan ishlash uchun mo'ljallangan kodlar majmuasidir. Xozirgi kunga kelib yangi CSS3 versiyasi ishlab chiqilmada, lekin xali ko'p brauzerlar bu stillarni aks ettira olmayapti. Shuning uchun CSS3 yaratuvchilari brauzer o'rtasida kelishmovchiliklarni oldini olish maqsadida brauzer turiga qarab turli xil prefikslar ishlab chiqishdi va bu prefikslarni CSS3 to'liq yaratilmaguncha ishlatish tavsiya etilgan. CSS3 da eng asosiy qo'shilgan stillardan biri bu animasiyalardir, ya'ni stillar almashinish jarayoni birdaniga emas, balki sekin astalik bilan sodir bo'lishidir. Quyida CSS3 versiyasiga yangi qo'shilgan ba'zi stillarni misol tariqasida ko'rsatib o'tamiz.

Elementlarni (matn, rasm, fon,..) och ko'rinishda aks ettirish.

Bu yerda sahifa foniga rang berib, shu rangni 0.7 darajali och tusga keltirilmoqda, bu CSS3 ning yangi elementi "rgba" yordamida amalga oshiriladi. Elementlarning istalgan burchagiga (ismiga) bir vaqtning o'zida turli xil rasmlarni joylashtirish. Bu saytni joylashuv jarayonini yengillashtiradi.

```
background:
```

```
url(top.gif) topleftno-repeat,
```

```
url(center.png) top 11px no-repeat,
```

```
url(bottom.png) bottom left no-repeat,
```

```
url(middle.png) left no-repeat;
```

"Resize" buyrug'i. Bu buyruq orqali foydalanuvchi istalgan elementning o'lchamini o'zgartirishi mumkin bo'ladi.

```
div.resize {
```

```
width: 25px;
```

```
height: 35px;
```

```
resize: both;
```

```
}
```

"CSS3" ning yana bir yangiligi bu – elementlar (shakllar, bloklar) burchagini istalgan radiusda burish mumkinligidir. Ko'p web saytlar forma yaratish jarayonida, formaning burchaklarini burishni joylashtirish paytida biror rasm orqali amalga oshirishadi, bu esa qo'shimcha ish va vaqt yo'qotishga olib keladi, CSS3 da bu muammo bartaraf etilgan va quyidagicha amalga oshirish mumkin bo'ladi.

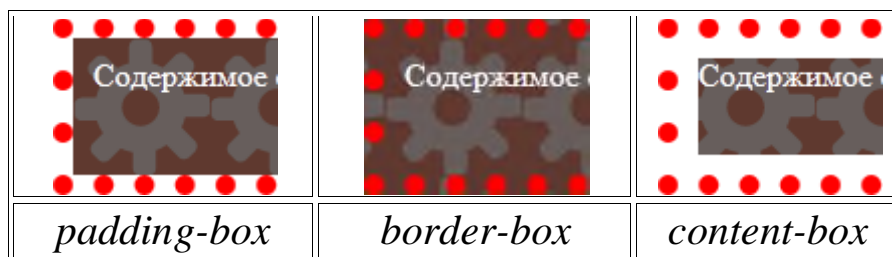
```
#forma {
border-bottom-right-radius: 2em;
border-bottom-left-radius: 1em;
border-top-left-radius: 5em;
border-top-right-radius: 3em;
}
```

CSS3ning soyalar bilan ishlash qismi. Quyida barcha <p> teglari uchun soyalar hosil qilish keltirib o'tilgan.

```
p {
text-shadow: #003471 /* soya rangi */ 2px /* o'ng tomonga
surilishi*/ 5px /* pastga surilish */ 2px /* degradasiyasi*/;
}
```

Shriftlar. Internetda ko'p foydalanuvchilar verdana shrifti bilan ishlaydi. Nega? Chunki bu shrift barcha kompyuterlarda mavjud va brauzerda chiroyli ko'rinishga ega. Agar stillarda qo'llanilgan shrift foydalanuvchi kompyuterida mavjud bo'lmasa, brauzer matnni istalgan boshqa shriftga ko'rsatishi mumkin. Bu esa shriftlar rang barangligiga olib keladi. CSS3 da shriftlar bilan ishlash uchun yangi komanda @font-face mavjud.

CSS3 da fon yaratish. Background-clip xususiyatlari. Bu xususiyat fon ishlatish yoki yo'qligini aniqlaydi, fon chegaralangan soxa ichida bo'lsinmi yoki tashqarisidami, shuni aniqlash uchun ishlatiladi. Masalan quyidagi dastur matni fon elementining barcha qismlarini chegaralangan soxagacha ifodalaydi: background-clip:border-box; Agar e'lon fon elementning chegaralarida ifodalansa: background-clip:padding-box; Agar e'lon fon elementning ichki qismini ifodalasa: background-clip:content-box; 9.1-rasmda brauzerda ifodalangan uchta element ko'rsatilgan: birinchida Safari brauzerda background-clip xususiyat border-box qiymat bilan; ikkinchida Safari brauzerda background-clip xususiyat padding-box qiymat bilan; uchinchidan Safari brauzerda background-clip xususiyat content-box qiymat bilan; Ichki blokning birinchi qatorida elementni ixtiyoriy yerda ifodalashga ruxsat beriladi. Bundan shuni aytish mumkinki u birinchi blok doirasida ifodalanadi, bunda stil chegarasi punktir bilan aniqlanadi. Ikkinchi qatorda soxa doirasida fon va uning tasvirlari ifodalanmaydi, ular background-clip xususiyatning padding-box qiymatiga ko'ra aniqlanadi. Uchinchi qatorda xususiyatlar element xususiyatiga background-clip ga content-box qiymatni berish bilan har bir elementning ichki mazmunini aniqlash uchun ishlatiladi.



9.1-rasm. Turli qiymatlarni qo'llash natijasi

Misol:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>background-clip</title>
    <style>
      .example {
        background: #5f392f url(images/gear.png); /* Fon rasmi*/
        border: 10px dotted red; /* Ramka parametrlari */
        background-clip: border-box; /* Fon ramka ichida */
        padding: 10px; /* Maydon */
        color: #fff; /* Matn rangi */
        min-height: 48px; /* Minimal balandlik */
      }
    </style>
  </head>
  <body>
    <div class="example">Содержимое</div>
  </body>
</html>

```

background-origin xususiyati.

Bu xususiyat bilan fon tasvir qayerda turishini va qayerdan boshlanishini aniqlashtirish (ko'rsatish) mumkin bo'ladi. Masalan keyingi e'lon fon tasviri tashqi chegaraning chap yuqori qismidan boshlanishini ko'rsatadi.

Qiymat

padding-box

Fon chegara kengligini hisobga olgan holda element burchaklariga nisbatan aniqlashtiriladi.

border-box

Fon chegaraga nisbatan aniqlanadi, shuning uchun chegara chiziqlari tasvirni to'sib qo'yishi mumkin.

content-box

Fon element tashkil etuvchilariga nisbatan aniqlanadi.



9.2-rasm. Turli qiymatlarni qo'llash natijasi

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>background-origin</title>
<style>
.example {
border: 20px solid #fc0;
padding: 20px;
height: 200px;
background: url(images/figure.jpg) no-repeat;
background-origin: content-box;
}
</style>
</head>
<body>
<div class="example">...</div>
</body>
</html>
```

9.2. CSS3-transformasiyalar

CSS3-transformasiya transform vositasi yordamida web-sahifadagi element holati, shakli va o'lchamini o'zgartiradi. Transformatsiyalar elementni web-sahifaning boshqa elementlariga ta'sir qilmasdan o'zgartiradi, ya'ni boshqa elementlar unga nisbatan harakat qilmaydi. Odatda transformasiya markaz elementiga nisbatan amalga oshadi.

O'zgartirish mumkin bo'lgan elementlar display: block; va display: elementlarni hamda display qiymat hususiyatlaridagi table-row, table-row-group, table-header-group, table-footer-group, table-cell yoki table-caption elementlarini o'z ishiga oladi. Transform xususiyatlarining hech qanday qiymatiga ega bo'lmagan element, o'zgaruvchan narsa deb hisoblanmaydi.

CSS3-transformasiyaning ikkita ko'rinishi mavjud 2D va 3D. 2D-transformasiyalar elementlarni ikki o'lchamli fazoda aks ettiradi.

Elementlarning 2D-transformasiyasini quyidagi brauzerlar qo'llab quvvatlaydi.

IE: 10.0, 9.0 -ms-

Edge: 12.0

Firefox: 16.0, 3.5 -moz-

Chrome: 36.0, 4.0 -webkit-

Safari: 9.0, 3.1 -webkit-

Opera: 23.0, 15.0 -webkit-

iOS Safari: 9, 7.1 -webkit-

Android Browser: 53, 2.1 -webkit-

Chrome for Android: -webkit-

2D-transformasiya funksiyalari (transform). Xususiyat elementni o'zgartirish turini bildiradi. Xususiyat elementni sahifadagi joriy pozitsiyaga nisbatan harakatlantiradigan yoki original o'lcham va shaklni o'zgartiradigan aylantirish vazifalari yordamida tasvirlangan.

Ruhsat etiladigan qiymatlar:

matrix() – ixtiyoriy son

translate(), translateX(), translateY() – uzunlik birliklari (salbiy va ijobiy), %

scale(), scaleX(), scaleY() – ixtiyoriy son

rotate() - burchak (deg, grad, rad yoki turn)

skew(), skewX(), skewY() - burchak (deg, grad, rad)

Funksiya	Izoh
none	Bu funksiya transformasiya mavjud emasligini bildiradi. Undan tashqari o'zgartirilayotgan elementlar guruhidagi elementlar uchun transformasiyani bekor qiladi.
matrix(a, c, b, d, x, y)	Elementlarni o'zgartiradi va ularni o'zgartirish usullarini belgilaydi, bu sizning 2D o'zgarishlaringizning bir

	<p>necha funktsiyalarini biriga birlashtirishga imkon beradi. Transformasiya sifatida burish, masshtablashtirish, bukish va holatni o'zgartirish mumkin.</p> <p>a qiymat gorizontal bo'yicha masshtabni o'zgartiradi. 0 dan 1 gacha qiymat elementni kichraytiradi, 1 dan katta esa kattalashtiradi.</p> <p>c qiymat Y o'qi bo'yicha element tomonlarini harakatlantiradi, yuqoriga – ijobiy qiymat, pastga – salbiy qiymat.</p> <p>b qiymat X o'qi bo'yicha element tomonlarini harakatlantiradi, chapga – ijobiy qiymat, o'ngga – salbiy qiymat.</p> <p>d qiymat vertikal bo'yicha masshtabni o'zgartiradi. 1 dan kichik qiymat elementni kichraytiradi, 1 dan katta qiymat kattalashtiradi.</p> <p>x qiymat X o'qi bo'yicha elementlarni aralashtiradi, o'ngga – ijobiy, chapga – salbiy.</p> <p>y qiymat Y o'qi bo'yicha elementlarni aralashtiradi, pastga – ijobiy qiymat, yuqoriga – salbiy qiymat.</p>
$\text{translate}(x,y)$	X va Y koordinatalaridan foydalanib, qo'shni elementlarga ta'sir qilmasdan elementni o'ngga va pastga an'anaviy holatga qarab harakat qilib, yangi joyga siljitadi. Agar elementni chap yoki yuqoriga siljitish kerak bo'lsa, u holda salbiy qiymatlardan foydalanish kerak.
$\text{translateX}(n)$	X o'qi bo'yicha odatiy holatga nisbatan elementni siljitadi.
$\text{translateY}(n)$	Y o'qi bo'yicha odatiy holatga nisbatan elementni siljitadi.
$\text{scale}(x,y)$	Elementlarni katta yoki kichik holatda

	<p>masshtablashtiradi. 0 dan 1 gacha qiymat elementni kichiklashtiradi. Birinchi qiymat elementni eni bo'yicha masshtablashtiradi, ikkinchi qiymat – balandligi bo'yicha masshtablashtiradi. Salbiy qiymatlar elementni oyna ko'rinishida akslantiradi.</p>
scaleX(n)	<p>Funksiya elementni eni bo'yicha toraytiradi yoki kengaytiradi. Agar qiymat 1 dan katta bo'lsa, element kengayadi, agar qiymat 0 va 1 oralig'ida bo'lsa, element torayadi. Salbiy qiymatlar elementni gorizontal bo'yicha oyna ko'rinishida aks ettiradi.</p>
scaleY(n)	<p>Funksiya elementni balandligi bo'yicha yuqori va pastga masshtablashtiradi. Agar qiymat 1 dan katta bo'lsa, element yuqoriga masshtablanadi, agar qiymat 0 va 1 oralig'ida bo'lsa, pastga masshtablanadi. Salbiy qiymatlar elementni oyna kabi vertikal bo'yicha aks ettiradi.</p>
rotate(burchak)	<p>Belgilangan graduslarda elementni aylantiradi, salbiy qiymatlar -1deg dan -360deg gacha soat millariga teskari holatda elementlarni aylantiradi, ijobiyalar esa – soat millari bo'yicha aylantiradi. rotate(720deg) qiymati elementlarni ikkita to'liq aylanada aylantiradi.</p>
skew(x-burchak, y-burchak)	<p>Koordinata o'qlariga nisbatan element tomonlarini deformatsiya (buzilish) qilish uchun foydalaniladi. Agar bitta qiymat ko'rsatilgan bo'lsa, ikkinchi qiymat brauzerda avtomatik aniqlanadi.</p>
skewX(burchak)	<p>X o'qiga nisbatan element tomonlarini</p>

	deformasiyalaydi.
skewY(burchak)	Y o'qiga nisbatan element tomonlarini deformasiyalaydi.
initial	Xususiyat qiymatini standart qiymatga sozlaydi.
inherit	Ota-ona elementidan xususiyat qiymatini meros qilib oladi.

Sintaksis.

```
div {
  -webkit-transform: rotate(360deg);
  -ms-transform: rotate(360deg);
  transform: rotate(360deg);
}
```

2. Transformasiya nuqtasi transform-origin. Xususiyat element shakli, o'lchami va holat o'zgarishi yuzaga kelishiga nisbatan, markaziy transformasiyani aralashtirishga imkon beradi. Qiymati - center yoki 50% 50%. Faqatgina transformasiyalanadigan elementlar uchun beriladi.

transform-origin	
Qiymat:	
X o'qi (left, center, right, uzunligi, %) Y o'qi (top, center, bottom, uzunligi, %)	Foizlarda, uzunlik birliklarida yoki kalit so'zlar yordamida berilgan qiymatlar juftligi, nisbatan elementning qaysi qismida transformasiya bajarilayorganini aniqlaydi. 100% dan katta qiymat elementning transformasiya maydonini kattalashtiradi.
initial	Xususiyat qiymatini standart qiymatga sozlaydi.
inherit	Ota-ona elementidan xususiyat

	qiymatini meros qilib oladi.
--	------------------------------

Sintaksis

```
div {  
-webkit-transform: rotate(45deg);  
-ms-transform: rotate(45deg);  
transform: rotate(45deg);  
-webkit-transform-origin: 20% 40%;  
-ms-transform-origin: 20% 40%;  
transform-origin: 20% 40%;  
}
```

3. Ko'p qiymatli transformasiyalar. Paydo bo'lish tartibida bo'sh joy orqali ularni hisoblash bilan bitta elementning bir qancha transformasiyalarini birlashtirish mumkin.

```
div {transform: scale(1.5) rotate(-10deg);}
```

NAZORAT SAVOLLARI

1. HTML5 va CSS3 standartlari?
3. CSS ni CSS3 dan farqli tomonlari?
4. CSS va CSS3 da rasmlarni ekranga joylashtirish?
5. CSS3 da transformasiyalar va ularning turlari?

10-BOB. JAVASCRIPT ASOSLARI. O'ZGARUVCHILAR. MA'LUMOTLAR TURI. MASSIVLAR. OPERATORLAR. DASTURLASH ASOSLARI

10.1. JavaScript ssenariyli tili asoslari

JavaScript brauzerda mijoz tomonidan web sahifalar ssenariyalarini boshqaruvchi tildir.

Bu tilga murojaat qilish uchun ochilgan va yopilgan HTML teglar ichiga script so'zi quyidagicha yoziladi:

`<script>` va `</script>`. Odatiy hujjat Hello World, HTML 4.01 da ifodalangan bo'lsa unga JavaScriptni tadbqiq qilsak u quyidagicha bo'ladi, Masalan: Hello World matni JavaScript vositasida quyidagicha ifodalanadi:

```
<html>
<head><title> Hello World </title></head>
<body>
<script type = "text/javascript">
document.write("Hello World")
</script>
<noscript>
```

Sizning brauzeringiz JavaScript ni qo'llab-quvvatlamaydi yoki u o'chirilgan

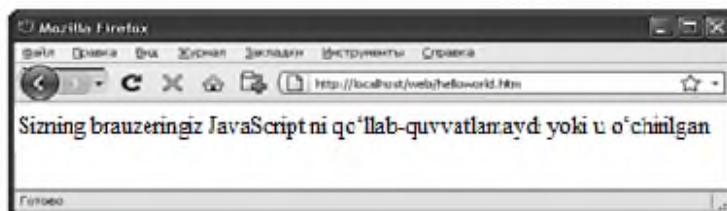
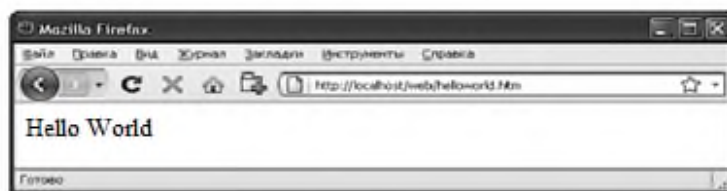
```
</noscript>
</body>
</html>
```

Bunda faqat bitta satr JavaScriptga tegishli, `<script>` tegi ostidagi `document.write` komanda bu komanda PHP dagi `echo` yoki `print` komandalariga ekvivalentdir. Bu satr undagi satr ma'lumotni ekranga chiqaradi. Bunda shunga e'tibor qilish mumkin komandalardan keyin nuqtali vergul ishlatilmaydi, chunki bunda nuqta vergul xat boshiga o'tishni anglatadi. Agar bitta satrga bir nechta instruksiyalarni joylashtirish kerak bo'lsa nuqta vergul (oxirgisidan tashqari) qo'yiladi.

Bu misolda `<noscript>` va `</noscript>` teglar juftligiga e'tibor qaratishimiz kerak, bu boshqa foydalanuvchilar uchun qaysiki JavaScript ni qo'llab quvvatlamaydigan brauzerlarni ishlatayotganda yoki brauzer ishlaymay qolganda turli xil xabarlar chiqarish va xakozolar uchun ishlatiladi. Yoqoridagi misolni ishga tushirsak quyidagicha matn paydo bo'ladi:

Hello World

Agarda JavaScript ajratilgan bo'lsa u holda 10.1-rasmdagi kabi xabar chiqadi:



10.1-rasm. Birinchi holat ishga tushgan holat, ikkinchisi JavaScript o'chirilgan holat

10.2. O'zgaruvchilar va ma'lumotlar turi

JavaScript tilida o'zgaruvchilarni ishlatish mumkin va ularni nomlari bilan manzillash mumkin. O'zgaruvchilar global va lokal bo'lishi mumkin. Global o'zgaruvchilarga ssenariyning istalgan joyida ruxsat bo'lishi mumkin. Lokal o'zgaruvchilarning harakati esa e'lon qilingan o'zgaruvchilar ichidagi funksiyalar bilan chegaralangan. Basic dasturlash tili singari JavaScript ssenariysini yaratayotgan vaqtda avvaldan e'lon qilinmagan o'zgaruvchilarni ishlatish mumkin.

O'zgaruvchilar e'loni. Java Script da hamma o'zgaruvchilar var kalit so'zi orqali e'lon qilinadi va quyidagicha amalga oshiriladi:

```
var MyHelloMsg;
```

O'zgaruvchi turi o'zlashtiriladi qachonki, unga biror bir qiymat o'zlashtirilsa, quyida avvaldan e'lon qilinmagan matnli qator o'zgaruvchiga yozilmoqda:

```
MyMsg = "Salom!";
```

MyMsg o'zgaruvchi nomi o'zlashtirilgandan so'ng ruxsat beriladi.

O'zgaruvchi nomini tanlaganda, quyidagi oddiy qoidalarni eslab qolish kerak:

O'zgaruvchi nomi harflardan yoki "_", "\$" belgilardan boshlanish kerak va faqat harflardan, sonlardan va "_", "\$" belgilardan iborat bo'lishi kerak;

O'zgaruvchilar nomi JavaScript ning zaxiralangan kalit so'zlari bilan

mos kelmasligi kerak.

10.1-jadval. JavaScript ning zaxiralangan kalit so'zlari

break	case	catch	Class	continuye	const
debugger	default	delete	Do	else	enum
export	extends	false	Finally	for	function
if	import	in	New	null	return
super	switch	this	Throw	true	try
typeof	var	void	While	with	

O'zgaruvchining qiymatini o'zlashtirish. "=" o'zlashtirish operatori yordamida o'zgaruvchilar qiymati o'zlashtiriladi. Misol qilib quyidagi o'zgaruvchi keltirilgan va unda matnli qator yozilgan:

```
var MyHelloMsg;
```

```
MyHelloMsg = "Hello, world!";
```

MyHelloMsg sonli o'zgaruvchini dasturning istalgan joyida o'zlashtirish mumkin, misol uchun:

```
MyHelloMsg = 4;
```

Bu operator bajarilgandan so'ng o'zgaruvchi turi o'zgaradi, shuningdek interpretatsiya jarayonida brauzer hech qanday o'gohlantiruvchi xabarlarni yubormaydi.

O'zgaruvchini maxsus null qiymati orqali o'zlashtirish mumkin:

```
MyHelloMsg = null;
```

Bunday o'zlashtirish hech qanday turda o'zgaruvchini belgilamaydi.

JavaScript da ma'lumotlar turi. JavaScript tilida bir nechta ma'lumotlar turi mavjud. Bular sonlar, matnli qatorlar, mantiqiy ma'lumotlar, ob'yektlar, aniqlanmagan turli ma'lumotlar hamda maxsus tur null.

Sonlar. JavaScript tili turli formatdagi sonlarni ishlatishga ruxsat beradi, Bular butun sonlar, suzuvchi nuqtali o'nli formatdagi sonlar va ilmiy notatsiya sonlar. Butun sonlar 8, 10, 16 asosida berilishi mumkin. Misol uchun:

```
25 10 asosidagi butun son
```

```
0137 8 asosidagi butun son
```

```
0xFF 16 asosidagi butun son
```

```
386.7 Suzuvchi o'nli nuqtali son
```

```
25e5
```

```
yoki 25E5 Ilmiy notatsiyadagi son, 2500000 ga teng.
```

Ayrim hollarda "son bo'lmagan" arifmetik funksiyalar kelib chiqishi mumkin. JavaScript da aytilganidek NaN (Not a Number). "Son

bo'lmagan" – bu hech qanday songa loyiq bo'lmagan maxsus qiymat. Bu sonlar ustida operatsiya bajarilayotgan vaqtda, va natija son ko'rinishida taqdim etilmagan hollarda paydo bo'ladi. "Son bo'lmagan" qiymatga to'g'ri kelishini isNaN funksiyasi yordamida tekshirish mumkin.

Matnli qator. Matnli qator – bu bir yoki qo'shtirnoq ketma ketlik belgisi, misol uchun:

```
"Hello, world!"
```

```
""
```

```
"12345"
```

```
'Bu matnli qator'
```

"" qatori –bo'shdir. Quyidagi 2 ta o'zlashtirish ekvivalent emasligini aniqlaymiz:

```
MyStr=""
```

```
MyStr1=null
```

Birinchi holda MyStr o'zgaruvchisida matnli qator saqlanmoqda (bo'sh bo'lsa ham), ikkinchisida esa hech narsa.

Mantiqiy ma'lumotlar. Mantiqiy ma'lumotlar faqat 2 ta qiymatni, ya'ni True va False ni o'z ichiga oladi. Bu qiymatlar 0 va 1 sonlar bilan bog'liq emas. Bu qiymatlarning asosiy obrazi solishtirish operatsiyasi bajarilayotgan vaqtga qaratilgan, hamda shartli operatsiyalar ishlatilganda ham.

Aniqlanmagan turli ma'lumotlar. Agar o'zgaruvchi e'lon qilingan bo'lsa, ammo unga xali qiymat o'zlashtirilmagan bo'lsa, u holda u aniqlanmagan turga ega bo'ladi. Misol uchun quyidagi qatorda aniqlanmagan turga ega bo'lgan MyVariable o'zgaruvchisi e'lon qilingan: *var MyVariable;*

Agarda bu o'zgaruvchini null qiymati bilan o'zgartirsak, u holda o'zgaruvchi turi o'zgaradi va null qiymatga ega bo'lgan o'zgaruvchiga aylanadi: *MyVariable = null;*

Ma'lumotlar turini o'zgartirish. Agarda ifodalarda turli xil turli o'zgaruvchilar uchrab qolsa, JavaScript interpretatori avtomatik holda sonli ma'lumotlarni matnli qatorlarga o'zgartirishi mumkin. Teskari aylantirishni (qatorni-songa) maxsus funksiyalar yordamida, ya'ni `parseInt` va `parseFloat` funksiyalari yordamida o'zgartirish mumkin. Buni quyidagi misol orqali ko'rish mumkin:

Misol

```
<HTML>
```

```
<HEAD><TITLE>Type conversion sample</TITLE></HEAD>
```

```
<BODY>
```

```

<H1>Type conversion sample</H1>
<TABLE>
<SCRIPT LANGUAGE="JavaScript">
var MyTextBuf = "";
MyTextBuf = 4 + " - to'rt soni" + "<BR>";
MyBuf2 = (parseInt("2") + 2) + "&nbsp; - to'rt soni" + "<BR>";
document.write(MyTextBuf + MyBuf2);
</SCRIPT>
</TABLE></BODY></HTML>

```

Bu yerda biz *My Text Buf* o'zgaruvchisini e'lon qildik va uni bo'sh qator bilan initsializatsiya qildik. Quyida biz bu qatorda 4 sonni summasini va 2 ta matnli qatorni o'zlashtirdik.

```
My Text Buf = 4 + " - to'rt soni" + "<BR>";
```

Bu ifoda hisoblanayotgan vaqtda 4 qiymat avtomatik holda matnli qatorga o'zgaradi. Keyingi yig'indi HTML hujjatlarida ishlatiladigan ajralmaydigan probel belgisiga ahamiyat berish lozim. Agarda uni oddiy probelga almashtirsak, qator konkatenatsiyasidan keyin bu probel yo'qoladi. Keyingi qatorda parseInt funksiyasi yordamida matnli qator "2" sonli qatorga o'zgaradi, natijada 2 soni qo'shiladi, undan keyin har ikkala matnli qatorlarda konkatenatsiya bajariladi:

```
My Buf 2 = (parseInt("2")+2)+" - to'rt soni" + "<BR>";
```

Natijada MyBuf2 o'zgaruvchisi MyTextBuf o'zgaruvchisi kabi qatorga ega bo'ladi.

10.3. JavaScript tili operatorlari. Unar operatori

Unar operatori belgining o'zgarishi uchun to'ldirish operatsiyasini bajarishda, inkrement hamda dekrementda ishlatiladi:

- teskari holatda belgining o'zgarishi

! Qo'shimcha. Mantiqiy o'zgaruvchilarning qiymatini tisinfh uchun ishlatiladi.

++ O'zgaruvchi qiymatini oshirish. O'zgaruvchi prefiksi yoki uning suffiksi bo'lib qo'llanishi mumkin.

-- O'zgaruvchi qiymatini kamaytirish. O'zgaruvchi prefiksi yoki uning suffiksi bo'lib qo'llanishi mumkin.

Unar operatorini ishlatishga doir misollar:

```
i=0; // i teng 0 dagi o'zgaruvchining boshlang'ich qiymati
```

```
i++; // i teng 1 dagi qiymat
```

```
--i; // i teng 0 dagi qiymati
```

```
var j=3; // j teng 3 dagi o'zgaruvchining qiymati
i = -j; // i teng -3 dagi o'zgaruvchining qiymati
var fYes = true; // fYes teng true dagi o'zgaruvchining qiymati
testFlag(!fYes); // testFlag funksiyasiga false qiymati uzatilmoqda
```

Binar operatori. Binar operatori 2 ta operandni birlashtiradi. JavaScript tilida binar operatorlari ayirish, bo'lish, qo'shish, ko'paytirish hamda bo'linmani qoldig'ini hisoblash uchun ishlatiladi (ko'riladi):

- Ayirish
- + Qo'shish
- * Ko'paytirish
- / Bo'lish
- % Bo'linmani qoldig'ini hisoblash

Bu operatorlar C tilida ishlatilganidek JavaScript da ham xuddi shunday ishlatiladi, misol uchun:

```
i=0; // i teng 0 dagi o'zgaruvchining qiymati
i = i + 1; // i teng 1 dagi qiymat
var j=9; // j teng 9 dagi o'zgaruvchining qiymati
i = j / 2; // i teng 4 dagi o'zgaruvchining qiymati
k = j % 2; // i teng 1 dagi o'zgaruvchining qiymati
```

Aloxida bitlar bilan ishlash operatori. Ssenariyalarda shunday operatorlar ishlatiladiki, ular aloxida bitlar bilan ishlash operatorlari hisoblanadi, ular quyidagilar: VA, YOKI, MAXSUS YOKI, INKOR:

- & VA
- | YOKI
- ^ AYRIM HOLLARDA YOKI
- ~ INKOR

Siljitish operatorlari

JavaScriptda siljish amalini bajarish uchun 3 ta operator ko'rilgan:

>> Siljish o'ng tomonga

<< Siljish chap tomonga

>>> Bo'shatiladigan razryadlarni nollar bilan to'ldirib o'ngga siljish.

Munosobat operatorlari. Munosobat operatorlari o'zgaruvchilarning qiymatini solishtirish uchun ishlatiladi. Bu operatorlar solishtirish natijalariga bog'liq true yoki false mantiqiy qiymatlarni qaytaradi va shartli operatorlarda asosiy bo'lib ishlatiladi. True qiymatini qaytaradigan JavaScript tilining munosabat operatorlari ko'rsatilgan:

> Chap operand o'ng operanddan katta

>= Chap operand o'ng operanddan katta yoki teng

< Chap operand o'ng operanddan kichik

<= Chap operand o'ng operanddan kichik yoki teng

== Chap operand o'ng operandga teng

!= Chap operand o'ng operandga teng emas

Mantiqiy operatorlar. || YOKI operatori. True qiymat qaytaradi, qachonki operandlardan biri true bo'lsa.

&& VA operatori. True qiymat qaytaradi, qachonki ikki operand true bo'lsa

O'zlashtirish operatori. O'zlashtirish operatori o'zgaruvchilarning qiymatini o'zlashtirish uchun ishlatiladi. JavaScript tilida va C dasturlash tilidagi kabi bu operator boshqa operatorlar bilan kombinatsiyasiga ruxsat etiladi. Quyida o'zlashtirish operatorini boshqa operatorlar bilan kombinatsiyasi berilgan:

= Oddiy o'zlashtirish

+= Sonli qiymatni kattalashtirish yoki qatorlarni qo'shilishi

-= Sonli qiymatni kichisinfhtirish

*= Ko'paytirish

/= Bo'lish

%= Bo'lishdan qolgan qoldiqni hisoblash

>>= O'ngga siljish

>>>= Bo'shatiladigan razryadlarni nollar bilan to'ldirib o'ngga siljish

<<= Chapga siljish

|= YOKI

&= VA

^= AYRIM HOLLARDA YOKI

C tili bilan tanish bo'lmaganlar uchun o'zlashtirish operatorini boshqa operatorlar bilan birgalikda ishlatilishi qiyinroq va g'ayrioddiy tuyulishi mumkin, lekin aslida ssenariyni osonlashtiradi, boshlang'ich matni soddalashtiradi.

Masalan sonli o'zgaruvchilar qiymatini oshirish uchun += operatori ishlatiladi. Avval bu vazifani yechimini += operatorini ishlatmagan holatda ko'rib chiqamiz. Quyida nCounter o'zgaruvchisi e'lon qilindi va unga boshlang'ich 1 qiymati o'zlashtirildi, so'ng bu qiymat 5 ga oshirildi:

```
var nCounter = 1;
```

```
nCounter = nCounter + 5;
```

Endi buni += operatori yordamida bajaramiz:

```
var nCounter = 1;
```

```
nCounter += 5;
```

Ko'rinib turibdiki 2-usul 1-usulga nisbatan qisqa.

O'zgaruvchi qiymatini 3 razryadga o'ngga siljitish uchun >>= operatoridan foydalanish mumkin va u quyidagi matnda ko'rsatilgan:

```
nCounter >>= 3;
```

Natija esa quyidagi matnda ko'rsatilganidek bo'ladi:

```
nCounter = nCounter >> 3;
```

Shart operatorlari. If operatori JavaScript dasturlash tilidagi muhim operatorlardan biridir. U shartga bog'liq ravishda kod fragmentini bajarishga mo'ljallangan. if operatorining strukturasi quyidagicha ifodalash mumkin:

1- variant. if operatori sintaksisi:

```
if(shart)
{
  Amallar ketma ketligi
}
```

2- variant. if operatori sintaksisi:

```
if(shart)
{
  Amallar ketma ketligi 1
}
else
{
  Amallar ketma ketligi 2
}
```

Bu yerda shart JavaScript tilidagi mantiqiy yoki ixtiyoriy shartdir. Agarda shart mantiqiy turdagi o'zgaruvchi bo'lsa, qiymati rost (True) bo'lsa, u holda *Amallar ketma ketligi1* bajariladi, aks holda *Amallar ketma ketligi2* bajariladi.

Misol. if shart operatori

```
function checkData()
{
  if (document.form1.threyeChar.valuye.length==3)
  {return true;
  }
  else
  { alert('3 raqamini kiriting');
  return false;
  }
}
```

else operatori. Biz yuqorida faqat *if* operatorining asosiy qisminigina ko'rdik. Bu operatorning bir nechta kengaygan shakli mavjud. *else* operatori *if* operatorini tekshirilayotgan ifoda noto'g'ri bo'lgan holatdagina kengaytiradi hamda bu holatda yangi shartda biror amal bajaradi.

else operatori yordamida kengaytirilgan *if* operatorining strukturasi quyidagicha ifodalash mumkin:

```
if (shartli ifoda) bajariladigan_blok1  
else bajariladigan_blok2
```

Bu *if...else* konstruksiyasi quyidagicha interpretatsiya qilinishi mumkin: agar shart bajarilsa (ya'ni ifoda=true), u holda *bajariladigan_blok1* dagi amallar bajariladi, aks holda *bajariladigan_blok2* dagi amallar bajariladi. *else* operatoridan foydalanish majburiy emas.

elseif operatori. *if* shart operatorining yana bir kengaygan shakli – bu *elseif* operatorining qo'llanilishidir. *elseif* – bu *else* hamda *if* operatorlarining kombinatsiyasidir. U xuddi *else* operatori kabi *if* operatorini shart bajarilmagan holda kengaytiradi. Biroq *else* operatoridan farqi bir-biriga zid amallarni faqat *elseif* shart rost bo'lgandagina bajaradi. *else* hamda *elseif* operatorlari yordamida kengaytirilgan *if* operatorining strukturasi quyidagicha ifodalash mumkin:

```
if (ifoda1) bajariladigan_blok1  
elseif (ifoda2) bajariladigan_blok2  
else bajariladigan_blokN
```

elseif operatorlari bitta *if*-blokida bir necha marta uchrashi mumkin. *elseif* tasdig'i faqat oldinda turgan *if*-shartlari hamda *elseif*-shartlari False qiymatni, berilgan *elseif*-sharti esa True qiymatni qaytargandagina bajariladi.

```
Misol. elseif operatori  
function makeMinutes() {  
  var minstring="";  
  var now = new Date();  
  var min = Date.getMinutes();  
  if (min<10) {  
    minstring+=":0"+min;}  
  elseif(min>10){  
    minstring+=":"+min;}  
  return minstring;  
}
```

Switch operatori. Yana bir shartni tekshirib turli amallarga bog'liq ravishda ish ko'rsatadigan konstruksiya bu – *switch* operatoridir. Bu operatorni o'zbek tiliga tarjima qilinganda “yo'nalishni o'zgartirgich” ma'nosini beradi hamda bu operatorning vazifasi ham shunga o'xshashdir. O'zgaruvchini qanday qiymatni qabul qilishiga bog'liq ravishda u yo'nalishni o'zgartirib turli bloklardagi amallarni bajaradi. *switch* operatori *if...elseif...else* yoki *if* operatori majmuiga juda o'xshash bo'ladi. *switch* operatorining strukturasi quyidagicha ifodalash mumkin:

```
switch (ifoda yoki o'zgaruvchi){
  case qiymat1:
    amallar_bloki1
    break;
  case qiymat2:
    amallar_bloki2
    break;
  ...
  default:
    amallar_bloki_avtomatik_tarzda
}
```

if operatoridan farqli tomoni bu yerda ifodalar mantiqiy tur qabul qilmay, balki faqat *case* kalit so'zidan keyingi qiymatlarni (*qiymat1*, *qiymat2* va h.k.) taqqoslaydi holos. Agar ifoda qiymati qandaydir variant bilan ustma-ust tushsa, ikki nuqtadan keyingi *break* operatorigacha bo'lgan *amallar_bloki*dagi amallarni bajaradi. Agarda ifoda qiymati berilgan variantlardan hech biriga ustma-ust tushmasa, *default* kalit so'zidan keyingi avtomatik tarzda bajariladigan blok (*amallar_bloki_avtomatik_tarzda*) bajariladi. *switch* operatoridagi ifoda faqat bir marta hisoblanadi, *elseif* operatorida esa har bir tekshirishda hisoblanadi, shuning uchun agarda ifoda yetarli darajada murakkab bo'lsa, u holda *switch* operatori tezroq ishlaydi.

Misol: *switch* operatori:

```
function getName(){
  var names = array("Name1"," Name2"," Name3");
  var selected="";
  switch (names[0]){
  case "Name1": selected="Name1 is selected";
  break;
  case "Name2": selected="Name2 is selected";
  break;
```

```

case "Name3": selected="Name3 is selected";
break;
default: selected="Default select"+$names[0];
}
return selected;
}

```

Yuqoridagi misoldan ko'rinib turibdiki, getName() funksiyasi ishga tushirilganda, names massividagi birinchi element tekshiriladi. Bu holda massiv 1-elementi Name1 bo'lgani uchun bizga selected="Name1 is selected"; qiymatni qaytaradi.

Misol: *switch* operatori:

```

var change = prompt ("Harakatni tanlang:\n1 – Mashina sotib olish\n2 – Mashina sotish \n3 – Mashina almashtirish ");
switch (change) {
case "1": {
document.write ("Siz mashina sotib olishni xoxlaysiz ");
break;
}
case "2": {
document.write ("Siz mashina sotishni xoxlaysiz ");
break;
}
case "3": {
document.write ("Siz mashina almashtirishni xoxlaysiz ");
break;
}
default: {
document.write ("Siz noto'g'ri komanda kiritdingiz ");
break;
}
}
}

```

switch operatorining konstruktiviyasi uchun xuddi *if* operatori kabi alternativ sintaksisi mavjud. Bu yerda *switch* operatoridagi ochiladigan figurali qavs ikki nuqtaga o'zgartiriladi, yopiladigani esa mos ravishda *endswitch*; kalit so'ziga o'zgartiriladi.

Ifoda va shart. O'zgaruvchi va massivlar yordamida siz o'zingizga kerakli bo'lgan axborotni turli ko'rinishda kompyuter xotirasida saqlaysiz. Biroq bu axborotni siz faqatgina ekranga chiqarish yoki oqoxlantiruvchi darchada joylashtirish uchun emas, balki ko'proq amallar bajarishingiz

mumkin. Siz uni o'zgartirishingiz, uni boshqarishingiz yoki tekshirishingiz mumkin. Ushbu maqsadlar uchun ifoda va shartlar keltirilgan.

Ifodalar (expressions) ikki yoki undan ortiq qiymatlarni birlashtirish uchun va natijada yangi qiymatga ega bo'lish uchun ishlatiladi. Quyidagi yig'indili ifoda bunga misol bo'la oladi:

$$1+2=3$$

Shartlar (conditions) qiymatlarni solishtirish va mantiqiy qiymatni (true yoki false) aniqlash imkonini beradi.

$$3+3=6 ? \text{ true}$$

Amallar bilan tanishuv. Shart va ifodalarda ma'lumotlar bir-biri bilan amallar bilan birlashtiriladi. Agar harakat bitta kattalik bilan amalga oshirilsa u holda bu amal bitta operandli amal (unary operator) deb ataladi. Agar bunaqa kattaliklar 2 ta bo'lsa, u holda bunday amallar 2 ta operandli amallar (Binary operator), agar 3 ta bo'lsa - 3 ta operandli amallar (ternary operator) deyiladi.

Arifmetik amallar. Arifmetik amallar bu barchaga ma'lum bo'lgan matematik amallar hisoblanadi:

- qo'shish (+) $1+3=4$
- ayirish (-) $2-1=1$
- bo'lish (/) $4/2=2$
- ko'paytirish (*) $2*2=4$
- bo'lishdan qolgan qoldiq (%) $9\%5=4$

Java Script ssenariylariga misollar.

Quyida arifmetik amallardan foydalanilgan bir nechta misollar keltiriladi. Bu yerda SCRIPT blokining o'zi berilgan.

Misol. Qo'shish amaliga misol 10.2- rasm

```
<script language="JavaScript" >  
var a=6, b=4;  
alert(a+b);  
</script>
```



10.2-rasm. Qo'shish amali natijasi

Misol. Ayirish amaliga misol 10.3- rasm

```
<script language="JavaScript ">  
var a=6, b=4;  
alert(a-b);  
</script>
```



10.3- rasm. Ayirish amali natijasi

Misol. Bo'lish amaliga misol 10.4- rasm

```
<script language="JavaScript ">  
var a=6, b=3;  
alert(a/b);  
</script>
```



10.4- rasm. Bo'lish amali natijasi

Misol. Ko'paytirish amaliga misol 10.5- rasm

```
<script language="JavaScript ">  
var a=6, b=3;  
alert(a*b);  
</script>
```



10.5- rasm. Ko'paytirish amali natijasi

Misol. Bo'lishdan qolgan qoldiq amaliga misol 10.6- rasm

```

<script language="JavaScript">
var a=6, b=4;
alert(a%b);
</script>

```



10.6- rasm. Bo'lishdan qolgan qoldiq amali natijasi

Taqqoslash amallari ifodalarni bir-biri bilan solishtirishda ishlatiladi. Ularga quyidagi amallar kiradi:

- kichik (<) $5 < 6$
- kichik yoki teng (<=) $6 <= 6$ yoki $6 <= 7$
- katta (>) $7 > 4$
- katta yoki teng (>=) $5 >= 5$ yoki $5 >= 4$
- teng (==) $5 == 5$
- teng emas (!=) $5 != 3$

Bularning barchasi 2 ta operandli amallarga kiradi. Bu misollarda sonli ma'lumotlar taqqoslanadi, biroq satrlarda ham ushbu amallar ishlatilishi mumkin. Eng asosiysi shuki, bitta turdagi ma'lumotlar taqqoslanilishi mumkin. Aks holda Java Script bir turdagi ma'lumotlarni boshqa turga o'tkazishga harakat qiladi va bu har doim ham amalga oshirilaverilmaydi. Xatolarga yo'l qo'ymaslik uchun bitta turdagi ma'lumotlarni taqqoslang.

Mantiqiy amallar. Quyida 3 ta mantiqiy amallar keltirilgan:

- mantiqiy XA (and) &&
- mantiqiy YOKI (or) ||
- mantiqiy YO'Q (not) !

Bitta operandli amallar. Nomidan ham ko'rinib turibdi bu amal bitta kattalik bilan amalga oshiriladi. Bularga:

- prefiksli va postfiksli o'sish (increment) - ++
- prefiksli va postfiksli kamayish (decrement) - --
- unar plyus +
- unar minus -

Unar minus ifoda ishorasini qarama-qarshisiga aylantiradi. 4 ta amal ichida bu eng oddiysi hisoblanadi. Bundan tashqari unar plyus ishorani almashtirish uchun emas, balki operandni songa aylantirish uchun ishlatiladi (masalan, bu satr bo'lgan holda).

Prefiksli amalda agar $a=5$ bo'lsa, u holda $++a+2=8$ bo'ladi, chunki o'zgaruvchi qiymatiga 2 soni qo'shilmadan oldin u bittaga orttirilgan edi. Shu vaqtning o'zida $-a+2=6$, chunki o'zgaruvchi qiymati 2 ga kamaytirilmasdan oldin u bittaga kamaytirilgan edi.

Postfiksli amalda esa aksincha, ifodada o'zgaruvchining eski qiymati ishlatilib, so'ngra uning qiymati bittaga kamaytiriladi yoki orttiriladi. Shu tariqa agar $a=5$ bo'lsa, u holda $a++ +2=7$, chunki o'zgaruvchining qiymati ifoda hisoblangandan so'ng bittaga orttiriladi. Bu yerda $a-- +2=7$ ham o'rinlidir, chunki o'zgaruvchining qiymati ifoda hisoblangandan so'nggina bittaga kamaytiriladi.

Misol. Prefiksli o'sish 10.7-rasm

```
<script language="JavaScript" >  
var a=6, b=2;  
alert(++a+b);  
</script>
```



10.7-rasm. Prefiksli o'sish

O'zlashtirish amali. Oddiy o'zlashtirish amali ($=$) bo'lib, u o'zgaruvchiga qiymatni yusinfh uchun xizmat qiladi. Siz bu amal bilan tanishsiz, bunga "a" o'zgaruvchiga 3 sonini yusinfhni misol qilish mumkin $a=3$.

Bu amaldan foydalanib bitta qiymatni bir nechta o'zgaruvchiga yusinfhni ham bajarish mumkin. Quyidagi misolda 5 qiymatini bir nechta (a, b, c va d) o'zgaruvchilariga o'zlashtirish ko'rsatilgan:

$a=b=s=d=5$

Boshqa o'zlashtirish amallari qo'shma amallar hisoblanib quyida ular keltirilgan:

- $x+=3$ bu amal $x=x+3$ bilan teng kuchlidir
- $x-=3$ bu amal $x=x-3$ bilan teng kuchlidir
- $x*=3$ bu amal $x=x*3$ bilan teng kuchlidir

- $x/=3$ bu amal $x=x/3$ bilan teng kuchlidir

Misol. += qo'shma amali 10.8- rasm

```
<script language="JavaScript">
var x=6;
alert(x+=2);
</script>
```



10.8- rasm. += qo'shma amali natijasi

Misol. -= qo'shma amali 10.9- rasm

```
<script language="JavaScript">
var x=6;
alert(x -=2);
</script>
```



10.9- rasm. -= qo'shma amali natijasi

JavaScriptda massivlar. Siz massiv haqida eshitgan bo'lsangiz kerak, chunki bu tushuncha har bir dasturlash tilida mavjud yoki matematika fanida (matritsa deb balki) eshitgandirsiz. Ko'pchilik array atamasini ishlatishni ma'qul ko'radi. Dasturlashda: massiv bu — ko'p sondagi o'zgaruvchilarni bitta o'zgaruvchi ko'rinishida ifodalash. Javascriptda u quyidagicha ko'rinishda e'lon qilinadi.

```
var array-name = [item1, item2, ...];
```

Bu yerda:

1. array-name - massiv nomi
2. [] - massiv tanasi, oddiy qilib aytganda uning ichida massiv elementlari yoziladi
3. item1 va item2 - massiv elemntlari.

Javascriptda massiv o'zgaruvchisini e'lon qilishning 2 usuli bo'lib ular quyidagicha yoziladi:

```
var cars = ["Matiz", "Nexia", "Spark"];
```

```
var cars = new Array("Matiz", "Nexia", "Spark");
```

Ikkalasi ham yangi massivni yaratadi, ikkala usul ham to'g'ri, lekin 1-usulni ishlatish maslahat bariladi. Chunki uning ko'rinishi oddiy, tushunishga oson va yozish uchun qulay.

```
var arr = new Array(2,3); // massiv yaratildi, qiymati: [2, 3]
arr = new Array(2); // massivda [2] element aniqlandimi?
alert(arr[0]); // yo'q! bizning massiv qiymati: [undefined, undefined]
```

Massivda elementlar sanog'i 0 dan boshlanadi cars[0] degani bu massivdagi 1-elementga muroajaat qilish bo'ladi. Massiv elementlariga muroajaat quyidagicha amalga oshiriladi:

```
var name = cars[0]; // massiv elementini boshqa o'zgaruvchiga o'zlashtirish
```

```
cars[1] = "Matiz"; // massiv elementini o'zgartirish. Agar massivni 2-elementi aniqlanmagan bo'lsa, bu ifoda massivda yangi element qo'shish bo'ladi.
```

JavaScriptning boshqa dasturlash tillaridan farqli hususiyatlaridan biri massiv elementlari turlari har xil bo'lishi mumkin. JavaScript bunga imkon beradi, masalan:

```
myArray[0] = Date.now;
myArray[1] = myFunction;
myArray[2] = myCars;
```

Massiv ustida length va sort amallarini bajarish.

length — bu massiz uzunligi, elementlar sonini bildiradi.

sort() — massiv elemntlarini alfabit bo'yicha o'sish tartibida saralaydi. Quyidagi misollarda '15' '2' dan oldin turib qolgan, chunki saralash paytida JavaScript ularni oddiy char (char bu bir simvol) deb hisoblaydi va '15' dagi 1-simvol '1' '2' bilan solishtiriladi.

Misol.

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
alert(fruits.length); // 4
alert(fruits.sort()); // ["Apple", "Banana", "Mango", "Orange"]
var arr = [ 1, 2, 15 ];
alert( arr.sort() ) // 1, 15, 2
var index, text = "";
for(index = 0; index < fruits.length; index++) {
    text += fruits[index];
}
var fruits = [] // bo'sh massiv
fruits[1] = 'Peach'
fruits[99] = 'Apple'
```

```

    alert(fruits.length) // 100 (lekin massivning 2 ta elementi aniqlangan
holos)
    var emptyArray = [];
    var days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday',
'Friday', 'Saturday', 'Sunday']
    var emptyArray = [];
    var days = ['Monday', 'Tuesday', 'Wednesday', 'Thursday',
'Friday', 'Saturday', 'Sunday']
    for(index = 0; index < 10; index++){
        emptyArray[emptyArray.length] = Math.floor(Math.random() * 6)
+ 1
    }
    for(index = 0; index < emptyArray.length; index++){
        console.log(days[emptyArray[index]]);
    }

```

NAZORAT SAVOLLARI

1. JavaScript tili ob'ekti va uning vazifalari, turlari?
2. JavaScript tili funksiyasini HTML hujjatda e'lon qilish?
3. JavaScript tilida massivlarni e'lon qilish?
4. JavaScriptda o'zgaruvchilar va ma'lumotlar turlari?
5. JavaScriptda operatorlar va ularning turlari?

11-BOB. JAVASCRIPT BIRIKTIRILGAN FUNKSIYA VA OB'YEKTLARI. GRAFIKA VA DOIMIY IFODALAR

11.1. JavaScript da funksiyalar

Boshlang'ich matn bo'lagini funksiya ko'rinishida yozish mumkin va ularni JavaScript ssenariysining turli joylaridan murojaat qilish mumkin. Odatda funksiyalar HTML hujjatini sarlavxa bo'limida aniqlanadi. Funksiyalar chaqirilishidan avval e'lon qilinishi kerak va barcha funksiya e'loni HTML hujjat sarlavxasida joylashtirilgan bo'lishi kerak.

Funksiyaning umumiy e'loni quyida keltirilgan:

```
function nomi(scriptter 1) [...,scriptter 2] [...,scriptter N])
{ ...
  Funksiya matni qatorlari
  ...
  [return qiymat]
}
```

Barcha parametrlar funksiya qiymatiga beriladi. Shuning uchun funksiya unga scriptter sifatida beriladigan o'zgaruvchilar qiymatini o'zgartira olmaydi.

Return kalit so'zi yordamida funksiya qiymati qaytariladi.

Java Script tilida tuzilgan ko'pchilik dasturlarda funksiyalardan foydalanamiz. Shuning uchun xozir bu tilning muhim elementi haqida gapirish juda muhim. Funksiyalar ko'pchilik hollarda o'zida bir necha buyruqlarni bog'lash usulini namoyon qiladi. Keling misol uchun biror bir matnni uch marta birdaniga chop qiluvchi script yozamiz. Dastlab oddiy yo'lni ko'raylik:

```
<html >
< script language= "JavaScript">
< ! - - hide >
document.Write( "Mening sahifamga xush kelibsiz! < br >");
document.Write( "Bu JavaScript! < br >");
document.Write( "Bu JavaScript! < br >");
document.Write( "Bu JavaScript! < br >");
< /Script >
< /html >
```

Va bu script qo'shimcha matnni bosib chiqaradi:
Mening sahifamga xush kelibsiz!
Bu JavaScript!

Bu JavaScript!

Bu JavaScript!

Agar dastlabki kodiga qaraydigan bo'lsak, kerakli natijani olish uchun kodning aniq bir qismi 3 marta takrorlangan. Axir bu samaralimi?

Yo'q biz ushbu masalani osonroq yechishimiz mumkin:

```
< html >
< script language= "JavaScript">
< ! - - hide
function myfunction () {
document. Write( 'Mening sahifamga xush kelibsiz! < br >');
document. Write( 'Bu JavaScript! < br >');
}
myfunction ();
myfunction ();
myfunction ();
< /Script >
< /html >
```

{ } qavs ichida joylashgan barcha skriptlar myfunction() funksiyasiga tegishli. Bu shuni anglatadiki document.write() dan iborat 2 ta buyruqni birlashtirildi va ular funksiya chaqirilishi bilan birgalikda bajariladi. Haqiqatdan ham bizning misolimizda funksiya 3 marta chaqirilgan va buni funksiyaning o'zi aniqlagandan so'ng myfunction() satrini 3 marta yozganimizni ko'rishimiz mumkin. Va o'z navbatida bu shuni bildiradiki, funksiyaning tarkibidagi narsalar (buyruqlar, { } qavs ichidagi ko'rsatilganlar) 3 marta bajarilgan. Bu funksiyaning ishlatilishiga yetarli darajadagi oddiy misol va sizda nima uchun funksiya JavaScript uchun shu darajada muhim degan savol tug'iladi. Ushbu qo'llanmani o'qish mobaynida siz albatta funksiyaning foydasini bilib olasiz. Funksiyani chiqarishda o'zgaruvchilarning uzatish imkoniyatining o'zi bizning skriptlar uchun juda ixchamlik yaratadi.

Funksiyalar yana xodisalarni ishlovchi protseduralar bilan birgalikda ishlatilishi mumkin. Navbatdagi misolni ko'ramiz:

```
< html >
< script language= "JavaScript">
< head>
function calculation( ){
var x = 12;
var y = 5;
var result = x+y;
```

```

    alert (result);}
    //- -
  </script>
</head>
<body>
<form>
  <input type = "button" value = "calculate" onclick =
"calculation()">
</form>
</body>
</html>

```

Bu yerda tugmachani bosish bilan calculation() funksiyasini chiqarish jarayoni ro'y beradi. Sezgan bo'lsangiz, bu funksiya x, y va result o'zgaruvchilaridan foydalanib, qo'shish amalini bajaradi. O'zgaruvchilarni var kalit so'zi bilan e'lon qilish mumkin. O'zgaruvchilar turli kattaliklarni son, satr va xakozalarni saqlash uchun ishlatiladi. Skriptning var result = x+y satri brauzerga result o'zgaruvchisini hosil qilish va unga x+u arifmetik amalni joylashtirishni (5+12) xabar beradi. Bundan so'ng result o'zgaruvchisi 17 qiymatni oladi. Bu holda alert (result) va alert (17) buyrug'i bir xil vazifani bajaradi. Boshqacha qilib aytganda, bu 17 son yozilgan darchaga ega bo'lamiz.

JavaScript ssenariyli tili ob'yektga-mo'ljallangan tildir. JavaScript ob'yektlari xususiyatlar va usullar to'plamini ifodalaydi. Ob'yekt xususiyati – bu, ob'yektga bog'liq bo'lgan ma'lumotlardir, usullar esa – ob'yekt ma'lumotlarini qayta ishlovchi funksiyalardir. JavaScript ssenariyda xususiyatlarni manzillash ularning nomlari bilan yoki ularning raqamlari bilan amalga oshishi mumkin. Keyingi variant bo'yicha, har bir xususiyat massivning bir elementi sifatida olinadi va ular o'zlarining unikal raqamlariga ega bo'ladilar.

JavaScript tilida C va Java dasturlash tillaridagi kabi protsedura va funksiyalar mavjud bo'lib, ular quyidagicha e'lon qilinadi:

- function kalit so'zi;
- funksiya nomi;
- vergul va qavs bilan ajratilgan argumentlar ro'yxati;
- figurali qavs ichiga olingan funksiya tanasi.

Function myFunction(arg1, arg2, ...)

{

...

Operatorlar ketma-ketligi

```
...  
}
```

Bu yerda myFunction – funksiya nomi, arg1, arg2 – parametrlar.

Misol:

```
function Factorial(n) {  
  if((n<0)||((round(n))!=n)) {  
    alert(“Factorial funksiyasi ushbu argumentda aniqlanmadi “+n);  
    return NaN;  
  }else{  
    result=(n*Factorial(n-1));  
    return result;  
  } }  
}
```

Funksiyada return kalit so’zi orqali qiymat qaytarilmasligi ham mumkin.

Misol:

```
function Greying(s){  
  document.write(“Hello, ”+s+ ”!”);  
  return ;  
}
```

Funksiyani chaqirish aniq parametrlar bilan chaqiriladi:

Misol:

Factorial(3); - bu funksiya natijasi 6 ga teng,

Greying(“world”); - bu funksiya ekranga “Hello, world!” satrini chiqaradi.

Har bir funksiya, masalan, myFunction funksiyasi myFunction nomli ob’yekt hisoblanadi, agarda argumentlar arguments nomi bilan berilsa, unga murojaat quyidagicha amalga oshiriladi:

myFunction.arguments[i], bu yerda i — argument raqami (raqamlash 0 dan boshlanadi).

Funksiya e’lonida aniq parametrlar formal parametrlarga teng yoki ko’p sonda bo’lishi lozim. Bunda funksiya ishga tushirilganda jo’natilayotgan argumentlar miqdori myFunction.arguments.length maydoni yordamida aniqlanadi va ushbu maydondagi qiymatni qayta o’zlashtirishni dinamik o’zgartirish mumkin.

Misol:

Ekranga HTML formatidagi ro’yxatni chiqarish.

Bu yerda ListTypening birinchi argumenti tartiblanmagan ro’yxat uchun “o” yoki “O”, tartiblanmagan ro’yxat uchun “u” yoki “U” bo’lishi mumkin.

```

Function myList(ListType)
{
document.write("<" + ListType + "L");
for(var i=1; I < myList.arguments.length; i=i+1) {
document.write("<LI> " + myList.arguments[i]);
}
document.write("</" + ListType + "L> ");
}

```

HTML hujjatida funksiyaga murojaat quyidagicha:

```

<script>
myList("o", "matn", 2, "3")
</script>

```

Natija:

matn

2

3

Global sinfi. Ushbu sinf JavaScriptning funksional qismi bo'lib, bu sinf bir ob'yektda bir nechta usul va xossalarni birlashtirish vazifasini bajaradi. Usulga murojaat qilinganda ob'yekt ko'rsatilmaydi, aniqrog'i bu usul konstruktorga ega bo'lmaydi. Bunday xossa va usullarga quyidagilarni keltirish mumkin:

11.1-jadval. Global sinfi xossalari.

Xossa	Mazmuni
Nan	NaN (NotANumber)
Infinity	Number.POSITIVE_INFINITY qiymatni o'z ichiga oladi

11.2-jadval. Global sinfi usullari.

Usul	Mazmuni
Escape	Qatorni barcha platformalarga mos holda tasvirlash
Eval	JavaScript tili funksiyasi yoki usullarini uzatish
isFinite	Argumentning oxirgi raqamliligini aniqlash
isNan	Argumentning raqam yoki raqam emasligini aniqlash
parseFloat	Qatorni ko'chib yuruvchi nuqtali son ko'rinishida tasvirlash
parseInt	Qatorni butun songa aylantirish
unescape	Escape funksiyasi natijasini qaytarish

eval(s) funksiyasi - s qatorni JavaScript operatorlari ketma-ketligi ko'rinishida tasvirlash.

getClass(Jobj) funksiyasi – JavaScript turidagi argument uchun JavaClass ob'yektini qaytaradi.

Misol:

```
var myJavaRClass=new java.awt.Rectangle()
```

```
var myJavaRClass=getClass(myJavaRect)
```

getClass() Java-usuli bilan adashtirmang:

```
var myJavaRObject=myJavaRect.getClass() - bu  
java.awt.Rectangle sinfining Java tilidagi amalga oshirilish holati.
```

isNaN(x) funksiyasi – x “Not a Number”, ya'ni son emasligini tekshirish.

parseFloat(s) funksiyasi – Float turidagi s raqamni aniqlash. Agar son topilmasa u holda NaN (“Not a Number”) qiymati qaytariladi.

parseInt(s) – Integer turi uchun yuqoridagi holat.

eval(s) funksiyasi. eval(s) funksiyasi – JavaScript ning ichki funksiyasi hisoblanadi. Ushbu funksiya bir yoki bir nechta JavaScript operatorlaridan iborat bo'lgan s satrni argument tomonidan uzatilgan kodni bajaradi. Bunda s satridagi operatorlar nuqtali vergul yordamida ajratiladi. Bu funksiya nafaqat operatorni bajarish, balki biror amallarni hisoblash imkonini ham beradi. Bunda u kodda keltirilgan amal hisobining oxirgi qiymatini qaytaradi.

isNaN(x) funksiyasi. Bu funksiya x argumentning “son emas” ligini tekshiradi. Natija NaN qiymatga ega emasligini, ya'ni mumkin bo'lmagan son (masalan, nolni nolga bo'lish natijasi) ni tekshiradi. Ushbu funksiya JavaScript da literal ko'rinishda NaN qiymatni berish mumkin emasligi uchun muhimdir. Bundan tashqari parseFloat(s) va parseInt(s) funksiyalar natijalarini tekshirish (mumkin bo'lgan son ekanligini) va arifmetik xatolar mavjudligi, masalan, nol soniga bo'lish mavjudligini tekshiradi.

parseFloat(s) funksiyasi. s satrini sintaktik tahlil qilish va dastlab raqamni qaytarish (satrni raqamga aylantiradi). parseFloat(s) da s satrida ruxsat etilmagan raqam elementlari (masalan, belgilar, raqam, o'qli vergullar, daraja ko'rsatkichi va hokazo) mavjud bo'lsa tahlil to'xtatiladi va qiymat qaytariladi. Agarda s satr son bilan boshlanmasa, u holda parseFloat(s) funksiya NaN qiymatni qaytaradi.

parseInt(s) funksiyasi. Bu funksiya satrni butun songa aylantiradi. parseInt(s) funksiyadagi s satrda hisoblash tizimida ko'rsatilmagan qiymatlarga ega bo'lganda sintaktik tahlil to'xtatiladi va qiymat qaytariladi. Odatda, parseFloat va parseInt funksiyalar s satr son bilan

boshlanmaganda NaN qiymat qaytaradi.

parseInt(s,n) holatida *n* asos hisoblanib, agarda *n*=10 bo'lsa, **parseInt(s)** funksiya satrdagi 10 lik sanoq tizimidagi sonlarni tekshiradi. *n*=8 bo'lsa, 8 lik sanoq tizimidagi sonlar mavjudligini (bunda *n* 0 dan 7 gacha bo'lgan sonlar qiymatiga ega bo'lishi mumkin). *n*=16 bo'lsa, 16 lik sanoq tizimidagi sonlar mavjudligini (bunda 0 dan 9 gacha bo'lgan sonlar va A dan F gacha bo'lgan harflar qiymatiga ega bo'linadi). Agarda *n*=0 bo'lsa yoki qiymat berilmasa, u holda **parseInt(s)** funksiya satrning o'zidan asosni aniqlaydi. Bu holatda agarda satr 0 bilan boshlansa, unda funksiya satrning qolgan qismini 16 lik sanoq tizimidagi son sifatida tahlil qiladi, agarda satr 0 dan boshlansa, satr 8 lik sanoq tizimidagi qiymat sifatida tahlil qilinadi.

11.2. JavaScriptda sikllar va ularning turlari

JavaScript tilida shartga bog'liq ravishda qaytariladigan amallardan iborat bir nechta konstruksiyalar mavjud. Bu *while*, *do..while*, *foreach* hamda *for* sikllardir. Ularni batafsil ko'rib chiqamiz.

While sikl operatori.

Strukturasi:

```
while (ifoda) { bajariladigan_blok }
```

yoki

```
while (ifoda): bajariladigan_blok endwhile;
```

while – bu oddiy sikl. U ifoda qiymati True (bu yerda xuddi *if* operatori kabi ifoda mantiqiy turga o'zlashtiriladi) bo'lgunicha *bajariladigan_blok*dagi buyruqlarni bajarishga buyuradi. Ifoda qiymati har sikl boshlanganda tekshirib boriladi, agarda uning qiymati *bajariladigan_blok* bajarilish jarayonida o'zgargan taqdirda ham iteratsiya tugamaguncha (ya'ni *bajariladigan_blok*dagi barcha buyruqlar bajarilmaguncha) sikl to'xtatilmaydi.

Misol: while operatori

```
i = 0
```

```
n = 0
```

```
while (i<5) {
```

```
  i ++;
```

```
  if (i == 3)
```

```
    continue n += i
```

```
}
```

Misol: while operatori

```

n = 0;
x = 0;
while( n < 3 ) {
  n ++;
  x += n;
}

```

do... while sikl operatori. *do..while* sikli *while* siklga o'xshaydi, ammo farqli tomoni shundaki, ifodaning rostligiga sikl boshida emas, balki oxirida tekshiriladi. Qulay tomoni shundaki, *bajariladigan_blok do..while* sikli ichida hech bo'lmaganda bir marta bajariladi.

Strukturasi:

```
do { do..while sikli } while (ifoda);
```

Misol. *do..while* operatori

```

do {
  i+=1;
  document.write(i);
} while (i<5);

```

For sikl operatori. Bu JavaScript tili sikl operatorlaridan biri. Ular C dasturlash tilidagi sikllar kabidir.

Strukturasi:

```
for (ifoda1; ifoda2; ifoda3) { bajariladigan_blok }
```

yoki

```
for (ifoda1; ifoda2; ifoda3): bajariladigan_blokendfor;
```

Bu yerda ko'rinib turibdiki shart uchta ifodadan tashkil topadi. Birinchi *ifoda1* ifoda sikl boshida shartsiz bajariladi. Har bir iteratsiyaning boshlanishida *ifoda2* bajariladi. Agar u True qiymatni qabul qilsa, u holda sikl o'z ishini davom ettiradi va *bajariladigan_blok*dagi barcha buyruqlarni bajaradi. Agar *ifoda2* False qiymatni qabul qilsa, u holda sikl to'xtatiladi. Har bir iteratsiya (ya'ni *bajariladigan_blok*dagi barcha buyruqlarni bajarilishidan keyin) oxirida *ifoda3* bajariladi.

Har bir 1-,2- va 3-ifodalar bo'sh bo'lishi mumkin. Agar *ifoda2* bo'sh bo'lsa, bu siklni cheksiz bajarilishini bildiradi. Bu unchalik befoyda emas, chunki siklni *break* operatori yordamida to'xtatsa bo'ladi.

Misol: Formada tanlangan elementlar sonini ekranga chiqarish

```
<SCRIPT>
```

```

function howMany(selectObject) {
  var numberSelected=0;
  for (var i=0; i < selectObject.options.length; i++) {
    if (selectObject.options[i].selected==true)

```

```

    numberSelected++;
}
return numberSelected;
}
</SCRIPT>
<FORM NAME="selectForm">
<P><B>Choose some music types, then click the button below:</B>
<BR><SELECT NAME="musicTypes" MULTIPLE>
<OPTION SELECTED> R&B
<OPTION> Jazz
<OPTION> Bluyes
<OPTION> New Age
<OPTION> Classical
<OPTION> Opera
</SELECT>
<P><INPUT TYPE="button" VALUE="How many are selected?"
onClick="alert('Number of options selected:' + howMany
(document.select Form. musicTypes))">
</FORM>

```

Agarda ushbu operator ichidagi barcha uchala ifoda ham tushirib qoldirilsa, u holda hisoblagich var i o'zgaruvchini boshlang'ich qiymati berilmaydi va har bir sikl oxirida u o'zgarmaydi. Bu barcha buyruqlarni alohida buyruqlar ko'rinishida yoki sikldan avval *bajariladigan_blok* ichida yozsa ham bo'ladi:

```

var i=0; // hisoblagichni boshlang'ich qiymatini beramiz
for ( ; ; ){
    if (i>=10) break;
    // agar i katta yoki teng 10 bo'lsa, u holda sikl ishini to'xtatamiz.
    if (i % 2 == 0) print $i;
    // agar son juft bo'lsa, uni ekranga chiqaramiz.
    i++; // hisoblagich qiymatini bittaga oshiramiz.
}

```

for sikli konstruksiyasidagi uchinchi ifodada verguldan keyin yana bir nechta oddiy buyruqlarni ham yozsa bo'ladi. Masalan, agar biz oddiygina barcha sonlarni ekranga chiqarmoqchi bo'lsak, dasturni quyidagicha yozsa bo'ladi:

```

for (i=0; i<10; document.write(i), i++)
/* Agarda bajariladigan_blok buyruqlardan tashkil topmagan
yoki bitta buyruqdan tashkil topgan bo'lsa,

```


figurali qavsga olingan qismni
tashlab ketsa bo'ladi.* /

Boshqaruv o'tkazuvchi operatorlar. Ba'zida sikl yoki uning alohida iteratsiya ishini tezda to'xtatishga to'g'ri keladi. Buning uchun *break* hamda *continue* operatorlari kerak bo'ladi.

Break operatori. Break operatori mavjud siklni amalga oshirishni tugallaydi, *for*, *while*, *do while* yoki *switch break* strukturani boshqaruvchi, sikl yoki shartni tekshirishni tugallash kerakligini bildiruvchi, uning tarkibiga kiruvchi raqamli argument bilan qo'llaniladi.

Misol. Break operatori
for (i = 0; i < a.length; i++) {
 if (a[i] = theValue)
 break;
}

Ushbu skriptda a massiv elementi the Value o'zgaruvchi qiymatiga teng bo'lganda sikl to'xtatiladi.

Continue operatori. Ba'zan sikl ishini butunlay to'xtatish lozim bo'lmaydi, faqatgina uning yangi iteratsiyasini boshlash kerak. *Continue* operatori istalgan siklni amalga oshirish blokidan keyingi instruktsiyalarni o'tkazib yuborish va yangi doira bilan amalga oshirishni davom ettirish imkonini beradi. *continue* ni uning tarkibida boshqariluvchi konstruktsiyalar ishini yakunlash kerakligini ko'rsatuvchi raqamli argument tarzida ishlatish mumkin.

Oldingi paragrafda berilgan misoldagi *break* operatorini *continue* ga almashtiramiz. Bundan tashqari to'rt sikli miqdorini kamaytiramiz.

Misol:
i = 0;
n = 0;
while (i < 5) {
 i++;
 if (i == 3)
 continue;
 n += i;
}

Bu skriptda agarda i o'zgaruvchi i=3 qiymatga ega bo'lsa, u holda n=(1,3,7,12) qiymatlarga ega bo'ladi.

Misol:
checkiandj :
 while (i<4) {

```

document.write(i + "<BR>");
i+=1;
checkj :
while (j>4) {
  document.write(j + "<BR>");
  j-=1;
  if ((j%2)==0)
    continue checkj;
  document.write(j + " is odd.<BR>");
}
document.write("i = " + i + "<br>");
document.write("j = " + j + "<br>");
}

```

Ushbu siklda agarda continue operatori topilsa, checkj sikli to'xtatiladi va checkj ning keyingi sikliga o'tiladi. Har safar continue topilganda checkj iteratsiyani boshidan boshlaydi, toki uning sharti false bo'lmaganicha. checkj sharti false bo'lganda, checkiandjning qolgan operatorlari bajariladi va bu sikl ham checkiandj sharti false bo'lguniga qadar davom etadi.

11.3. JavaScriptda grafika va doimiy ifodalar bilan ishlash

JavaScript dasturlashda ko'proq effektlarga grafika bilan ishlash orqali erishiladi. Bunda dasturchida unchalik ko'p instrumentlar yo'q: dokumentga ichki qurilgan rasmlar, image ob'ektini hosil qilish imkoniyati, rasmlarni gipermatnli o'tishlar va jadvallar bilar birga qo'llash. Shunga qaramasdan bu sodda usullar orqali erishiladigan effektlar juda ajoyib.

JavaScriptda grafikani dasturlash quyidagi hossa, metod va hodisalar bilan xarakterlanuvchi image ob'ektiga asoslanadi:

11.3-jadval. Grafika bilan ishlovchi hossa, metod va hodisalar

Hossalar	Metodlar	Hodisalar
border complete height hspace name src vspace	yo'q	onAbort onError onLoad

width		
lowsrc		

Hossalarning ko'pligiga qaramasdan ularning ko'pchiligini faqatgina o'qish mumkin, lekin o'zgartirib bo'lmaydi. Bunga metodlarning yo'qligi ham guvoh bo'la oladi. Lekin baribir ikki hossani o'zgartirish mumkin: src va lowsrc. Shularning o'zi ham rasmlar bilan ko'pgina effektlarni bajarish uchun yetarli.

Image sinfining barcha ob'ektlarini ichki qurilgan va dasturchi tomonidan yaratilganlarga bo'lish mumkin. Ichki qurilganlari – bu img konteyneridagi rasmlar. Agar bu rasmlar nomlansa, u holda ularga nomlari bo'yicha murojaat qilish mumkin:

```
<a href="javascript:void(0);"
onclick="window.alert('image
document.images[0].name)">
<img name=tuit src=images.gif border=0></a>
```

src va lowsrc hossalari. src va lowsrc hossalari dokumentning ichiga joylashtiriladigan rasmlarning turlarini aniqlaydi. Bunda lowsrc asosiy rasm yuklanayotgan paytda tasvirlanuvchi, odatda kichkina bo'ladigan, vaqtinchalik rasmni aniqlaydi, uning turi img konteynerining src atributi qiymatida ko'rsatiladi. src hossasi img konteynerining src atributi qiymatini qabul qiladi. Dasturchi src ni ham, lowsrc ni ham qiymatlarini o'zgartira oladi. Quyida src hossasiga misol keltirilgan:

```
document.i2.src="images2.gif";
```

Bu misoldan ko'rinib turibdiki, ichki qurilgan image ob'ektining src hossasi qiymatini o'zgartirish hisobiga joylashtirilgan rasmni moslashtirish imkoniyati mavjud. Agar siz berilgan sahifani birinchi marta ko'rayotgan bo'lsangiz (ya'ni rasm brauzer tomonidan keshga joylashtirilmagan), u holda rasmning asta-sekin o'zgarishi sezilarli bo'ladi.

Rasmni o'zgartirish. Rasmni o'zgartirish mumkin, faqatgina ichki qurilgan image ob'ektining src hossasiga yangi qiymat bergan holda. Aniqki, rasmni serverdan sekin yuklash tez varaqlab ko'rishga imkon bermaydi. Bu muammoni hal qilishga urinib ko'ramiz. Hususan, masala yechimi rasmni va uning tasvirlarini vaqt bo'yicha ko'paytirib borishdir. Buning uchun image ob'ektining konstruktoridan foydalaniladi:

```
<table>
<td>
<a href="javascript:void(0);"
onmouseover="document.m0.src=color[0].src; return true;"
```

```

onmouseout="document.m0.src=mono[0].src; return true;">

</a>
</td>
...
</table>

```

Kod bo'lagi sichqoncha kursorining o'tishida rasmning almashishi va qayta tikkalanishining tipik usulini ko'rsatadi. Tabiiyki, faqat bitta emas, balki birdaniga bir nechta rasmlarni almashtirish mumkin.

Shunga qaramay asosiysi rasmlarni almashtirishda emas, balki bu qanday tezlikda amalga oshirilishidir. Kerakli natijaga erishish uchun sahifa boshida tasvirlanishidan oldin grafika yuklanadigan rasmlar massivi hosil qilinadi (sahifani yuklashda status satriga e'tibor bering):

```

color = new array(32);
mono = new array(32);
for(i=0;i<32;i++)
{
mono[i] = new image();
color[i] = new image();
if(i.toString().length==2)
{
mono[i].src = "images0"+i+".gif";
color[i].src = "images0"+i+".gif";
}
else
{
mono[i].src = "images00"+i+".gif";
color[i].src = "images00"+i+".gif";
}
}

```

Yana bir xarakterli usul – javascript-kodni kechiktirib bajarish funksiyasini qo'llanilishi (eval()):

```

function def()
{
for(i=0;i<32;i++)
{
eval("document.m"+i+".src=mono["+i+"].src");
} for(i=0;i<5;i++)
{

```

```
eval("document.r"+i+".src=rm["+i+"].src");  
}}
```

NAZORAT SAVOLLARI

1. JavaScript da funksiyalar va ularni e'lon qilish?
2. JavaScript da global sinfi va uni e'lon qilish?
3. JavaScript da eval(s) funksiyasi va uning vazifasi?
4. JavaScript da shart operatorlari va ularning vazifalari?
5. JavaScript da sikl operatorlari va ularning vazifalari?
7. JavaScript da switch operatori va uning ishlash prinsipi?

12-BOB. JAVASCRIPTDA BRAUZER VA WEB HUJJAT OB'YEKTLARI MODELI BILAN ISHLASH

12.1. JavaScript da ob'yektlar

JavaScript tilida uch turdagi ob'yektlar mavjud: standart ob'yektlar, brauzer ob'ektlari va dasturchi tomonidan yaratiluvchi ob'yektlar. Ularning har biri o'zlarining tasnifi va xususiyatlariga ega.

Standart ob'yektlar. Quyida JavaScript da qo'llaniluvchi standart ob'yektlar, xususiyatlar va usullar keltirilgan. Ularni ishlatishda oldindan e'lon qilish talab etilmaydi.

12.1-jadval. Standart ob'yektlar.

Ob'yekt	Tasnifi
Array	Massiv
Boolean	Mantiqiy ma'lumotlar
Date	Kalendar vaqt
Function	Funksiya
Global	Global usullar
Math	Matematik konstanta va funksiyalar
Number	Son
Object	Ob'yekt
String	Satr

Standart ob'yektlar bilan qanday ishlash mumkin? Ancha oddiy. Ob'yektni amalga oshiruvchi dastur yoziladi va uning xususiyat va usullariga murojaat qilinadi. Misol sifatida joriy vaqtni ko'rsatuvchi HTML hujjat ko'ramiz.

```
<HTML>
<HEAD>
<TITLE>Joriy kun va vaqt </TITLE>
</HEAD>
<BODY BGCOLOR=WHITE>
  <H1> Joriy kun va vaqt </H1>
  <SCRIPT LANGUAGE="JavaScript">
    var dt;
    var MyDate="";
    dt = new Date();
```

```

    MyDate = "Date: " + dt.getDate() + "." + dt.getMonth() + "." +
dt.getYear();
    document.write(MyDate);
    document.write("<BR>");
    document.write("Time: " + dt.getHours()
    + ":" + dt.getMinutes() + ":" + dt.getSeconds());
</SCRIPT> </BODY></HTML>

```

Bu yerda JavaScript senariy new kalit so'zi yordamida Date ob'yektini yaratadi. Bunda Date konstruktori parametrlarsiz keltiriladi:

```
var dt;
```

```
dt = new Date();
```

```
MyDate = "Date: " + dt.getDate() + "."
```

```
  + dt.getMonth() + "." + dt.getYear();
```

getDate, getMonth va getYear usullar yordamida joriy sana olinadi.

Ushbu usullar dt ob'yekti uchun chaqiriladi.

Matn satri esa HTML hujjatga write usuli yordamida bosmaga chiqariladi. Bu usul document ob'yektining usuli hisoblanadi:

```
document.write(MyDate);
```

Date ob'yekti joriy vaqtni ham o'z ichiga oladi. Bu ma'lumotlar getHours, getMinutes va getSeconds (soat, minut va sekund) usullari yordamida ko'riladi:

```
document.write("Time: " + dt.getHours()
```

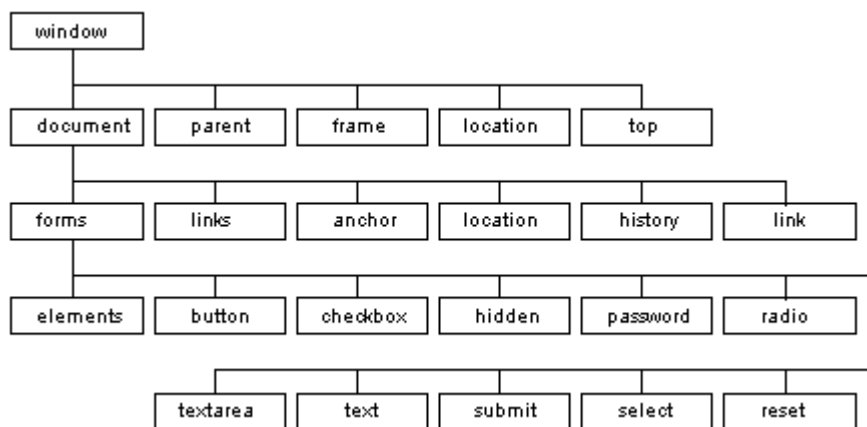
```
  + ":" + dt.getMinutes() + ":" + dt.getSeconds());
```

12.2. Brauzer ob'yektlari

JavaScript senariy niqtai nazari bo'yicha ob'yektlar iyerarhik darahit ko'rinishda tashkil etiladi.

Brauzer ob'yektlari foydalanuvchi uchun yaratilgan, brauzer oynasida joylashgan ob'yektlar hisoblanadi. JavaScript senariyida brauzer ob'yektlari, xususiyat va usullaridan foydalanib bir sinf asosida boshqa sinf yaratib bo'lmaydi.

Quyidagi rasmda ob'yektlar darahiti iyerarhiyasi keltirilgan.



12.1-rasm. Brauzer ob'yektlari iyerarhiyasi.

window ob'yekti bu iyerarhiyaning ildizi hisoblanadi. Qachonki HTML hujjat yuklansa uning ichida document, parent, frame, location va top boshqa ob'yektlar hosil bo'ladi.

Ob'yektlar bilan bog'liq holatlar. Brauzerning har bir ob'yekti bilan aniq bir holatlar to'plamidan tashkil topadi.

Masalan, window ob'yekti onLoad va onUnload holatlari bilan bog'liq holda ishlaydi. Birinchi holat brauzer oynani yuklab bo'lgach ishga tushad. Ikkinchisi esa brauzer oynani yopish vaqtida ishga tushadi.

Dasturchi tomonidan yaratiluvchi ob'yektlar. Avvalo myRecord nomli sinf yaratamiz. Hozircha unda usullar mavjud emas, ularni keyinchalik qo'shamiz. Bu sinf quyidagicha yaratiladi:

```

function myRecord(name, family, phone, address) {
  this.name = name;
  this.family = family;
  this.phone = phone;
  this.address = address;
  this.secure = false;
}
  
```

Yaratilayotgan ob'yektni xususiyatlarini ko'rsatish uchun maxsus this kalit so'zidan foydalaniladi. Bu kalit so'z ob'yektning xususiyatlariga bo'lgan murojaatini ko'rsatadi.

Keltirilgan sinfdan qanday foydalanish mumkin? Yaratilgan sinf asosida istalgan sondagi ob'yektlar yaratish mumkin. Quyida berilgan myRecord sinfida ikkita rec1 va rec2 ob'yektlari yaratilgan:

```

var rec1;
var rec2;
rec1 = new myRecord("Olim", "Umarov",
  "000-322-223", "A. Temur ko'cha, d. 225, kv. 226");
  
```



```
rec2 = new myRecord("Rustam", "Karimov",  
    "001-223-3334", "Bobur ko'cha, d. 552, kv. 662");  
rec2.secure = true;  
Ob'yektlar new operatori yordamida yaratiladi.
```

NAZORAT SAVOLLARI

1. JavaScriptda brauzer va web hujjat ob'yektlari modellari va ular bilan ishlash?
2. Standart ob'yektlar usullari va xususiyatlari?
3. Brauzer ob'yektlari iyerarhiyasi?

13-BOB. PHP ASOSLARI. O'ZGARUVCHILAR, KONSTANTALAR, MA'LUMOTLAR TURI. OPERATORLAR

13.1. PHP ilovalar komponentalari. PHP loyihalarni yaratish. PHP-kodni HTML faylga joylashtirish

Ko'p hollarda *PHP* tilini interpretatori ishlayotganligini tekshirib ko'rish uchun tuziladigan dastur eng sodda dastur deb ataladi. Hozir biz *PHP* tilidagi ushbu dasturni chuqur o'rganamiz hamda uni boshqa dasturlash tillari C, Perl va JavaScript lardan farqli tomonini tekshiramiz. Ushbu misolni ko'ramiz:

```
<html>
<head>
<title> Misol </title>
</head>
<body>
<?php
echo "<p> Salom, bu men – PHP skript!</p>";
?>
</body>
</html>
```

Bu *PHP* dasturlash tilining maxsus kodli teglari yordamida tuzilgan sodda html-fayldir.

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. Bu yerda 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 turidagi skriptlar bilan taqqoslash mumkin emas, chunki JavaScript tilidagi skriptlarda yozilgan buyruqlar faqat mijoz kompyuteridagina bajariladi. Mijoz 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* protsessori qayta ishlaydigan qilib sozlab olishingiz ham mumkin. Ya'ni mijozlar oddiy HTML-faylni qabul qildimi yoki skript natijasini ko'rdimi buni bila olmaydi. Agarda skript mijoz kompyuterida qayta ishlansa (masalan, JavaScript tilidagi dastur), u holda mijoz skript kodidan iborat HTML-sahifani ko'radi.

Biz yuqorida aytib o'tgandikki, *PHP-skriptlar HTML*-kod ichida yoziladi. Qanday qilib degan savol tug'iladi. Buning bir nechta usullari mavjud. Bulardan biri birinchi misolda keltirilganidek, `<?php` tegi bilan boshlanib `?>` tegi bilan tugagan sintaksis. Bunday ko'rinishdagi maxsus teglar HTML va *PHP* rejimidagina ishlatiladi. Bu sintaksis *PHP* tilini *XML* hujjatlari bilan birgalikda ishlaydigan dasturlarida juda ma'qul ko'riladi (masalan, XHTML tilida yozilgan dasturlarda). Biroq ba'zan quyidagi alternativ variantdan foydalansa ham bo'ladi (echo "Some text" buyrug'i «Some text» matnini ekranga chiqaradi.):

1. `<? echo "Bu PHP tilida`
2. `oddiy qayta ishlashning`
3. `instruksiyasi"; ?>`
4. `<script language="php">`
5. `echo "Bir nechta redaktorlar`
6. `(FrontPage) quyidagicha`
7. `qabul qilishadi";`
8. `</script>`
9. `<% echo " ASP texnologiyasidagi tegdan`
10. `ham foydalansa bo'ladi"; %>`

Bu keltirilgan usullardan birinchisi har doim ham bajarilavermaydi. Undan foydalanish uchun qisqa teglarni ishlatish kerak, yoki *PHP3* uchun `short_tags()` funksiyani ishlatish kerak, yoki *PHP* tilining konfiguratsion fayliga `short_open_tag` buyruqni o'rnatish kerak, yoki *PHP* dasturlash

tilida *enable-short-tags* parametr bilan kompilyatsiya qilish kerak. Agarda *php.ini-dist* buyruqqa yuqoridagilar avtomatik qo'shilgan bo'lsa, u holda qisqa teglardan foydalanish tavsiya etilmaydi. Ikkinchi usul xuddi o'rniga qo'yishga o'xshaydi, masalan, JavaScript kodlari va uning uchun mos html teglar. Shuning uchun undan har doim foydalanish mumkin, lekin bu noqulayligi uchun kamdan-kam ishlatiladi. Uchinchi usuldan faqat ASP texnologiyasidagi teglar *asp_tags* konfiguratsiyasida ishlatilgandagina foydalaniladi.

PHP dasturlash tili faylni qayta ishlayotganda u oddiy matnni *PHP* kod interpretatsiya qilishi kerak bo'lgan maxsus teglarni uchratmaguncha qaytarib beradi. Interpretator haqida gapirganda u topilgan barcha kodni yopiladigan teggacha bajaradi, so'ng yana oddiy matn qaytariladi. Bu mexanizm *PHP*-kodni HTML sahifaga aylantiradi, ya'ni barcha *PHP* teglardan tashqari barcha matnlarni o'zgarishsiz saqlaydi va ichkaridagilarni esa interpretatsiyalaydi. Yana shuni aytish kerakki, *php-fayl CGI*-skriptga o'xshamaydi.

*php-fayl*ni serverda qayta ishlash uchun jo'natishda server tomonidan brauzer satrida bu faylni yo'lini ko'rsatish shart. *PHP* skriptlar *www* orqali kirishga ruxsat etilgan joyda joylashishi shart. Agarda *php-fayl* lokal kompyuterda mavjud bo'lsa, u holda uni buyruqlar satri interpretatori yordamida qayta ishlash mumkin.

13.2. PHP asosiy sintaksislar

Instruksiyani bir nechta qismga bo'lib ko'rib chiqamiz, ya'ni kommentariylar yaratish, o'zgaruvchilar, o'zgarmlar va ma'lumot turlari, operatorlarga. Biz endi *PHP* dasturlash tilining asosiy sintaksis elementlarini o'rganishga o'tamiz. Misol sifatida elektron maktub tayyorlash masalasini ko'rib o'taylik. Uning ma'nosi quyidagidan iborat.

Faraz qilamizki, sizda qandaydir e'lon va e'lonni jo'natishingiz kerak bo'lgan bir nechta odamlar mavjud bo'lsin. Buning uchun siz e'lonni ichida o'zgaradigan (qabul qiluvchi bilan bog'liq bo'lmagan) bir nechta parametrlar mundarijasi bilan tayyorlaysiz.

Birinchi navbatda *PHP* dasturlash tili sintaksisiga nisbatan nimalarni bilish kerak. Bu HTML-kod ichiga o'rnatilgan va *PHP* dasturlash tilidagi koddir, uni interpretator farqlay biladi. Avvalgi bo'limlarda bular haqida aytib o'tgandik. Hammasini qaytarib o'tmaymiz, faqat biz ko'p hollarda misollarda `<?php ?>` variant o'rniga qisqartirilgan `<? ?>` teglardan foydalanishni aytib o'tamiz.

Instruksiyalarni ajratilishi. PHP dasturlash tilidagi dastur (ixtiyoriy dasturlash tilidagi) – bu buyruqlar (instruksiyalar) to'plamidir. Dasturni qayta ishlash uchun bir buyruqni boshqa buyruqdan farqini bilish kerak. Buning uchun maxsus simvollar – ajratgichlardan foydalaniladi. PHP dasturlash tilida instruksiyalarni xuddi C yoki Perl dasturlash tillari kabi ajratiladi, ya'ni har bir ifoda nuqtali vergul (“;”) bilan tugaydi.

«?» yopiladigan teg ham instruksiyani tugashini anglatadi, shuning uchun undan oldin nuqtali vergul qo'yilmaydi. Masalan, quyidagi ikki fragmentlar ekvivalentdir:

```
<?php
echo "Hello, world!"; // buyruqlar oxirida nuqtali vergul qo'yish
shart
```

```
?>
```

```
<?php
```

```
echo "Hello, world!" ?>
```

```
<!-- "?" borligi uchun
```

```
nuqtali vergul tashlab ketildi -->
```

Kommentariylar (izohlar). Ko'p hollarda dastur tuzganda kodni tushunarli bo'lishi uchun unga qandaydir izoh-kommentariylar qo'yish kerak bo'lib qoladi. Bu holat katta hajmdagi dasturlar yaratganda hamda bitta dastur ustida bir nechta dasturchi ishlayotganda juda muhim. Kommentariylar dasturning kodi tushunarli bo'lishi uchun yoziladi. Bundan tashqari masalani qismlarga ajratib hal qilinganda ishning kamchiligi bor joyida keyinchalik esdan chiqmasligi uchun kommentariya yozib qo'yiladi. Barcha dasturlash tillarida dastur ichiga kommentariya qo'shish imkoniyati mavjud. *PHP* dasturlash tili bir qancha ko'rinishdagi kommentariylarni qo'llab quvvatlaydi: C, C++ dasturlash tillari stilidagi hamda Unix qobig'idagi kommentariylar. // va # belgilar bir satrli kommentariylarni anglatadi, /* va */ belgilar esa mos ravishda ko'p satrli kommentariylarning boshlanish va tugashini anglatadi.

Misol: PHP dasturlash tilida kommentariyning qo'llanilishi

```
<?php
```

```
echo "Mening ismim Alisher";
```

```
// Bu bir satrli kommentariy
```

```
// C++ dasturlash tili stilidagi
```

```
echo "Mening familiyam Boliyev";
```

```
/* Bu ko'p satrli kommentariy. Bu yerga bir qancha satr yozish
mumkin. Dastur bajarilish jarayonida bu yerdagi barcha yozuvlar
(kommentariylangan), o'qilmaydi. */
```

```
echo "Men PHP dasturlash tilini INTUIT.ru dan o'rganmoqdaman";  
# Bu kommentariy  
# Unix qobig'idagi kommentariy. ?>
```

13.3. PHP da o'zgaruvchilar, o'zgarmaslar va operatorlar

Har bir dasturlash tilida muhim elementlardan biri bu *o'zgaruvchilar*, *o'zgarmaslar* va ular qo'llaniladigan *operatorlar*dir. PHP dasturlash tili elementlarni qanday belgilashi va qayta ishlashini ko'rib chiqamiz.

O'zgaruvchilar. *PHP* dasturlash tilida *o'zgaruvchilar* oldiga dollar belgisi (“\$”) qo'yib e'lon qilinadi, masalan, \$my_var.

O'zgaruvchilar nomlari registrnlarni farqlaydi, ya'ni \$my_var hamda bosh harfli \$My_var o'zgaruvchilari turli xil o'zgaruvchilardir.

PHP dasturlash tilida o'zgaruvchilar nomi qolgan dasturlash tillari qoidalari kabi e'lon qilinadi: o'zgaruvchi nomi lotin alfaviti bilan boshlanishi va undan keyin harflar yoki tagiga chizilgan belgi yoki raqamlar bo'lishi mumkin.

PHP4 dasturlash tilida bulardan tashqari o'zgaruvchiga qiymat o'zlashtirishning yana bir usuli mavjud: Xavola (ssilka) bo'yicha *o'zlashtirish*. Xavola (ssilka) bo'yicha o'zgaruvchiga qiymat o'zlashtirish uchun uning nomi bo'lishi shart, ya'ni u qandaydir o'zgaruvchini taqdim etishi kerak. Bir o'zgaruvchi qiymatini boshqa o'zgaruvchiga xavola (ssilka) bo'yicha *o'zlashtirish* uchun birinchi o'zgaruvchi oldiga ampersand & belgisi qo'yish shart. Bunga yuqoridagi misolni ko'rib chiqamiz, faqat first o'zgaruvchi second o'zgaruvchiga xavola (ssilka) bo'yicha o'zlashtiriladi:

Misol. Xavola (ssilka)lar bo'yicha o'zlashtirish.

```
<?php  
$first = ' Text ' ; // $first o'zgaruvchiga  
        // ' Text ' qiymat o'zlashtirildi  
$second = &$first;  
/* $second.orqali $first o'zgaruvchiga xavola (ssilka) qilamiz  
   Endi bu o'zgaruvchilar qiymatlari  
   har doim tengdir */  
// $first o'zgaruvchi qiymatini  
// ' New text ' qiymatga o'zgartiramiz  
$first = ' New text ' ;  
echo "first nomli o'zgaruvchi qiymati$first ga teng <br>";  
// $second o'zgaruvchi qiymatini ekranga chiqaramiz
```

echo "second nomli o'zgaruvchi qiymati " . "\$second ga teng";

?>

Bu skriptni natijasi esa quyidagicha bo'ladi:

- first nomli o'zgaruvchi qiymati New text ga teng.
- second nomli o'zgaruvchi qiymati New text ga teng.

Ya'ni \$first o'zgaruvchi qiymati o'rniga \$second o'zgaruvchi qiymati o'zlashtirildi.

O'zgarmlar. Skript bajarilish jarayonida o'zgarilmagan qiymatli kattaliklarni saqlash uchun *o'zgarmlardan* foydalaniladi. Bunday kattaliklar matematik o'zgarmlar, parollar, fayllarning yo'llari va boshqalar bo'lishi mumkin. O'zgarmlarning o'zgaruvchilardan asosiy farqi shuki, ularni faqat bir martagina o'zlashtiriladi va uni qiymatini e'lon qilingandan keyin bekor qilib bo'lmaydi. Bundan tashqari o'zgarmlar oldida dollar belgisi qo'yilmaydi hamda unga oddiy qiymat o'zlashtirish kabi qarash mumkin emas. O'zgarmlar qanday aniqlanadi? Buning uchun maxsus define() funksiyasi mavjud, uning sintaksisi quyidagichadir:

```
define ("O'zgarmlar nomi", "O'zgarmlar qiymati",  
[registrga_sezgirli_kichik])
```

O'zgarmlar nomi registrga sezgirli katta. Har bir o'zgarmlarda uni o'zgartirish mumkin, ya'ni *registrga_sezgirli_kichik* argumentni qiymati sifatida True qiymati ko'rsatiladi. O'zgarmlar nomi har doim katta registr bilan yoziladi.

O'zgarmlarni qiymatini bilish uchun uning nomini ko'rsatish kerak. O'zgaruvchidan farqi o'zgarmlar nomi oldiga \$ belgi qo'yilmaydi. Bundan tashqari o'zgarmlarni qiymatini bilish uchun konstanta nomi bilan parametr sifatida constant() funksiyasidan foydalanish mumkin.

Misol. PHP dasturlash tilida o'zgarmlar.

```
<?php
```

```
// o'zgarmlarni aniqlaymiz PASSWORD
```

```
define("PASSWORD","qwerty");
```

```
// registrlanmagan PI o'zgarmlarni qiymatini aniqlaymiz 3.14
```

```
define("PI","3.14", True);
```

```
// PASSWORD o'zgarmlar qiymatini olamiz, ya'ni qwerty
```

```
echo (PASSWORD);
```

```
// bu ham qwertyni chiqaradi
```

```
echo constant("PASSWORD");
```

```
echo (password);
```

```
/* password ni chiqaradi va biz registrlangan o'zgarmlar  
PASSWORD ni kutgandik.*/
```

echo pi;
 // 3.14 ni chiqaradi, chunki o'zgarma PI registrlanmagan va aniqlangan.

?>

Dasturchi tomonidan o'zgaruvchilardan tashqari yuqorida aytib o'tganimizdek *PHP* dasturlash tilida mavjud o'zgarma ham interpretator tomonidan aniqlanadi. Masalan, `__FILE__` o'zgarma dastur bajarilish jarayonida fayl nomini (va fayl yo'lini), `__FUNCTION__` funksiya nomidan tashkil topadi, `__CLASS__` - sinf nomi, `PHP_VERSION` – *PHP* dasturlash tili interpretatori versiyasini o'zida saqlaydi. Bunday o'zgarimalarning barcha ro'yxatini *PHP* dasturlash tili uchun mo'ljallangan qo'llanmalardan topish mumkin.

Amallar. O'zgaruvchilar, o'zgarma va ifodalar ustida turli hisoblashlarni bajaradigan bu *amallar*dir. Biz hali bu ifodalar haqida to'xtalib o'tganimiz yo'q. Ifodalar qiymatini ushbu amallar yordamida aniqlanadi. O'zgaruvchilar va o'zgarma – bu ifodalarning asosiy va juda sodda shaklidir. Shunday ifodalarni ko'paytirish mumkin bo'lgan amallar to'plami mavjud. Ularni quyida to'liqroq muhokama qilamiz:

13.1-jadval. Arifmetik amallar.

Belgilanishi	Nomlanishi	Misol
+	Qo'shish	$\$a + \b
-	Ayirish	$\$a - \b
*	Ko'paytirish	$\$a * \b
/	Bo'lish	$\$a / \b
%	Bo'lishdagi qoldiq	$\$a \% \b

13.2-jadval. Satrli amallar.

Belgilanishi	Nomlanishi	Misol
	<i>Konkatenatsiya (satrlarni qo'shish)</i>	$\$c = \$a . \$b$ (bu $\$c$ satr $\$a$ va $\$b$ satrlardan iborat)

13.3-jadval. O'zlashtirish amallari.

Belgilanishi	Nomlanishi	Izox	Misol
=	O'zlashtirish	Operatoridan o'ng tomonda turgan o'zgaruvchilar ustida bajarilgan amallardan hosil bo'lgan natija qiymati o'zlashtiriladi.	$a = (b = 4) + 5;$ (a ga teng, b ga teng)
+=		Qisqartirish. O'zgaruvchiga son qo'shiladi va keyin natija o'zlashtiriladi.	$a += 5;$ ($a = a + 5$ ifodaga ekvivalent;)
.=		O'zlashtirish va konkatenatsiya amallari kombinatsiyasini qisqartirilgan shakl (dastavval satrlar qo'shiladi, so'ngra hosil bo'lgan satr o'zgaruvchiga o'zlashadi).	$b = "Hammaga ";$ $b .= "salom";$ ($b = b . "salom"$ ifodaga ekvivalent;) Natijasi: $b="Hammaga salom"$

13.4-jadval. Matiqiy amallar.

Belgilanishi	Nomlanishi	Izox	Misol
And	VA	a va b rostd (True)	a and b
&&	VA		a && b
Or	YoKI	a yoki b o'zgaruvchilardan hech bo'lmaganda bittasi rostd bo'lsa (ikkalasi ham rostd bo'lishi mumkin).	a or b
	YoKI		a b
xor	Inversiya YoKI	O'zgaruvchilardan bittasi rostd bo'lsa. Agarda ikkalasi ham rostd bo'lsa inversiyalanadi.	a xor b
!	Inversiya (NOT)	Agarda $a=True$, u holda $!a=False$ va aks holda teskarisi bo'ladi.	$!a$

13.5-jadval. Taqqoslash amallari.

Belgilanishi	Nomlanishi	Izox	Misol
--------------	------------	------	-------

==	Tenglik	O'zgaruvchilar qiymatlari teng	\$a == \$b
===	Ekvivalentlik	O'zgaruvchilar qiymatlari va turlari teng	\$a === \$b
!=	Tengsizlik	O'zgaruvchilar qiymatlari teng emas	\$a != \$b
<>	Tengsizlik		\$a <> \$b
!==	Noekvivalentlik	O'zgaruvchilar ekvivalent emas	\$a !== \$b
<	Kichik		\$a < \$b
>	Katta		\$a > \$b
<=	Kichik yoki teng		\$a <= \$b
>=	Katta yoki teng		\$a >= \$b

13.6-jadval. Inkrement va dekrement amallari.

Belgilanishi	Nomlanishi	Izox	Misol
++\$a	Pre- <i>inkrement</i>	\$a qiymati birga oshiriladi va \$a qiymati qaytariladi	<pre><? \$a=4; echo "4 bo'lishi shart:" . \$a++; echo "6 bo'lishi shart:" . ++\$a; ?></pre>
\$a++	Post- <i>inkrement</i>	\$a qiymati qaytariladi va so'ngra \$a qiymati birga oshiriladi	
--\$a	Pre- <i>dekrement</i>	\$a qiymati birga kamaytiriladi va \$a qiymati qaytariladi	
\$a--	Post- <i>dekrement</i>	\$a qiymati qaytariladi va so'ngra \$ aqiymati birga kamaytiriladi	

13.4. PHP da ma'lumotlar turlari

PHP dasturlash tili sakkizta sodda *ma'lumot turlarini* qo'llab quvvatlaydi:

To'rttasi skalyar turlar:

- *boolean* (mantiqiy);
- *integer* (butun);

- *float* (nuqtasi siljiydigan);
- *string* (satrli).

Ikkitasi aralash turlar:

- *array* (massiv);
- *object* (ob'yekt).

Ikkitasi maxsus turlar:

- *resource* (resurs);
- *NULL*.

PHP dasturlash tilida o'zgaruvchilar turlari oshkora e'lon qilinmaydi. Ko'pincha o'zgaruvchi qo'llanilgan kontekstdan, ya'ni o'zgaruvchiga o'zlashtirilgan qiymat turidan mustaqil ravishdagi dastur bajarilish jarayonidan interpretator o'zi bu ishni bajaradi. Quyida yuqorida sanab o'tilgan *ma'lumotlar turlarini* birma-bir ko'rib chiqamiz.

Boolean turi (Bul yoki mantiqiy tur). Bu sodda tur qiymatni rost ekanligini ifodalaydi, ya'ni o'zgaruvchi faqat ikkita qiymat qabul qiladi – rost TRUE yoki yolg'on FALSE.

Mantiqiy turlarni aniqlash uchun TRUE yoki FALSE kalit so'zlaridan foydalanamiz. Bu ikkala turlar registrlanmagan.

Misol. Mantiqiy tur.

```
<?php
$test = True;
?>
```

Mantiqiy turlar turli *boshqariladigan konstruksiyalarda* (sikllar, shartlar va shunga o'xshash qo'llaniladi. Bir qancha amallar (masalan, tenglik amali) ham mantiqiy tur qabul qilishi mumkin, ya'ni faqat ikki qiymat rost yoki yolg'on qiymatni qabul qiladi. Ular *boshqariladigan konstruksiyalarda* shartlarni tekshirish uchun qo'llaniladi. Masalan, shartli konstruktorda amallar yoki o'zgaruvchilar qiymati haqiqiylikini tekshiradi va natijadan qat'iy nazar shu yoki boshqa amallarni bajarilishini tekshiradi. Bu yerda shart rost yoki yolg'on bo'lishi mumkin, chunki *mantiqiy tur amallari* va *o'zgaruvchilar* ko'rsatilgan.

Misol. Mantiqiy turlarning qo'llanilishi.

```
<?php
// '==' amal tenglikka tekshiradi mantiqiy qiymatni qaytaradi
if ($know == False) { // agar $know qiymat false bo'lsa
echo "PHP dasturlash tilini o'rgan!";
}
if (!$know) { // xuddi yuqoridagidek $know qiymati false bo'ladi
echo " PHP dasturlash tilini o'rgan!";
```

```
}
```

/* == amal \$action o'zgaruvchi qiymati bilan "PHP dasturlash tilini o'rganish!" satrni ustma-ust tushishini tekshiradi. Agar ustma-ust tushsa true qiymatni qaytaradi, boshqa holda false ni qaytaradi. Agar true ni qaytarsa figurali qavs ichidagi amallar bajariladi. */

```
if ($action == " PHP dasturlash tilini o'rganish ")
{ echo "O'rganishni boshladim";}
?>
```

Integer (butun) turi. Bu tur butun sonlar to'plamidan $Z = \{ \dots, -2, -1, 0, 1, 2, \dots \}$ birini qaytaradi. Butun sonlar xohishga qarab oldiga «-» yoki «+» belgilarni qo'yib sanoq tizimining o'nlik, o'n oltilik yoki sakkizlik tizimlarida ko'rsatilgan bo'lishi mumkin.

Agar siz sakkizlik sanoq tizimidan foydalanayotgan bo'lsangiz, oldindan 0 (nol) raqamini ko'rsatishingiz kerak. O'n oltilik sanoq tizimida esa raqamlar oldiga 0x belgini qo'yish shart.

```
<?php
# o'nlik raqam
$a = 1234;
# manfiy son
$a = -123;
# sakkizlik son (o'nlik tizimidagi
# 83 songa ekvivalent)
$a = 0123;
# o'n oltilik son (o'nlik tizimidagi
# 26 songa ekvivalent)
$a = 0x1A;
?>
```

Butun sonni o'lchami platformaga bog'liq, lekin qoidaga ko'ra maksimal qiymati ikki milliard (bu ishorali 32 bitli qiymat) atrofida bo'ladi. Ishorasiz *butun sonni PHP* dasturlash tili qo'llab quvvatlamaydi.

Agar siz *butun son* chegarasidan tashqari biror qiymat bersangiz interpretator bu sonni *qo'zg'aluvchan vergulli songa* o'zgartiradi. Xuddi shunday *butun son* chegarasidan tashqari chiqib ketadigan biror amal bajarsangiz ham bu sonni *qo'zg'aluvchan vergulli songa* o'zgartiriladi.

PHP dasturlash tilida butun sonlarni bo'lish amali mavjud emas. $1/2$ ifoda qiymati *qo'zg'aluvchan vergulli son* 0.5 ga teng. Siz natijangizni butun turga standart qoida asosida yoki round() funksiyasidan foydalangan taqdirda o'zgartirishingiz mumkin. O'zgaruvchini aniq bir turga o'zgartirish uchun uning oldiga qavs ichida kerakli turni yozish kerak

bo'ladi. Masalan, $\$a=0.5$ o'zgaruvchini butun turga o'zgartirish uchun (integer) (0.5) yoki (integer) $\$a$ ko'rinishda yoki qisqartirilgan (int) (0.5) ko'rinishda yozish kerak bo'ladi. Bunday oshkora yangi turga o'tish imkoniyati barcha *ma'lumotlar turlari* uchun o'rinli bo'ladi (albatta, har doim ham qiymatni bir turdan boshqasiga olib o'tish shart emas). Biz keltirilgan barcha turlarni chuqur o'rganishimiz shart emas, chunki *PHP* dasturlash tili kontekstdan mustaqil ravishda o'zi bu ishlarni bajaradi.

Float (qo'zg'aluvchan vergulli son) turi. *Qo'zg'aluvchan vergulli sonlar* (ular ikki karra aniqlik yoki haqiqiy sonlardir) quyidagi sintaksislar yordamida aniqlanishi mumkin:

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

Qo'zg'aluvchan vergulli sonni o'lchami ham platformaga bog'liq, lekin qoidaga ko'ra maksimal qiymati $\sim 1.8e308$ aniqlik bilan 14 xonali raqam atrofida bo'ladi.

Resource (resurslar) turi. *Resurs* – bu tashqi resursga (masalan, ma'lumotlar bazasi bilan bog'lanish) xavola (ssilka) orqali bog'langan maxsus o'zgaruvchidir. Resurslar maxsus funksiyalar (masalan, `mysql_connect()`, `pdf_new()` va shunga o'xshashlar) yordamida yaratiladi va foydalaniladi.

Null turi. Maxsus *NULL* qiymati *o'zgaruvchini* qiymatga ega emasligi haqida ogohlantiradi.

O'zgaruvchi NULL qiymat qabul qiladi, agarda:

- unga *o'zgarvas NULL* ($\$var = NULL$) o'zlashtirilgan bo'lsa;
- unga hech qanday qiymat berilmagan bo'lsa;
- u *unset()* funksiya yordamida tozalangan bo'lsa.

NULL turli faqat bitta qiymat mavjud – bu registrga sezgirligi kichik *NULL* kalit so'zidir.

String turi. Satr – bu belgilar to'plamidir. *PHP* dasturlash tilida belgi (bir bayt) va 256 ta turli belgilar mavjud. *PHP* dasturlash tili Unicode turidagi belgilarni qabul qilmaydi. *PHP* dasturlash tilida amalda satrlarga chegirma mavjud emas, shuning uchun satrlarni ishlatganda uning aniq uzunligi haqida o'ylash shart emas.

PHP dasturlash tilida satrlar uchta turli xil usullarda aniqlanadi:

- *bittalik qo'shtirnoqlar* yordamida ("");

- *qo'shtirnoqlar* yordamida (“”);
- *heredoc-sintaksisi* yordamida.

Bittali qo'shtirnoqlar

Satrlarni aniqlashning oddiy usuli – u «'»*bittali qo'shtirnoqlar* ichida yoziladi. Agarda satr ichida ham bittali qo'shtirnoq ishlatishga to'g'ri kelib qolsa, bittali qo'shtirnoqdan oldin «\» belgini qo'yish shart. Agarda «\» belgi bittali qo'shtirnoqdan oldin yoki satrning oxirida bo'lsa, u holda belgini ikkilantirish kerak, ya'ni «\\'».

Agarda bittali qo'shtirnoq ichidagi satr ichida ixtiyoriy belgidan oldin («\» va «'» lardan farqli ravishda) teskari slesh «\» belgisi uchrasa, u holda uni oddiy belgi deb qarab barcha belgilarni o'z holicha ekranga chiqaradi. Shuning uchun teskari slesh «\» belgisini satr oxirida yopiladigan qo'shtirnoqdan avval qo'yish shart.

PHP dasturlash tilida teskari slesh «\» belgisi bilan ifodalanadigan bir qator belgilar majmui mavjud. Ularni *ketma-ketlikni boshqaruvchilar* deb ataladi hamda ular maxsus vazifalarni bajaradi. O'zgaruvchilar va ketma-ketlikni boshqaruvchilar bittalik qo'shtirnoqlar satri ichida uchrasa, ular o'rtasidagi farq, *ketma-ketlikni boshqaruvchilar qayta ishlanmaydi*.

```
<?php
echo 'Satrlar majmui';
// Ekranga chiqaradi: ' belgini chiqarish uchun undan oldin \ belgi
qo'yiladi.
echo ' Belgini \' chiqarish uchun undan oldin'
  '\ belgini qo'yish kerak';
// Ekranga chiqaradi: Siz shuni o'chirmoqchimisiz C:\*.*?
echo ' Siz shuni o'chirmoqchimisiz C:\\*.*?';
// Ekranga chiqaradi: Buni qo'ymang: \n
// yangi qatorga
echo ' Buni qo'ymang: \n yangi qatorga ';
// Ekranga chiqaradi: o'zgaruvchi $expand ham
// $either qo'yilmaydi
echo 'o'zgaruvchi $expand ham $either' .
  'qo'yilmaydi';
?>
```

Masalaning yechilishi. Endi bo'limning boshida qo'yilgan masalaga qaytsak. U turli sabablar bo'yicha har xil odamlarga tuzilgan maktubni jo'natishdan iborat edi. Bu masalani hal etish uchun o'rganilgan vositalardan – *o'zgaruvchilar, amallar, o'zgarmaslar, satrlar* va *massivlardan* foydalanishga harakat qilamiz. Ko'rsatilgan maktub qabul

qiluvchiga bog'liq ravishda murojaat va holati o'zgaradi, shuning uchun tabiiy ravishda bu kattalikni *o'zgaruvchi* deb belgilaymiz. Bundan tashqari hodisalar va odamlar ko'p, shuning uchun *massiv o'zgaruvchi turidan* foydalanish qulay. Maktub matni har doim o'zgarmas, shuning uchun uni *o'zgarmas* deb berish maqsadga muvofiqdir. Juda uzun va qo'pol satrlarni yozmaslik uchun satrlar *konkatenatsiya* (qo'shish) amalidan foydalanamiz. Shunday qilib, quyidagiga ega bo'lamiz:

```
<?
// bizning yozuvimiz
// o'zgarmas bo'lsin.
define("SIGN","Hurmat bilan, Azamat");
// odamlar va hodisalar massivini beramiz
$names = array("Umarov Ikrom",
               "Odil Botirov",
               "Alisher Xakimov");
$events = array(
    "f" => "ochiq eshiklar kuni",
    "o" => "ko'rgazmaning ochilishi",
    "p" => "bitiruvchilar bali");
// taklifnoma matnini tuzamiz.
$str = "Hurmatli, $names[0]";
$str .= "<br> Sizni taklif etamiz ".
$events["f"];
$str .= "<br>" . SIGN;
echo $str; // matnni ekranga chiqaramiz.
?>
```

13.5. PHP da massivlar

Array (massiv) turi. *PHP* dasturlash tilida *massiv* turi tartiblangan kartalarga o'xshaydi va qiymatini kalitga o'zlashtiradigan turdir. Bu tur bir necha yo'nalishlarda optimallashtiriladi, shuning uchun siz uni xususiy *massiv*, ro'yxat (vektor), xesh-jadvali (kartani amalga oshirish uchun ishlatiladi), stek, navbat va boshqalar sifatida foydalanishingiz mumkin. Modomiki, *PHP* dasturlash tilida bir massivni qiymatini boshqasiga o'zlashtirish uchun daraxtlardan foydalanasiz.

Massivlarni array() konstruksiyasi yordamida aniqlanadi yoki elementlariga qiymat berish bilan aniqlanadi.

array() konstruksiyasi yordamida aniqlash.

array ([key] => valuye,

[key1] => valuye1, ...)

PHP dasturlash tilining *array()* konstruksiyasi vergul bilan ajratilgan juft parametrlar *kalit* =>*qiymat* bilan ajratilgan. => belgi mos ravishda qiymat va uning kaliti o'rtasida aloqa o'rnatadi. Kalit butun son bo'lishi mumkin, uning qiymati esa *PHP* dasturlash tilidagi ixtiyoriy turni qabul qilishi mumkin. Kalit raqamini biz ko'pincha indeks deb ataymiz. *PHP* dasturlash tilida indekslash noldan boshlanadi. Massiv elementining qiymatini olish uchun massiv nomi va kvadrat qavs ichida uning kaliti ko'rsatilishi kerak. Agar massiv kaliti standart butun son bo'lsa, u holda uning qiymatini butun son deb qarasa bo'ladi, aks holda u satr deb qaraladi. Shuning uchun `$a["1"]` yozuv `$a[1]` yozuvga teng kuchli, `$a["-1"]` yozuv esa `$a[-1]` yozuvga teng kuchli.

Misol. *PHP* dasturlash tilida massivlar.

```
<?php
$books = array ("php" =>
    "PHP users guide",
    12 => true);
echo $books["php"];
//ekranga chiqaradi: "PHP users guide"
echo $books[12]; //ekranga chiqaradi: 1
?>
```

Agarda element uchun kalit berilmagan bo'lsa, u holda kalit sifatida kalitning maksimal qiymatiga bir qo'shib hisoblanadi. Agarda qiymati mavjud kalit ko'rsatilgan bo'lsa, u holda shu kalit qiymatini ekranga chiqaradi. *PHP* 4.3.0 dasturlash tili versiyasidan boshlab kalitning maksimal qiymati manfiy son deb qaralsa, u holda massivning keyingi kaliti nol (0) bo'ladi.

Misol. *PHP* dasturlash tilida massivlar.

```
<?php
// $arr hamda $arr1 massivlar ekvivalentdir.
$arr = array(5 => 43, 32, 56, "b" => 12);
$arr1 = array(5 => 43, 6 => 32,
    7 => 56, "b" => 12);
?>
```

Agarda `TRUE` yoki `FALSE` kalit sifatida qo'llanilsa, u holda uning qiymati mos ravishda *integer* turining bir va noliga o'zlashtiriladi. Agar `NULL` dan foydalanilsa, u holda kalit o'rniga bo'sh satr hosil bo'ladi. Bu bo'sh satrni kalit sifatida foydalansa bo'ladi, ammo uni qo'shtirnoqqa olish

kerak bo'ladi. Bu usul bo'sh kvadrat qavsni ishlatish kabi emas. Massivlar yoki ob'yektlar kaliti sifatida foydalanish mumkin ham emas.

To'rtburchak qavs sintaksisi yordamida aniqlash. Massivga qiymat berish orqali massiv yaratish mumkin. Biz yuqorida aytib o'tganimizdek, massiv elementi qiymatiga ega bo'lish uchun kvadrat qavs ichiga uning kaliti ko'rsatilishi kerak, masalan, `$book["php"]`. Agarda yangi kalit va yangi qiymat ko'rsatsangiz quyidagicha bo'ladi: `$book["new_key"]="new_valuye"` hamda massivga yangi element qo'shiladi. Agarda kalitni ko'rsatmay faqat qiymatni o'zlashtirsak, ya'ni `$book[]="new_valuye"`, u holda massivga yangi element qo'shiladi va uni kaliti mavjud maksimal qiymatga bir qo'shiladi. Agarda biz qiymat bergan massiv yaratilmagan bo'lsa, u holda biz qiymat bergandan keyin u yaratiladi.

```
<?
```

```
$books["key"]= valuye; // key kaliti bilan valuye qiymat $books massivga qo'shiladi
```

```
$books[] = valuye1; /* 13-kalit bilan valuye1 qiymati massivga qo'shiladi, chunki bizda kalitning maksimal qiymati 12 edi. */
```

```
?>
```

Massivning aniq bir elementini o'zgartirish uchun uning shu kaliti bilan yangi qiymatga o'zlashtirish kerak. Massiv elementi kalitini o'zgartirish mumkin emas, faqat o'chirish (kalit va elementi juftligini) va yangi qo'shish mumkin holos. Massiv *elementini o'chirish* uchun `unset()` funksiyasidan foydalanish kerak.

```
<?php
```

```
$books = array ("php" =>"PHP users guide",12 => true);
```

```
$books[] = "Book about Perl"; /* 13-kalit (indeks) bilan yangi element qo'shildi, bu quyidagiga ekvivalent $books[13] = "Book about Perl";
```

```
$books["lisp"] = 123456; /* Bu massivga yangi "lisp" kalitli 123456 qiymali yangi element qo'shish*/
```

```
unset($books[12]); // Bu 12-kalitli elementni massivdan o'chirish
```

```
unset ($books); // massivni butunlay o'chirish
```

```
?>
```

Bo'sh kvadrat qavsdan foydalanganda kalitning maksimal qiymati massivda mavjud oxirgi qayta indekslangan kalitlar orasidan qidiriladi. Massivni `array_valuyes()` funksiyasi yordamida *qayta indekslash* mumkin.

Misol. Massivni qayta indekslaymiz.

```
<?php
```

```

$arr =
array ("a","b","c");
/* "a", "b" va "c" qiymatli massivni yaratamiz. Bu yerda kalit
ko'rsatilmagan biroq mos ravishda ular 0,1,2 bo'ladi. */
print_r($arr); // massivni ekranga chiqaramiz (kaliti va qiymatini)
unset($arr[0]);
unset($arr[1]);
unset($arr[2]);
// massivdan hamma elementini o'chiramiz
print_r($arr); // massivni ekranga chiqaramiz (kaliti va qiymatini)
$arr[] = "aa"; // massivga yangi element qo'shamiz. Uni indeksi
(kaliti) 3 bo'ladi, 0 emas.
print_r($arr);
$arr =
array_valuyes($arr); // massivni qayta indekslaymiz.
$arr[] = "bb"; // bu elementni kaliti 1 bo'ladi.
print_r($arr);
?>

```

Bu skriptning natijasi quyidagicha bo'ladi:

```
Array ( [0] => a [1] => b [2] => c )
```

```
Array ( )
```

```
Array ( [3] => aa )
```

```
Array ( [0] => aa [1] => bb )
```

NAZORAT SAVOLLARI

1. PHP tili asoslari va uning imkoniyatlari?
2. PHP tili asosiy sintaksislari?
3. PHP kodni HTML hujjatga joylashtirish?
4. PHP tilida o'zgaruvchilar, konstantalar, ma'lumotlar turlari va operatorlar bilan ishlash va ularni e'lon qilish?
5. PHP tilida massivlar va ularni e'lon qilish?

14-BOB. PHP FUNKSIYALAR VA OB'YEKTLAR

14.1. PHP standart funksiyalar

Funksiyalar nima uchun kerak? Bu savolga javob berish uchun, funksiya o'zi nima ekanligini tushunib olish lozim bo'ladi. Dasturlashda, xuddi matematikadagi kabi, unga bog'liq ko'pgina argumentlarning ko'pgina ma'nolarida aks etishidir. Demak, funksiya argumentning har bir ma'nolari jamlanmasi uchun uning bajargan ishi natijasi sifatida qandaydir ma'no qaytaradi. Funksiyalar nima uchun kerak, buni misollar bilan oydinlashtirishga harakat qilamiz. Dasturlashdagi funksiyaga misol – bu sonning faktorial qiymatini hisoblab beruvchi funksiya. Demak, biz unga son beramiz, u esa bizga uning faktorialini qaytaradi. Bunda biz faktorialini olishni xohlagan har bir son uchun aynan bir xil kodni qaytaravermaymiz – bu songa teng bo'lgan argumentli funksiyani chaqirishning o'zi kifoya qiladi.

Natural son faktorialini hisoblash funksiyasi

Misol:

```
<?php
function fact($n){
if ($n==0) return 1;
else return $fact = $n * fact($n-1);
}
echo fact(3);
// echo (3*2) deb yozish mumkine di; lekin son kata bo'lsa,
echo fact(50);
// funksiyadan foydalanish qulayroq, echo (50*49*48*...*3*2) deb
yozgandan;
?>
```

Shu yo'l bilan biz biron-bir ma'lumotga bog'liqlik zaruriyati tug'ilgan amalni bajarganimizda, bu holda ham biz aynan shunday amallarni bajarishimiz lozim bo'ladi, faqat boshqa boshlang'ich ma'lumotlardan foydalanamiz, funksiyalar mexanizmidan foydalanish – funksiya tanasi ko'rinishidagi amallar blokini taxt qilish, o'zgaruvchan ma'lumotlarni esa – uning parametrlari sifatida foydalanish qulayroq bo'ladi.

Funksiya topshirig'i (e'loni) umumiy tarzda qanday bo'lishini ko'ramiz. Funksiya quyidagi sintaksis yordamida aniqlanadi:

function Funksiya_nomi (1-parametr, 2-parametr, ... N-parametr) {

```

Amallar bloki
return "funksiyaga aylanuvchi ma'no";
}

```

Agar php-dasturda to'g'ridan-to'g'ri yozilsa, hech narsani ishlab bo'lmaydi. Birinchidan, funksiya nomi funksiya parametrlari nomlari (1-parametr, 2-parametr va b.) PHP da nomlanish qoidalariga muvofiq kelishi kerak (unda yaxshisi kirillcha simvollarni ham ishlatmagan ma'qul). Funksiya nomlari registrga nisbatan sezuvchan bo'ladi. Ikkinchidan, funksiya parametrlari – tilning o'zgaruvchan qismlari, shuning uchun ularning har birining nomlari oldidan \$ belgisi turishi lozim bo'ladi. Parametrlar ro'yxatida hech qanday ko'p nuqtalarni qo'yish mumkin emas. Uchinchidan, amallar bloki so'zi bilan birga funksiya tanasida istalgan to'g'ri PHP-kod mavjud bo'lishi kerak (parametrlarga muvofiq bo'lishi majburiy emas). Va nihoyat, return kalit so'zidan so'ng tartibli php-ifoda kelishi lozim (ma'noga ega bo'lgan qandaydir simvollar). Bundan tashqari, funksiyada qaytariluvchi ma'no kabi parametrlar bo'lmasligi ham mumkin. Funksiyani to'g'ri e'lon qilishga misol – yuqorida keltirilgan faktorialni hisoblash funksiyasi.

Funksiya chaqirish qanday amalga oshadi? Funksiya nomi va yumaloq qavslar ichida uning parametrlari ma'nolari ro'yxati ko'rsatiladi, agar shundaylari mavjud bo'lsa:

Misol:

```

<?php
    Funksiya_nomi          ("1-parametr_uchun_ma'no",          "2-
parametr_uchun_ma'no ",...);
    /* Funksiyani chaqirishga misol – funksiyaning faktorialni
hisoblash yuqorida bor, 3 soni faktorialini hisoblash uchun u yerda biz
fact(3) deb yozganmiz; u yerda fact – chaqiriluvchi funksiya nomi, a 3 –$n
nomli uning parametri ma'nosi */
?>

```

Funksiyani qachon chaqirish mumkin? Bu g'alati savol bo'lib tuyulishi mumkin. Funksiyani uni aniqlangandan keyin chaqirish mumkin, ya'ni function f_name(){...} blokidan pastda istalgan dastur qatorida. PHP3 da bu aynan shunday. Lekin PHP4 da bunday talab yo'q. Hamma gap interpretator olingan kodni qanday qayta ishlashida. Birgina istisno shartli ravishda aniqlanadigan funksiyadan tashkil topadi (shartli operatorlar yoki boshqa funksiyalar ichida). Funksiya shu tarzda aniqlangan taqdirda, uni aniqlash uni chaqirishdan oldin bajariladi.

Misol. Shartli funksiya ichida funksiyaning aniqlash

```

<?
$make = true;
/* bu yerda Make_event() ni chaqirish mumkin emas; Chunki u hali
mavjud emas, lekin Save_info() ni chaqirish mumkin*/
Save_info("Sobir","Sodiqov", "Men PHP kursini tanladim");
if ($make){
// Make_event() funksiyasini aniqlash
function Make_event(){
echo "<p> Python<br> ni o'rganmoqchiman";
}
}
// endi Make_event() ni chaqirish mumkin
Make_event();
// Save_info funksiyasini aniqlanadi
function Save_info($first, $last, $message){
echo "<br>$message<br>";
echo "Ism: ". $first . " ". $last . "<br>";
}
Save_info("Murod","Yoqubov", "Men Lisp ni tanladim");
// Save_info ni bu yerda ham chaqirish mumkin
?>

```

Agar funksiya dastur ichida aniqlangan bo'lsa, uni keyin qayta aniqlash yoki o'chirib tashlash mumkin emas. Funksiya nomlariga registr ta'sir qilmasligiga qaramasdan, yaxshisi funksiyani aniqlash paytida berilgan nom bilan chaqirish mumkin bo'ladi. Misol. Funksiya ichidagi funksiyani aniqlash

```

<?php
/* ma'lumotlarni saqlash, ya'ni DataSave() funksiyasini chaqirish
mumkin emas. Uning to'g'riligi tekshirilmasdan oldin, ya'ni DataCheck()
funksiyasi chaqirilmasdan oldin bu mumkin emas.*/
DataCheck();
DataSave();
function DataCheck(){
// ma'lumotlar to'g'riligini tekshirish
function DataSave(){
// ma'lumotlarni saqlaymiz
}}?>

```

Funksiya argumentlari, ularning ma'nolari va ishlatilishini batafsil ko'rib chiqamiz.

Funksiyalar ichida o'zgaruvchilardan foydalanish

1. Global o'zgaruvchilar
2. Statistik o'zgaruvchilar
3. Qaytariluvchan ma'nolar
4. Xavola (ssilka)ni qaytarish

Global o'zgaruvchilar. Funksiyalar ichida undan tashqarida berilgan o'zgaruvchilardan foydalanish uchun, bu o'zgaruvchilarni global deb e'lon qilish kerak. Buning uchun funksiya tanasida uning nomlarini global kalit so'zidan keyin keltirish lozim bo'ladi:

```
global $var1, $var2;
<?
$a=1;
function Test_g(){
global $a;
$a = $a*2;
echo ' $a=', $a funksiya ishida natija;
}
echo funksiyadan tashqarida $a=', $a, ', ' ;
Test_g();
echo "<br>";
echo funksiyadan tashqarida $a=', $a, ', ' ;
Test_g();
?>
```

Misol. Global o'zgaruvchilar

Bu skript ishidan quyidagi natijalarni olamiz:

\$a=2 funksiyadan tashqarida, \$=2 funksiya ishida natija

\$a=2 funksiyadan tashqarida, \$=4 funksiya ishida natija

O'zgaruvchi global deb e'lon qilinganda, aniq global o'zgaruvchi uchun xavola (ssilka) tuziladi. Buning uchun bunday yozuv quyidagiga ekvivalent (GLOBALS massivi mavjud ko'rinish sohalariga muvofiq barcha global o'zgaruvchilarni o'z ichiga oladi):

```
$var1 = & $GLOBALS["var1"];
$var2 = & $GLOBALS["var2"];
Bundan kelib chiqadiki, $var1 o'zgaruvchini o'chirish
```

```
$_GLOBALS["var1"] global o'zgaruvchini o'chirib tashlamaydi.
```

Statistik o'zgaruvchilar. O'zgaruvchilardan faqat funksiya ichida foydalanish uchun bunda uning ma'nosini saqlagan holda va funksiyadan chiqqandan so'ng, bu o'zgaruvchilarni statistik deb e'lon qilish kerak. Statistik o'zgaruvchilar faqat funksiyalar ichida ko'rinadi va dasturni

yusinfh funksiya doirasidan tashqariga chiqsa o'z ma'nosini yo'qotmaydi. Bu o'zgaruvchilarni e'lon qilish static kalit so'zi yordamida amalga oshiriladi:

```
static $var1, $var2;
```

Har qanday ma'no statistik o'zgaruvchi sifatida talqin qilinishi mumkin, faqat xavola (ssilka) emas.

Misol. Statistik o'zgaruvchilardan foydalanish

```
<?
```

```
function Test_s(){
```

```
static $a = 1;
```

```
// ifoda yoki xavola (ssilka)ni o'zlashtirib bo'lmaydi
```

```
$a = $a*2;
```

```
echo $a;
```

```
}
```

```
Test_s(); // 2 chiqadi
```

echo \$a; // hech narsa chiqmaydi, zero \$a faqat funksiya ichida kirish yo'lagi bor

Test_s(); // \$a=2 funksiya ichida, shuning uchun funksiya ishi natijasi 4 soni bo'ladi

```
?>
```

Qaytariluvchan ma'nolar. Yuqorida misol qilib keltirilgan barcha funksiyalar biror-bir amal bajaradi. Bunday hollardan tashqari, har qanday funksiya o'z ishi natijasi sifatida qandaydir qiymat qaytaradi. Bu return tasdig'i yordamida amalga oshiriladi. Qaytariluvchan qiymat har qanday turda, shu jumladan, ro'yxat va ob'yektlar bo'lishi mumkin. Interpretator funksiya tanasida return komandasiga uchraganda, u darhol uni bajarishni to'xtatadi va funksiya chaqirilgan qatorga o'tib ketadi.

Masalan, inson yoshini qaytaruvchi funksiya tuzamiz. Agar inson vafot etmagan bo'lsa, yosh joriy yilga muvofiq hisoblanadi.

```
<?php
```

/* agar ikkinchi parametr true kabi hisoblab chiqilsa, u vafot etgan sana day ko'rib chiqiladi, */

```
function Age($birth, $is_dead){
```

```
if ($is_dead) return $is_dead-$birth;
```

```
else return date("Y")-$birth;
```

```
}
```

```
echo Age(1971, false); // vivedet 33
```

```
echo Age(1971, 2001); // vivedet 30
```

```
?>
```

Bu misolda return funksiyasidan foydalanmasa ham bo'ladi, shunchaki uni chiqarish funksiyasini echo ga almashtiriladi. Aksincha, agar biz funksiya biror-bir qiymat qaytaradigan qilsak (bu misolda inson yoshi), biz dasturda o'zgaruvchini bu funksiya qiymatini istalgan o'zgaruvchiga o'zlashtirishimiz mumkin.

```
$my_age = Age(1981, 2004);
```

Funksiya ishi natijasida faqat bitta qiymat qaytarilishi mumkin. Bir necha qiymatni qiymatlar ro'yxati qaytarilgan taqdirda olish mumkin (bir o'lchamli massiv). Biz inson yoshini kunigacha aniqlikda olmoqchimiz, deb hisoblaylik.

```
<?php
function Full_age($b_day, $b_month, $b_ear)
{
    $y = date("Y");
    $m = intval(date("m"));
    $d = intval(date("d"));
    $b_month = intval($b_month);
    $b_day = intval($b_day);
    $b_ear = intval($b_ear);
    $day = ($b_day > $d ? 30 - $b_day + $d : $d - $b_day);
    $tmpMonth = ($b_day > $d ? -1 : 0);
    $month = ($b_month > $m + $tmpMonth
        ? $b_month + $tmpMonth - $m : $m + $tmpMonth - $b_month);
    $tmpYear = ($b_month > $m + $tmpMonth ? -1 : 0);
    if ($b_ear > $y + $tmpYear)
    {
        $sear = 0; $month = 0; $day = 0;
    }
    else
    {
        $sear = $y + $tmpYear - $b_ear;
    }
    return array ($day,$month,$sear);
}
$age = Full_age("29","06","1986");
echo "Siz $age[2] yosh, $age[1] oylar va $age[0] kunlar";
?>
```

Funksiya bir necha qiymatlarni ularni dasturda qayta ishlash uchun qaytarganda, bir amal bilan ma'noni birdaniga bir necha o'zgaruvchilarni

o'zlashtirishga imkon beruvchi list() til konstruksiyasidan foydalanish qulay bo'ladi. Masalan, yuqoridagi misolda funktsiyani, uning qiymatiga o'zgartirish kiritmay qayta ishlash quyidagicha bo'lishi mumkin:

```
<?
// Full_age() funktsiya kiritish
list($day, $month, $ear) = Full_age("07",
  "08", "1974");
echo "Sizning yoshingiz $ear, $month oy va
  $day kun";
?>
```

list() konstruksiyasini umuman o'zgaruvchini o'zlashtirish uchun istalgan massiv elementi qiymatidan foydalanish mumkin.

Misol. list() dan foydalanish

```
<?
$arr = array("first", "second");
list($a, $b) = $arr;
  // o'zgaruvchi $a o'zlashtiriladi, birinchi massiv qiymati, $b –
ikkinchi
echo $a, " ", $b;
// «first second» qatori keltiriladi
?>
```

Xavola (ssilka)ni qaytarish. Funktsiya o'z ishi natijasida shuningdek xavola (ssilka)ni biror-bir o'zgaruvchiga qaytarishi mumkin. Bu funktsiyani qanday o'zgaruvchi xavola (ssilka)ga o'zlashtirish kerakligini aniqlash uchun foydalaniladi. Funktsiyadan xavola (ssilka) olish uchun, e'lon oldidan ampersand (&) belgisini yozish kerak bo'ladi va har safar funktsiya chaqiruvchi paytida uning nomi oldidan ham ampersand (&) yozish kerak bo'ladi. Ko'pincha funktsiya xavola (ssilka)ni biror-bir global o'zgaruvchiga (yoki uning qismini – xavola (ssilka)ni global massiv elementiga), xavola (ssilka)ni statistik o'zgaruvchiga (yoki uning qismini) yoki xavola (ssilka)ni argumentlardan biriga qaytaradi, agar u xavola (ssilka) bo'yicha berilgan bo'lsa.

Misol. Xavola (ssilka)ni qaytarish

```
<?
$a = 3; $b = 2;
function & ref($par){
global $a, $b;
if ($par % 2 == 0) return $b;
else return $a;
```

```

}
$var =& ref(4);
echo $var, " i ", $b, "<br>"; // 2 va 2 keltiriladi
$b = 10;
echo $var, " i ", $b, "<br>"; // 10 va 10 keltiriladi
?>

```

Xavola (ssilka) sintaksisidan foydalanishda bizning misoldagi \$var o'zgaruvchi, \$b o'zgaruvchining \$ref qiymati qaytarilgan funksiyasiga ko'chirilmaydi, bu o'zgaruvchiga xavola (ssilka) tuziladi. Demak, endi \$var va \$b teng kuchli o'zgaruvchilar va ular bir paytda o'zgartiriladi.

14.2. PHP da grafik ma'lumotlar va GD kutubxona

Php brauzerda faqatgina matnlarni emas grafik ob'yektlarni ham chiqaradi va yuboradi. Lekin Php xususiyatlarida grafika bilan ishlashda bir qancha noqulayliklar mavjud.

Grafik funksiyalar bilan ishlash uchun Php ixtiyoriy distributivida qo'yiladigan GD library kutubxonasi talab etiladi. Barchaga ma'lum bo'lgan: jpeg, gif, png, wbmp grafik formatlardan foydalaniladi. 1.6 versiyada GD gif formatni qo'llab quvvatlamaydi, va aksincha, faqatgina 1.8 versiyada jpeg format uchun ayrim kerakli funksiyalar qo'llab quvvatlanadi. Ya'ni, bundan kelib chiqadiki, eski versiyani qo'yib jpeg formatdan foydalanish kerak emas, va aksincha yangi versiyani qo'yib gif formatdan foydalanish kerak emas.

Shuning uchun, biz yuqorida keltirib o'tilgan sharoitlarda ham ishlaydigan funksiyalarni keltirib o'tamiz. Ya'ni gif format borligini butunlay unutib o'rniga png formatdan foydalanamiz.

Birinchi navbatda GD serverga ulanganligini tekshirib ko'rish kerak. ext.php fayl.

```

<?php
$bibl_ext=dirname($_SERVER['SCRIPT_FILENAME']).'/extensions/php_gd.dll';
if (!extension_loaded ('gd') and @is_file ($bibl_ext)){ dl
("extensions/php_gd.dll");}
if (! extension_loaded ('gd')){ print "Kutubxonada fayl yo'q
$bibl_ext";}
?>

```

Agar bu script "Kutubxonada fayl yo'q" deb chiqarsa, u holda u yerda nimalar yozilganini o'qish shart emas, chunki, bu satrlar bilan grafik

modul ulangan yoki ulanmaganligini tekshirib ko'rildi, agar yo'q bo'lsa ulashga harakat qilib ko'rildi, lekin u Php ning sizdagi versiyasida topilmagan bo'ladi. U holda, uning php_gd.dll manbasini topish kerak va php.exe bilan yonma-yon turadigan "/extensions" papkaga yozish kerak.

Agar xatolik haqidagi satrlar chiqmasa, u holda sizda grafik funksiyalarni qo'llab quvvatlovchi bor, va davom etamiz. Biz ext.php faylini quyidagi fayllar bilan bir xil papkada saqlaymiz va barcha skriptlar boshiga quyidagicha bog'laymiz: 'ext.php'.

Grafik funksiyalar senariy bajarilish vaqtida murojaat qilish mumkin bo'lgan ob'yektlarni yaratadi. Ya'ni bu grafik ob'yektlarni almashtirish, u erga matn qo'shish va boshqalar mumkin. Eng avvalo shuni tushunish kerakki, grafik ob'yekt – bu hali brauzerga chiqarish mumkin bo'lgan rasm emas. Afsuski, eng avvalo uni fayda saqlab, so'ngra chop etishga chiqarish mumkin. Dinamik dizayn uchun bu yahshi emas.

ImageCreate - grafik ob'yekt yaratish.

```
<?php
    // 200x20 o'lchamdagi grafik ob'yekt yaratish
    $im = ImageCreate (200,20);
?>
```

Grafik ob'yekt yaratib uni \$im o'zgaruvchiga o'zlashtirdik.

ImageColorAllocate – rasm uchun ranglarni belgilash. Red (qizil), green (yashil), blue (ko'k) komponentalaridan tuzilgan ranglar palitrasiga qo'shilgan indeksni qaytaradi.

```
<?php
    $im = ImageCreate (100,20);
    // qizil
    $red = ImageColorAllocate($im, 255, 0, 0);
    // oq
    $white = ImageColorAllocate($im, 255, 255, 255);
    // qora
    $black = ImageColorAllocate($im, 0, 0, 0);
?>
```

ImageString – ob'yektga gorizontaal satr qo'shish.

```
<?php
    $im = ImageCreate (100,20);
    $red = ImageColorAllocate($im, 255, 0, 0);
    // balandlik shrifti - 3, o'ng tomon 4 pixel va pastga 5 pixel, $red
    qizil rang bilan Super!!! yozuvini yozamiz
    ImageString ($im, 3, 4, 5, "Super!!!", $red);
```

?>

Endi "Super !!!" satri bizning ob'ektimizda uchraydi, unda shriftning balandligi 3, matnning o'ng tomonga - 4 va pastga siljish - 5 piksel, matn rangi \$red rang bilan boshlanadi.

Natijada:

```
<?php
```

```
header ("Content-type: image/png");
```

```
$im = ImageCreate (100,20);
```

```
$red = ImageColorAllocate($im, 255, 0, 0);
```

```
$white = ImageColorAllocate($im, 255, 255, 255);
```

```
ImageString ($im, 3, 3, 3, "Super!!!", $white);
```

```
ImagePng ($im);
```

```
ImageDestroy ($im);
```

```
?>
```

Brauzerda: Super!!! yozuvni ko'rish mumkin.

Boshqacha aytganda, header brauzerga ma'lumotlarni grafik obyekt sifatida yuborgan.

ImageCreate – 100 ga 20 pixel o'lchamli grafik ob'yekt yaratdi.

ImageColorAllocate(\$im, 255, 0, 0) – ob'yekt foni qizil.

ImageColorAllocate(\$im, 255, 255, 255) – fon oq fon.

ImageString - o'ng va pastga 3 va 3 piksellarda oq rangda Super !!!.

ImagePng – ob'yektni Png formatga o'tkazamiz va brauzerga uzatamiz.

ImageDestroy - Png tasviriga ajratilgan xotirani bo'shatish.

GetImageSize – grafik ob'yekt bo'lmagan rasm xususiyatlarini olish.

Bu funksiya aynan rasmlar bilan ishlaydi ob'yektlar bilan emas. .php faylingiz yonida "/images" papkasini yarating va u yerga .png va .jpeg formatdagi ikkita faylni yozing, masalan: 'baner.png' va 'bad.jpg'.

```
<?php
```

```
// 'bad.jpg' fayl bo'lmasa – uni yaratamiz
```

```
$img = 'bad.jpg';
```

```
if (! @file_exists ($img)){
```

```
$im = ImageCreate (200,200);
```

```
$red = ImageColorAllocate($im, 255, 0, 0);
```

```
ImageString ($im, 3, 3, 3, "Super!!!", $red);
```

```
ImageJpeg ($im,$img);
```

```
ImageDestroy ($im);
```

```
}
```

```
// 'baner.png' fayl bo'lmasa – uni yaratamiz
```

```

$img = 'baner.png';
if (! @file_exists ($img)){
$img = ImageCreate (200,200);
$bgcolor = ImageColorAllocate($im, 255, 255, 255);
$bgcolortrans = ImageColorTransparent($im, $bgcolor);
$black = ImageColorAllocate($im, 0, 0, 0);
ImageString ($im, 3, 3, 3, "Super!!!", $black);
ImagePng ($im,$img);
ImageDestroy ($im);
}
// 'bad.jpg' tasvirni qo'yamiz
$img = 'bad.jpg';
$arr = GetImageSize ($img);
if ($arr[2]==2){ print "$img <b>Jpeg file </b> <br>";}
if ($arr[2]==3){ print "$img <b>Png file </b> <br>";}
// 'baner.png' tasvirni qo'yamiz
$img = 'baner.png';
$arr = GetImageSize ($img);
if ($arr[2]==2){ print "$img <b>Jpeg file </b> <br>";}
if ($arr[2]==3){ print "$img <b>Png file </b> <br>";}
/*
natija:
bad.jpg Jpeg file
baner.png Png file
*/
?>

```

Ya'ni bu funksiya fayl turini chiqaradi, agar \$arr[2]==2 bo'lsa, u holda bu Jpeg format, agar \$arr[2]==3 bo'lsa, u holda bu Png format.

```

<?php
function TypeFile ($img_name){
    $arr = @GetImageSize ($img_name);
    return $arr[2];
}
// bu funksiya keyinchalik fayl turini aniqlaydi
// agar 2 chiqarsa - jpeg format, 3 chiqarsa – png format
?>

```

Biz TypeFile funksiya yaratdik va uni skriptga qo'shamiz.

ImageCreateFrom – mavjud tasvirlarni grafik ob'yektda o'zlashtiradi va ochadi. ImageCreateFromPng - Png formati uchun,

ImageCreateFromJpeg – esa Jpeg formati uchun. Bu funksiya ishlashiga ishonch hosil qilish uchun ma'lum bo'lgan GetImageSize () dan foydalanamiz.

```
<?php
    // agar 'bad.jpg' fayl bo'lmasa – uni yaratamiz
    $img = 'bad.jpg';
    if (! @file_exists ($img)){
    $im = ImageCreate (200,200);
    $red = ImageColorAllocate($im, 255, 0, 0);
    ImageString ($im, 3, 3, 3, "Super!!!", $red);
    ImageJpeg ($im,$img);
    ImageDestroy ($im);
    }
    // agar 'baner.png' fayl bo'lmasa – uni yaratamiz
    $img = 'baner.png';
    if (! @file_exists ($img)){
    $im = ImageCreate (200,200);
    $bgcolor = ImageColorAllocate($im, 255, 255, 255);
    $bgcolortrans = ImageColorTransparent($im, $bgcolor);
    $black = ImageColorAllocate($im, 0, 0, 0);
    ImageString ($im, 3, 3, 3, "Super!!!", $black);
    ImagePng ($im,$img);
    ImageDestroy ($im);
    }
    function TypeFile ($img_name){
    $arr = @GetImageSize ($img_name);
    return $arr[2];
    }
    // tayyor rasmlardan ob'yekt yaratish bo'yicha funksiya
    function LoadImage ($img_name){
    // TypeFile qo'llaymiz
    //agar ==2 bo'lsa, u holda Jpeg format, Jpeg ob'yekt yaratamiz
    if (TypeFile ($img_name)==2) {
    $im = @ImageCreateFromJpeg($img_name); }
    //agar ==3 bo'lsa, u holda Png format, Png ob'yekt yaratamiz
    elseif (TypeFile ($img_name)==3) {
    $im = @ImageCreateFromPng($img_name); }
    // agar boshqa formatlar bo'lsa, u holfa ob'yekt yo'q
    // yangi ob'yekt yaratamiz va unda xatolarni yozamiz
```

```

else {
$im = ImageCreate (200,200);
$bgc = ImageColorAllocate($im, 255, 255, 255);
$tc = ImageColorAllocate($im, 0, 0, 0);
ImageString ($im, 2, 3, 3, "Error loading $img_name", $tc);
}
// yaratilgan ob'yektni chiqaramiz
return $im;
}
// rasmlar nomi
$img1 = 'baner.png';
$img2 = 'bad.jpg';
// boshida $img1 qo'ying, so'ngra $img2 ga almashtiring
$img_name = $img1;
// funksiyani ishga tushiring
$im = LoadImage ($img_name);
// brauzerga bu rasm ekanligini bildiramiz
header ("Content-type: image/png");
// brauzerga rasmni chiqaramiz
ImagePng ($im);
// xotirani bo'shatamiz
ImageDestroy ($im);
?>

```

Bizda nima paydo bo'ldi: agar .jpeg yoki .png fayllar uzatilgan bo'lsa, ob'yekt yaratildi va uni brauzerda chiqardik, agar formatlar boshqa bo'lsa, yoki fayl nomi xato bo'lsa, biz yangi ob'yekt yaratdik va unga xatolarni yozdik.

ImageSX va ImageSY – ob'yekt eni va balandligi

```
<?php
```

```

// agar 'bad.jpg' fayl bo'lmasa – uni yaratamiz
$img = 'bad.jpg';
if (! @file_exists ($img)){
$im = ImageCreate (200,200);
$red = ImageColorAllocate($im, 255, 0, 0);
ImageString ($im, 3, 3, 3, "Super!!!", $red);
ImageJpeg ($im,$img);
ImageDestroy ($im);
}
// agar 'baner.png' fayl bo'lmasa – uni yaratamiz

```

```

$img = 'baner.png';
if (! @file_exists ($img)){
$img = ImageCreate (200,200);
$bgcolor = ImageColorAllocate($im, 255, 255, 255);
$bgcolortrans = ImageColorTransparent($im, $bgcolor);
$black = ImageColorAllocate($im, 0, 0, 0);
ImageString ($im, 3, 3, 3, "Super!!!", $black);
ImagePng ($im,$img);
ImageDestroy ($im);
}
function TypeFile ($img_name){
$arr = @GetImageSize ($img_name);
return $arr[2];
}
function LoadImage ($img_name){
if (TypeFile($img_name)==2) {
$img = @ImageCreateFromJpeg($img_name); }
elseif (TypeFile($img_name)==3) {
$img = @ImageCreateFromPng($img_name); }
else {
$img = ImageCreate (200,200);
$bgc = ImageColorAllocate($im, 255, 255, 255);
$tc = ImageColorAllocate($im, 0, 0, 0);
ImageString ($im, 2, 3, 3, "Error loading $img_name", $tc);
}
return $im;
}
$img1 = 'baner.png';
$img2 = 'bad.jpg';
// 'baner.png' birinchi rasmni qo'yamiz
$img_name = $img1;
// Png ob'yekt yaratamiz
$img = LoadImage($img_name);
// ob'yekt eni va bo'yini aniqlaymiz
$width = ImageSX ($im);
$height = ImageSY ($im);
print "$img_name:  Eni  <b>$width</b>  Balandligi  esa
<b>$height</b> <br>";
// 'bad.jpg' ikkinchi rasmni qo'yamiz

```



```

$img_name = $img2;
    // Jpeg ob'yektni yaratamiz
$img = LoadImage($img_name);
    // ob'yekt eni va balandligini aniqlaymiz
$width = ImageSX ($img);
$height = ImageSY ($img);
print    "$img_name:    Eni    <b>$width</b>    Balandligi    esa
<b>$height</b> <br>";
/*
natija:
baner.png: Eni 200 Balandligi esa 200
bad.jpg: Eni 200 Balandligi esa 200
*/
?>

```

GetImageSize – bu funksiyaga yana qaytamiz. Bundan tashqari u rasm turini aniqlaydi, ImageSX, ImageSY grafik ob'yektlar o'lchamini chiqargani kabi u ham o'lchamlarni chiqaradi.

```

<?php
    // agar 'bad.jpg' fayl bo'lmasa – uni yaratamiz
$img = 'bad.jpg';
if (! @file_exists ($img)){
$img = ImageCreate (200,200);
$red = ImageColorAllocate($img, 255, 0, 0);
ImageString ($img, 3, 3, 3, "Super!!!", $red);
ImageJpeg ($img,$img);
ImageDestroy ($img);
}
    // agar 'baner.png' fayl bo'lmasa – uni yaratamiz
$img = 'baner.png';
if (! @file_exists ($img)){
$img = ImageCreate (200,200);
$bgcolor = ImageColorAllocate($img, 255, 255, 255);
$bgcolortrans = ImageColorTransparent($img, $bgcolor);
$black = ImageColorAllocate($img, 0, 0, 0);
ImageString ($img, 3, 3, 3, "Super!!!", $black);
ImagePng ($img,$img);
ImageDestroy ($img);
}
    // fayllarimizni o'zgaruvchiga o'zlashtiramiz

```

```

$img1 = 'baner.png';
$img2 = 'bad.jpg';
    // 'baner.png' birinchi rasmni qo'yamiz
$img_name = $img1;
    // GetImageSize qo'llaymiz
$arr = GetImageSize ($img_name);
    // eni
$width = $arr[0];
    // balandligi
$height = $arr[1];
    // satr ko'rinishi "width=xxx height=xxx"
    // html-tegga qo'yish mumkin
$html_teg = $arr[3];
print "
$img_name:<br>
<b>$width</b> Eni <br>
<b>$height</b> Balandligi <br>
<b>$html_teg</b> html teg uchun
";
/*
baner.png:
Eni 200
Balandligi 200
width="200" height="200" html teg uchun
*/
?>

```

Endi biz rasmning o'lchamini aniqlashni bilamiz.

Image - brauzerga yuborish yoki faylda grafik obyekt saqlash.
 ImagePng - Png format uchun, ImageJpeg – esa Jpeg format uchun.
 Yuqorida bu fayllar ishlashini ko'rdik. Endi ular ob'yektlarni faylga saqlashini ko'rib chiqamiz.

```

<?php
    // agar 'bad.jpg' fayl bo'lmasa – uni yaratamiz
$img = 'bad.jpg';
if (! @file_exists ($img)){
    $im = ImageCreate (200,200);
    $red = ImageColorAllocate($im, 255, 0, 0);
    ImageString ($im, 3, 3, 3, "Super!!!", $red);
    ImageJpeg ($im,$img);

```

```

ImageDestroy ($im);
}
// agar 'baner.png' fayl bo'lmasa – uni yaratamiz
$img = 'baner.png';
if (! @file_exists ($img)){
$img = ImageCreate (200,200);
$bgcolor = ImageColorAllocate($im, 255, 255, 255);
$bgcolortrans = ImageColorTransparent($im, $bgcolor);
$black = ImageColorAllocate($im, 0, 0, 0);
ImageString ($im, 3, 3, 3, "Super!!!", $black);
ImagePng ($im,$img);
ImageDestroy ($im);
}
function TypeFile ($img_name){
$arr = @GetImageSize ($img_name);
return $arr[2];
}
function LoadImage ($img_name){
if (TypeFile ($img_name)==2){
$img = @ImageCreateFromJpeg($img_name); }
elseif (TypeFile ($img_name)==3){
$img = @ImageCreateFromPng($img_name); }
else {
$img = ImageCreate (200,200);
$bgc = ImageColorAllocate($im, 255, 255, 255);
$tc = ImageColorAllocate($im, 0, 0, 0);
ImageString ($im, 2, 3, 3, "Error loading $img_name", $tc);
}
return $im;
}
$img1 = 'baner.png';
$img2 = 'bad.jpg';
// 'baner.png' birinchi rasmni qo'yamiz
$img_name = $img1;
// _new so'zini qo'shamiz, endi $img_new = 'baner_new.png'
preg_match ("/(.+)(\..+)/",$img_name,$arr);
$img_new = $arr[1].'_new'.$arr[2];
// fayldan grafik ob'yekt yaratamiz
$img = LoadImage ($img_name);

```

```

    // baner_new.png yangi fayliga grafik ob'yektni saqlaymiz
if (TypeFile ($img_name)==2){ ImageJpeg ($im,$img_new); }
if (TypeFile ($img_name)==3){ ImagePng ( $im,$img_new); }
    // 'bad.jpg' ikkinchi rasmni qo'yamiz
$img_name = $img2;
    // _new so'zini qo'shamiz, endi $img_new = 'bad_new.jpg'
preg_match ("/(.+)(\..+)/",$img_name,$arr);
$img_new = $arr[1].'_new'.$arr[2];
    // fayldan grafik ob'yekt yaratamiz
$im = LoadImage ($img_name);
    // bad_new.jpg yangi fayliga grafik ob'yektni saqlaymiz
if (TypeFile ($img_name)==2){ ImageJpeg ($im,$img_new); }
if (TypeFile ($img_name)==3){ ImagePng ( $im,$img_new); }
    // html kiritamiz va brauzerga chiqaramiz
print"
<!DOCTYPE HTML PUBLIC '-//W3C//DTD HTML 4.0
Transitional//EN'>
<HTML><HEAD><TITLE></TITLE></HEAD>
<body>
<img src='baner_new.png'> <br>
<img src='bad_new.jpg'>
</body></html>";
?>
    Fayl nomlari dublikatini yaratish bilan birga, fayl nomlariga "_new"
ni qo'shdik, so'ngra fayllarimizdan grafik ob'yektlar yaratdik va bu
ob'yektlarni fayl dublikatlariga joyladik. Yaratilgan '/images' papkada
ikkita yangi fayl paydo bo'ladi. Shundan so'ng ularni oddiy html matn
orqali chiqardik.
    Boshqa narsalar bilan bir qatorda, ImageJpeg brauzerga chiqarish
yoki rasmni faylga sifatsiz holda saqlashi, ya'ni fayldagi rasmni sifati
yomonlashishi mumkin.
    <?php
ImageJpeg ($im,$img_new,20);
    // 20% sifati bilan rasmni chiqarish, ya'ni
    // Jpeg rasmlarni brauzerga chiqarish yoki saqlash
    // 0% dan 100% gacha sifati bo'lishi mumkin
?>

```

ImageDestroy – grafik ob'yektni yo'qotish. Berilgan ob'yektni saqlashda ahratilgan xotirani bo'shatish uchun senariyni shu funktsiya bilan tugatish kerak.

```
<?php
// .....
ImageDestroy ($im);
?>
```

ImageCopy – rasm qismlarini nushalash.

```
ImageCopy ($dst_im, $src_im, $dst_x, $dst_y, $src_x, $src_y,
$src_w, $src_h)
```

\$src_x, \$src_y nuqtalarda, eni - \$src_w va balandligi \$src_h, \$dst_im grafik ob'yektdan \$dst_x, \$dst_y pozitsiyaga, \$src_im - grafikaning to'rtburchaklar qismini ko'chiradi.

```
<?php
// agar 'bad.jpg' fayl bo'lmasa – uni yaratamiz
$img = 'bad.jpg';
if (! @file_exists ($img)){
    $im = ImageCreate (200,200);
    $red = ImageColorAllocate($im, 255, 0, 0);
    ImageString ($im, 3, 3, 3, "Super!!!", $red);
    ImageJpeg ($im,$img);
    ImageDestroy ($im);
}
// agar 'baner.png' fayl bo'lmasa – uni yaratamiz
$img = 'baner.png';
if (! @file_exists ($img)){
    $im = ImageCreate (200,200);
    $bgcolor = ImageColorAllocate($im, 255, 255, 255);
    $bgcolortrans = ImageColorTransparent($im, $bgcolor);
    $black = ImageColorAllocate($im, 0, 0, 0);
    ImageString ($im, 3, 3, 3, "Super!!!", $black);
    ImagePng ($im,$img);
    ImageDestroy ($im);
}
header ("Content-type: image/jpeg");
$img1 = imageCreateFromJpeg("bad.jpg");
$img2 = imageCreateFromPng("baner.png");
ImageCopy ($img1, $img2, 10, 15, 0, 0, 90, 100);
imageJpeg ($img1);
```

```
imageDestroy ($im1);
imageDestroy ($im2);
?>
```

Berilgan misolda "baner.png" fayl tasvirning qismlari 90x100 o'lchamli, 0, 0 pozitsiyadan 10, 15 pozitsiyaga "bad.jpg" faylga ko'chiriladi va u ekranga chiqariladi.

ImageCopyResized – yoqoridagi funksiya kabi, lekin be yerda chiquvchi ob'yekt o'lchamlarini o'zgartirish uchun ikkita argument qo'shilgan (то же самое, что и предыдущая функция, но здесь добавлены два аргумента для изменения размера объекта на выходе (kenglik va uzunlik, mos ravishda, chiqishda teng bo'ladi: \$dst_w, \$dst_h)

```
ImageCopy ($dst_im, $src_im, $dst_x, $dst_y, $src_x, $src_y,
$dst_w, $dst_h, $src_w, $src_h)
```

```
<?php
```

```
    // agar 'bad.jpg' fayl bo'lmasa – uni yaratamiz
```

```
$img = 'bad.jpg';
```

```
if (! @file_exists ($img)){
```

```
$im = ImageCreate (200,200);
```

```
$red = ImageColorAllocate($im, 255, 0, 0);
```

```
ImageString ($im, 3, 3, 3, "Super!!!", $red);
```

```
ImageJpeg ($im,$img);
```

```
ImageDestroy ($im);
```

```
}
```

```
    // agar 'baner.png' fayl bo'lmasa – uni yaratamiz
```

```
$img = 'baner.png';
```

```
if (! @file_exists ($img)){
```

```
$im = ImageCreate (200,200);
```

```
$bgcolor = ImageColorAllocate($im, 255, 255, 255);
```

```
$bgcolortrans = ImageColorTransparent($im, $bgcolor);
```

```
$black = ImageColorAllocate($im, 0, 0, 0);
```

```
ImageString ($im, 3, 3, 3, "Super!!!", $black);
```

```
ImagePng ($im,$img);
```

```
ImageDestroy ($im);
```

```
}
```

```
header ("Content-type: image/jpeg");
```

```
$im1 = imageCreateFromJpeg("bad.jpg");
```

```
$im2 = imageCreateFromPng("baner.png");
```

```
ImageCopyResized($im1, $im2, 10, 15, 0, 0, 45, 50, 90, 100);
```

```
imageJpeg ($im1);
imageDestroy ($im1);
imageDestroy ($im2);
?>
```

Ushbu misolda 90x100 o'lchamli "baner.png" faylining 0, 0 pozitsiyasidan 0, 0 pozitsiyasida "bad.jpg" faylning 10, 15 pozitsiyasiga 45x50 o'lchamgacha tasvir bir qismi ko'chiriladi va u chop etiladi.

```
<?php
    // agar 'baner.png' fayl bo'lmasa – uni yaratamiz
    $img = 'baner.png';
    if (! @file_exists ($img)){
        $im = ImageCreate (200,200);
        $bgcolor = ImageColorAllocate($im, 255, 255, 255);
        $bgcolortrans = ImageColorTransparent($im, $bgcolor);
        $black = ImageColorAllocate($im, 0, 0, 0);
        ImageString ($im, 3, 3, 3, "Super!!!", $black);
        ImagePng ($im,$img);
        ImageDestroy ($im);
    }
    $img_name = "baner.png";
    // 'baner_new.png' nomli $img_new dublikat
    preg_match ("/(.+)(\..+)/",$img_name,$arr);
    $img_new = $arr[1].'_new'.$arr[2];
    // 'baner.png' rasmidan ob'yekt yaratamiz
    $im = @imageCreateFromPng($img_name);
    // 'baner.png' o'lchamlarini bilib olamiz
    $width = ImageSX ($im);
    $height = ImageSY ($im);
    // 100x75 o'lchamli yangi ob'yekt yaratamiz
    $new = @imageCreate (100, 75);
    // 0,0 pozitsiyada 'baner.png' bajarilgan $im nusxalaymiz, ya'ni
    // $width, $height o'lchamlardagi nolinch nuqta – ya'ni uning
rostligi
    //o'lchamlarda, $new yangi ob'yekt 100, 75 o'lchamgacha,
    // 0,0 nolinch nuqtada 100x75 o'lchamga ega
    @imageCopyResized($new, $im, 0, 0, 0, 0, 100, 75, $width,
    $height);
    // kichiklashtirilgan rasmni 'baner_new.png' faylga ko'chiramiz
    ImagePng ($new,$img_new);
```

```

    // html chop etamiz
print"
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML><HEAD><TITLE></TITLE></HEAD>
<body>
<img src='baner_new.png'> <br>
<img src='baner.png'>
</body></html>";
?>

```

Bu yerda rasmning kichraytirilgan o'lchami kompyuterga saqlandi.

ImageColorTransparent – bu funksiya Png formatni shaffof ko'rinishga olib keladi. Jpeg formatda ishlamaydi.

```

<?php
$img_new = "baner_new.png";
$im = ImageCreate (100,20);
$bgcolor = ImageColorAllocate($im, 255, 255, 255);
$bgcolortrans = ImageColorTransparent($im, $bgcolor);
$black = ImageColorAllocate($im, 0, 0, 0);
ImageString ($im, 3, 3, 3, "Super!!!",$black);
ImagePng ($im,$img_new);
ImageDestroy ($im);

```

```

    // html chop etamiz

```

```

print"
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML><HEAD><TITLE></TITLE></HEAD>
<BODY bgcolor='#aa22ff'>
<img src='baner_new.png'>
</BODY></HTML>";
?>

```

Berilgan funksiya rasmni shaffof ko'rinishga olib keladi, ya'ni asosiy sahifaning fon rangiga moslaydi.

ImageRectangle – to'ldirilmagan to'g'ri burchakni chizish.

```

<?php
$img_new = "baner_new.png";
$im = ImageCreate (100,20);
$bgcolor = ImageColorAllocate($im, 255, 255, 255);
$bgcolortrans = ImageColorTransparent($im, $bgcolor);

```



```

$blue = ImageColorAllocate($im, 20, 93, 233);
ImageRectangle ($im, 5, 10, 60, 15, $blue);
ImagePng ($im,$img_new);
ImageDestroy ($im);
    // html chop etamiz
print"
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML><HEAD><TITLE></TITLE></HEAD>
<BODY>
<img src='baner_new.png'>
</BODY></HTML>";
?>

```

5, 10 nuqtalardagi yuqori chap burchak. O'ng pastki - 60,15. Agar ImageFilledRectangle qo'yilsa – to'ldirilgan to'g'riburchak chiziladi.

ImagePolygon – ko'pburchakni chizish.

```

<?php
$img_new = "baner_new.png";
$im = ImageCreate (100,20);
$bgcolor = ImageColorAllocate($im, 255, 255, 255);
$bgcolortrans = ImageColorTransparent($im, $bgcolor);
$blue = ImageColorAllocate($im, 20, 93, 233);
$points = array (12, 10, 15, 20, 50, 17, 70, 10);
ImagePolygon ($im, $points, 4, $blue);
ImagePng ($im,$img_new);
ImageDestroy ($im);
    // html chop etamiz
print"
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<HTML><HEAD><TITLE></TITLE></HEAD>
<BODY>
<img src='baner_new.png'>
</BODY></HTML>";
?>

```

Agar ImageFilledPolygon qo'llanilsa, ko'pburchak to'ldirilgan bo'ladi.

ImageArc – rasm chiziqlar.

```

<?php

```

```

$img_new = "baner_new.png";
$im = ImageCreate (100,20);
$bgcolor = ImageColorAllocate($im, 255, 255, 255);
$bgcolortrans = ImageColorTransparent($im, $bgcolor);
$red = ImageColorAllocate($im, 233, 93, 20);
ImageArc ($im, 4, 5, 50, 40, 10, 180, $red);
ImagePng ($im,$img_new);
ImageDestroy ($im);
    // html chop etamiz
print"
<!DOCTYPE HTML PUBLIC '-//W3C//DTD HTML 4.0

```

Transitional//EN'>

```

<HTML><HEAD><TITLE></TITLE></HEAD>
<BODY>
<img src='baner_new.png'>
</BODY></HTML>";
?>

```

ImageInterlace - chiziq orqali tasvirni yaratadi.

```

<?php
$img_new = "baner_new.png";
$im = ImageCreate (100,20);
ImageInterlace ($im, 1);
$bgcolor = ImageColorAllocate($im, 255, 255, 255);
$bgcolortrans = ImageColorTransparent($im, $bgcolor);
$red = ImageColorAllocate($im, 233, 93, 20);
ImageArc ($im, 4, 5, 50, 40, 10, 180, $red);
ImagePng ($im,$img_new);
ImageDestroy ($im);
    // html chop etamiz
print"
<!DOCTYPE HTML PUBLIC '-//W3C//DTD HTML 4.0

```

Transitional//EN'>

```

<HTML><HEAD><TITLE></TITLE></HEAD>
<BODY>
<img src='baner_new.png'>
</BODY></HTML>";
?>

```

Agar ikkinchi argument 1 ga teng bo'lsa, u holda tasvir satr orqali o'zlashtiriladi.

ImageChar – belgini gorizontal ko'rinishda chiqaradi.

```
<?php
$img_new = "baner_new.png";
$stroka = "Hello!";
$im = ImageCreate (140,50);
ImageInterlace ($im, 1);
$bgcolor = ImageColorAllocate($im, 255, 255, 255);
$bgcolortrans = ImageColorTransparent($im, $bgcolor);
$red = ImageColorAllocate($im, 233, 93, 20);
$x = 0;
for ($i=0; $i<strlen($stroka); $i++, $x+=15){
if (! empty ($stroka[$i])) ImageChar ($im, 5, $x, 20, $stroka[$i],
$red);
}
ImagePng ($im,$img_new);
ImageDestroy ($im);
// html chop etamiz
print"
<!DOCTYPE HTML PUBLIC '-//W3C//DTD HTML 4.0
Transitional//EN'>
<HTML><HEAD><TITLE></TITLE></HEAD>
<BODY>
<img src='baner_new.png'>
</BODY></HTML>";
?>
```

ImageCharUp – belgini vertikal ko'rinishda chiqaradi.

```
<?php
$img_new = "baner_new.png";
$stroka = "Hello!";
$im = ImageCreate (50,140);
ImageInterlace ($im, 1);
$bgcolor = ImageColorAllocate($im, 255, 255, 255);
$bgcolortrans = ImageColorTransparent($im, $bgcolor);
$red = ImageColorAllocate($im, 233, 93, 20);
for ($i= strlen ($stroka), $x=20; $i>=0; $i--, $x+=15){
if (! empty ($stroka[$i])) ImageCharUp ($im, 5, 10, $x, $stroka[$i],
$red);
}
ImagePng ($im,$img_new);
```

```

ImageDestroy ($im);
    // html chop etamiz
print"
<!DOCTYPE HTML PUBLIC '-//W3C//DTD HTML 4.0
Transitional//EN'>
<HTML><HEAD><TITLE></TITLE></HEAD>
<BODY>
<img src='baner_new.png'>
</BODY></HTML>";
?>

```

14.3. PHP da GET va POST usullar

PHP da o'zgaruvchilar dollar (\$) belgisidan boshlanadi. Bu simvoldan ixtiyoriy sondagi harf, raqam va simvollar kelishi mumkin, lekin birinchi simvol albatta harf bo'lishi kerak. Shuni esda tutish kerakki, php da o'zgaruvchilarning nomlari kalit so'zlardan farqli registrga bog'liqdir.

Tashqi o'zgaruvchilar. Klient so'rovi web-server tomonidan taxlil qilinib, php mashinaga uzatilgandan so'ng u so'rovga tegishli ma'lumotlarni o'z ichiga olgan va bajarish davomida murojaat qilish mumkin bo'lgan bir necha o'zgaruvchilarni yaratadi.

GET- so'rov satrini taxlil qilishda yaratiladi. So'rov satri \$QUERY_STRING o'zgaruvchida saqlanadi va so'ralgan URL dagi "?" simvoldan keyingi axborotdan iborat. Php da so'rov satrini & simvollar bo'yicha aloxida elementlarga ajratadi, va har bir elementda "=" belgisini qidiradi. Agar "=" belgisi topilgan bo'lsa, tenglik chap tomonidagi simvoldan iborat o'zgaruvchi yaratadi. Quyidagi formani ko'ramiz:

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
<title>php da get</title>
</head>
<body>
<?php
$a = $_GET['a'];
$b = $_GET['b'];
$c = $a + $b;

```

```

echo "<br>yig'indining qiymati: $c";
echo "<br><a href='index.php?a=35&b=15'>Yangi qiymat</a>";
?>
</body>
</html>

```

POST - o'zgaruvchilarning guruxi yaratilib, interpretatsiya qilinadi va \$HTTP_POST_VARS massivga joylashtiriladi.

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
<title>Php da POST</title>
</head>
<body>
<form action="index1.php" method="post" name="form1"
target="_blank">
<p>Sizning ismingiz: <br><input name="ismi" type="text"
size="20"></p>
<p>Sizning familiyangiz: <br><input name="fam" type="text"
size="20"></p>
<p><input name="submit" type="submit" value="Yuborish"></p>
</form>
</body>
</html>

```

Index1.php bu o'zgaruvchilar bilan oddiy o'zgaruvchilar bilan ishlagandek ishlashingiz mumkin. Bizning misolimizda ular ekranga chiqariladi:

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
<title>POST php ni qabul qiladi</title>
</head>
<body>
<?php
$ismi = $_POST['ismi'];
$familiya = $_POST['fam'];
echo "Ismingiz: $ismi va Familiyangiz: $familiya";
?>

```

</body>

</html>

NAZORAT SAVOLLARI

1. PHP tilida funksiyalar va ularni e'lon qilish?
2. PHP tilida funksiyalar va ularning argumentlari?
3. PHP tilida grafik ma'lumotlar va GD kutubxona?
4. PHP tilida GET va POST usullar?

15-BOB. PHP XATOLARNI QAYTA ISHLASH VA MUSTASNO QILISH

15.1. Xatolarning roli va turlari

Ichki xato xabari foydalanuvchining brauzerida namoyish qilishning mantiqiy bo'lmagan xatosidir (agar foydalanuvchi o'zi foydalanuvchi bo'lsa). Bunday xatoni keying tahlil uchun jurnal fayllariga yozib qo'yish yahshiroq, brauzerda esa standart matnni chiqarish kerak: masalan, "Ichki xatolik yuzaga keldi, u haqidagi ma'lumot skript ishlab chiqaruvchiga keyinroq beriladi". Ko'pchilik dasturchilar xato haqida qo'shimcha ma'lumotlarni sahifa so'ngida berishni ma'qul ko'rishadi, ya'ni xabarni fayl jurnalga yozish va ekranda chiqarish. Ushbu amaliyot, ehtimol, javobgarlikka tortilishi mumkin emas, chunki ko'p hollarda, dasturchilarga "joyida" nima yuz berganini bilishga yordam beradi.

Shu bilan birga xato haqidagi foydalanuvchi xabari foydalanuvchiga ko'rsatilishi uchun mo'ljallangan – uning nomi aynan shu yerdan. Bunday xatolik yuz berganda foydalanuvchi brauzerda mazmunli matnni ko'rishi kerak, shuningdek, nima qilish kerakligi haqida maslahat bo'lishi mumkin.

Foydalanuvchi xatolarini ichki sifatida qarsh mumkin emas - ular odatda ma'lum darajada berkingan bo'ladi. Maslan, SQL-server bilan ulana olmaslikda dasturda bir vaqtning o'zida ikkita ko'rinishli xabar chiqadi:

ichki: SQL-server javobi, kun va vaqt xatosi, dasturdagi qator raqami va boshqalar;

foydalanuvchi: masalan, matnda "SQL-server bilan bog'lanishda xatolik, keyinroq urinib ko'ring".

Xatolar turlari. Bu yerda asosan "xato" atamasining ikkinchi va uchinchi qiymatlarini ko'rib chiqamiz. Oddiy holda bu axborot o'ziga diagnostik xabarlar matnini oladi, lekin qo'shimcha ma'lumotlar ham aniqlanishi mumkin, masalan, xatolik yuz bergan fayl nomi va satr raqami.

Agar dasturda zatolik yuz bergan bo'lsa, u holda bu holatda nima qilish kerakligi haqida echim qabul qilish kerak. Bu bilan shug'ullanadigan dastur kodini (agar u mavjud bo'lsa), xatodan keyingi tiklovchi kod deb nomlanadi, bu kodni ishga tushirish esa – xatodan keyingi tiklanish deb ataladi. Masalan, quyidagi misolni ko'ramiz:

```
$f = @fopen("spoon.txt", "g");  
if (!$f) return;
```

Bu misolda qayta tiklovchi kod – bu faylni ochib bo’lmaslik holatini qayta ishlovchi if qo’llanmadir. E’tibor qiling, @ operatorini fopeno dan oldin PHP dan diagnostik xabarni olmaslik uchun foydalandik – u bizga kerak emas, bizda xato holatlarni qayta ishlovchi mavjud (qayta tiklash kodi).

Xatolarni “jiddiy”ligi bo’yicha ikkita katta sinfga bo’lish mumkin:

- qayta tiklash imkoni bo’lmagan jiddiy xatolar. Masalan, mavjud bo’lmagan faylni ochishga harakat qilmoqdasiz, faylni ocha olmayotgan bo’lsangiz, u hold anima qilish kerakligini ko’rsatishingiz kerak: ahir ochilmagan fayldan ma’lumotlarni o’qish yoki yozish mumkin emas;

- qayta tiklanishi talab etilmaydigan "jiddiy bo’lmagan" xatolar, masalan, ogohlantirish (warnings), xabarnomalar (notices), hamda qoldirilgan xabarlar (debug notices). Odatda bu ko’rinishdagi xatolar yuzaga kelgan hollarda biron-bir nostandart va asosiy choralarni ko’rish shart emas, faqatgina xatolik haqidagi ma’lumotni biron joyga saqlab qo’yish etarli (masalan, jurnal fayliga).

Shunday qilib, jiddiy xatolar uchun qayta tiklovchi kodni qo’lda yozishga va dasturning odatiy yo’lini buzishga majbur bo’lamiz, bu vaqtning o’zida jiddiy bo’lmagan xatolat uchun biron nima qilish shart emas.

15.2. Jiddiy bo’lmagan xatolar

“Shaxsiy” tiklanishi talab etilmaydigan jiddiy bo’lmagan xatolarni qayta ishlash uchun PHP da xatolarni qayta ishlashni sozlovchi deb nomlangan vosita mavjud. Qayta ishlovchi vazifasi – xato haqidagi ma’lumotni saqlash yoki uni brauzerga chiroyli dizayn bilan chiqarish.

```
<?php ## Xato va ogohlantirishlarni ushlab turish.
```

```
// Qayta ishlovchi yangi funksiyasini aniqlaymiz.
```

```
function myErrorHandler($errno, $msg, $file, $line) {
```

```
// agar @ foydalanilayotgan bo’lsa, hech narsa qilish kerak emas.
```

```
if (error_reporting() == 0) return;
```

```
// Aks holda – xabarni chiqaramiz.
```

```
echo ‘<div style="border-style:inset; border-width:2">‘;
```

```
echo "<b>$errno</b>!<br> kod bilan xatolik yuz berdi";
```

```
echo "Fayl: <tt>$file</tt>, satr $line.<br>";
```

```
echo "</div>";
```

```
}
```



```
// Uni barcha turdagi xatolar uchun ro'yhatdan o'tkazamiz.
set_error_handler("myErrorHandler", E_ALL);
// To'xtatiladigan ogohlantirishlarni yaratishda
// mavjud bo'lmagan fayl uchun funksiya chaqirish
filemtime ("spoon") ;
?>
```

Ko'rib turganimizdek nima bo'lganda ham dastur o'z ishisni har doim davom ettiradi – faqatgina xatolik yuz bergan holatdagina qayta ishlovchi – funksiya chaqiriladi, undan so'ng esa bajarish davom etadi. Aynan shu sababli qayta ishlovchini qayta tiklovchi kod deb bo'lmaydi.

15.3. Jiddiy xatolar

Jiddiy xatolarni avvalgi keltirilgan misollardan farqli ravishda, umumiy holda seterrorhandiero dan foydalanib qayta ishlash mumkin emas, chunki har bir aniq holatda qayta tiklovchi “shaxsiy” kodni yozish kerak.

Shunga ahamiyat berish kerakki, qayta ishlovchi – funksiyada xatodan so'ng qayta tiklash uchun birgina mazmunli harakatni bajarish mumkin – bu dasturni tugatadi.

Bazali sintaksis. Demak, istisno – bu “jiddiy” turdagi xato haqida ayrim xabarlar. Qachon yaratilsa, u avtomatik ravishda dasturning "ma'lumotli" qismiga o'tkaziladi va bu muayyan vaziyatda nima qilish kerakligini bildiradi.

Bu maydon istisno qayta ishlovchisi deb nomlanadi.

Dasturdagi ixtiyoriy istisno odatdagidek new operatorida yaratiladigan ayrim sinflarning ob'yektlarini o'zida tasvirlaydi. Bu ob'yekt turli xil axborotlarni, masalan, diagnostik xabar matni, hamda istisno generatsiyasi yuz bergan fayllar nomi va satrlar raqamini tashkil qilishi mumkin. Boshqa ixtiyoriy parametrlarni qo'shish mumkin.

Qayta ishlovchini chaqirishga misol ko'ramiz.

```
<?php #fr istisnolardan foydalanish bo'yicha oddiy misol,
echo "Dastur boshi.<br>";
try {
// Istisnolardan saqlanadigan kod.
echo " Har bir narsaning boshlanishi...<br>";
// Istisnoni yaratish.
throw new Exception("Hello!");
echo "...ega va ohiri.<br>";
```

```

} catch (Exception $e) {
// Qayta ishlovchi kodi.
echo " Istisno: {$e->getMessage()}<br>";
}
echo "Dastur tugadi.<br>";
?>

```

Yuqoridagi misolda try...catch konstruksiyasining bazaviy sintaksislariga misol ko'rib chiqildi.

throw vositasi. throw vosita istisno-ob'yektni generasiya qilib qolmasdan uni catch blok qayta ishlovchisiga uzatadi. U yana joriy try-blok ishini tez tugatadi.

Aynan shuning uchun yuqorida keltirilgan misolning natijasi quyidagicha bo'ladi:

Dastur boshi.

Boshidagi ega bo'lingan barcha narsa...

Istisno: Hello!

Datur tugadi.

Ushbu munosabatda throw vositasi return, break va continue vositasiga juda o'hshash: ular ham joriy funksiya yoki sikl iteratsiyasi ishini tez tugatishga olib keladi.

Qo'llab quvvatlanmaydigan try...finally konstruksiyasi. Java va Delphi dasturlash tillarida finalist-kodini amalga oshirish uchun try...finally juda qulay konstruksiya mavjud, return funksiyasi bo'yicha tasodifan tugatish yoki mustasno holatlar yuzaga kelganda ayrim harakatlarni bajarishni kafolatlashga foydalaniladi. PHP da u quyidagi ko'rinishda amalga oshiriladi:

```

function eatThisO { throw new Exception("bang-bang!"); }
function hello() {
echo "Boshlang'ich ega bo'lgan barcha narsa..., ";
try {
eatThisO;
} finally {
echo "Tugatishga ega.";
}
echo "this never prints!";
// Funksiyani caqirish.
helloO;

```

try...finally vositasining sintaksisi tushunarli bo'lishi shart: u try-blokdan tasodifan chiqish yuzaga kelgan holatda ham finally-blok bajarilishini kafolatlaydi.

Afsuski, PHP5 da qurilgan Zend Engine 2 try...finally konstruksiyasini qo'llab quvvatlamaydi, agar try...finally konstruksiyasini qo'llab quvvatlovchi sizda mavjud bo'lsa u holda yuqorida keltirilgan dastur matnini natijasini ko'rish mumkin bo'ladi.

NAZORAT SAVOLLARI

1. PHP tilida xatolar va ularni qayta ishlash?
2. Xatolarning roli va ularning turlari?
3. Jiddiy va jiddiy bo'lmagan xatolar?

16-BOB. PHP COOKIE VA SESSIYALAR. O'ZGARUVCHILARNING KO'RINISH SOHALARI

16.1. Cookie ni o'rnatish va qabul qilish

Cookie — mijoz kompyuterida saqlanuvchi va u har safar serverga murojaat qilayotganda web-serverga yuboriladigan matn satridir. Shu tariqa ma'lumotlar turli skriptlar aro formalar yoki URL manzillarsiz uzatilishi mumkin. Cookie qiymatini o'rnatish uchun quyidagi sintaksisdan iborat bo'lgan setcookie funksiyasidan foydalaniladi:

```
bool setcookie (string name [, string valuye [, int expire [, string path [, string domain [, bool secure]]]])
```

Funksiya mijoz kompyuterida saqlanuvchi cookie larni tasnif etadi. Quyida ushbu funksiya parametrlarining tasnifi ketirilgan:

- name. Cookie ning nomi.
- valuye. Cookie ning qiymati.
- expire. Cookie ning amal qilish muddati. Agar u berilsa, u tugagach cookie o'chirib tashlanadi. Agar ko'rsatilmasa, cookie brauzer oynasi yopilgandan so'ng o'chirib tashlanadi
- path. Serverdagi cookie ga ruxsat etilgan manzil.
- domain. cookie ga ruxsat etilgan domen.
- secure. HTTPS protokol orqali bog'lanishda cookie ning havfsizlik belgisi. Standart holatda cookie HTTPS uchun ham HTTP dagi kabi ishlaydi.

Ko'rinib turibdiki, cookie HTTP-so'rovning qismi hisoblanadi va u brauzerga yuboriladi. Shu sabab HTML-kod formalashtirishdan oldin uning qiymati o'rnatilishi kerak. Bu shuni anglatadiki, setcookie funksiyasining qiymati HTML-tegdan oldin va echo operatorigacha o'rnatiladi.

Agar bu qoida saqlanmasa setcookie funksiyasini chaqirish FALSE qiymat qaytaradi va bu cookie shakllanishida xatolik bo'lganligini anglatadi. cookie ning to'g'ri shakllanishi funksiyaning TRUE qiymat qaytarishiga olib keladi. Bu cookie mijozga qabul qilindi degani emas albatta. Chunki brauzerni sozlashda cookie ni o'chirib tashlash yoki serverga jo'natmaslik parametrlari o'rnatilgan bo'lishi mumkin.

16.2. Cookie ni o'qish

Cookie qiymati o'rnatilganidan so'ng, sahifa qayta yuklanmagunicha darhol undagi skript uchun cookie ga ruxsat mavjud bo'lmaydi. Chunki Cookie foydalanuvchi kompyuterida saqlanadi va brauzer orqali web-serverga jo'natiladi. Bundan tashqari, cookie qachonki uning domeni server domeni bilan mos tushgandagina jo'natiladi.

Cookie ga ruxsatni olish uchun maxsus `$_COOKIE` superglobal massividan foydalaniladi. Massivning qiymati sifatida oldin ishlatilgan Cookie nomi olinadi. Massiv odatda skript yuklanayotgan vaqtda `$_GET`, `$_POST` va `$_REQUEST` massivlari bilan avtomatik tarzda to'ldiriladi.

Cookie dan foydalanishdan oldin uning qiymati o'rnatilganligiga ishonch hosil qiling. Buning uchun `isset` funksiyasidan foydalanish juda qulay. Quyidagi misolda message nomli Cookie ning qiymati tekshirilishi va ko'rsatilishi keltirilgan.

Misol:

```
<HTML>
<HEAD><TITLE>          Cookie          ning          qiymatini
o'qish</TTITLE></HEAD>
<BODY>
<CENTER>
<N1> Cookie ning qiymatini o'qish</H1>
Cookie ning qiymati:
<?php
if (isset ($_COOKIE ['message']))
{
echo Cookie ning qiymati:' . $_COOKIE ['message' ] ;
}
else
{
echo 'Cookie o'rnatilmagan' ;
}
?>
</CENTER;
<BODY>
</HTML>
```

Cookie massivlarda ham tashkil etilgan bo'lishi mumkin. Masalan quyida uchta cookie o'rnatilgan:

```
setcookie ("cookie[one]" , "Bugun");
```

```

setcookie ("cookie[two] ", "Hayot");
setcookie ("cookie[threya] ", "go'zal!");
Natijada $_COOKIE['cookie'] massiv qiymatlari quyidagi tarzda
bosmaga chiqarilishi mumkin:
if (isset ($_COOKIE ['cookie'])) {
foreach ($_COOKIE ['cookie'] as $data)
{
echo "$data <BR>";
...

```

16.3. PHP da sessiyalarnin ishlash mexanizmlari. Sessiya hosil qilish va o'chirish

Sessi va Cookie lar bir nechta sahifalar orasida o'tishlardagi foydalanuvchilar haqida ma'lumotlarni saqlash uchun mo'ljallangan. Sessidan foydalanish jarayonida ma'lumotlar serverdagi vaqtinchalik fayllarda saqlanadi. Cookie bilan fayllar foydalanuvchi kompyuterida saqlanadi, va brauzer bilan serverga so'rov bo'lganda jo'natadi.

Sessi va cookie lardan Internet-magazinlar, forumlar, e'lonlar doskasi kabi ilovalarda foydalanish juda ham qulay, birinchidan, bir nechta sahifalar o'tilgandagi foydalanuvchilar haqida ma'lumotlarni saqlash kerakligi, ikkinchidan esa foydalanuvchiga yangi ma'lumotlarni zamonaviy ko'rinishda taqdim etish.

Sessiyalar bilan ishlash quyidagicha bosqichlarga ajratiladi:

- Sessiyalarni ochish
- Sessiyalar o'zgaruvchilarini ro'yxatdan o'tkazish va ulardan foydalanish
- Sessiyalarning yopilishi

Sessiyalarni ochish. Sessiyalarni ochishning eng sodda usuli PHP-ssenariy boshida chaqiriladigan session_start funksiyasidan foydalanishdir:

Sintaksis:

```
session_start();
```

Bu funksiya sessiya identifikatori mavjudligini tekshiradi, agar mavjud bo'lmasa uni yaratadi. Agarda joriy sessiyaning identifikatori mavjud bo'lsa, u holda ro'yxatdan o'tkazilgan sessiya o'zgaruvchilari yuklanadi.

Sessiya o'zgaruvchilarini ro'yxatdan o'tkazish. Sessiya initsializatsiyasidan so'ng \$_SESSION superglobal massivida ma'lumotlarni saqlash imkoniyati paydo bo'ladi. Masalan, \$_SESSION

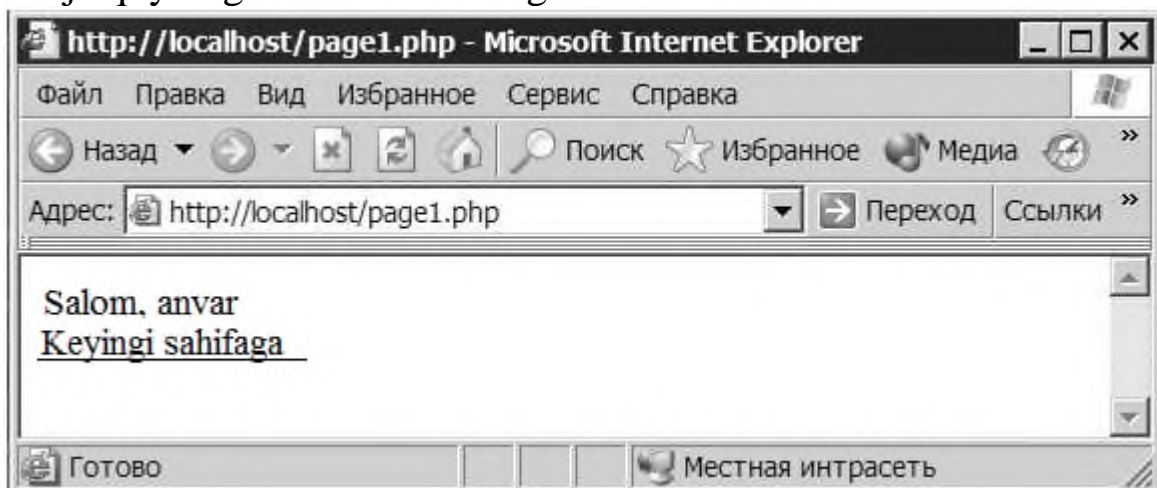
massivida o'zgaruvchilar va massiv saqlanayotgan index.php nomli fayl mavjud bo'lsin.

```
<?php
// Sessiyani boshlash
session_start();
// Sessiyaga qiymatlarni joylaymiz
$_SESSION['name'] = "value";
// Sessiyaga massivni joylaymiz
$arr = array("first", "second", "third");
$_SESSION['arr'] = $arr;
// Boshqa sahifaga xavola (ssilka)ni chiqaramiz
echo "<a href='other.php'>boshqa sahifa</a>";
?>
```

Uchta sahifa bilan ishlovchi sodda sessiyaga misol. Foydalanuvchi kirishi bilan birinchi sahifada sessiya ochiladi va \$username o'zgaruvchisi ro'yxatdan o'tkaziladi. Bu jarayon quyidagi ko'rinishda amalga oshiriladi:

```
<?
session_start();
$_SESSION['username'] = "anvar";
echo 'Salom, '.$_SESSION['username'].'<br>';
?>
<a href="page2.php">Keyingi sahifaga </a>
```

Natija quyidagi tasvirda keltirilgan:



16.1-rasm. Xavola (ssilka)ni bosib keyingi sahifaga o'tish

Foydalanuvchi anvar xavola (ssilka)ni bosadi va keyingi page2.php sahifaga o'tadi:

```
<?
session_start();
```



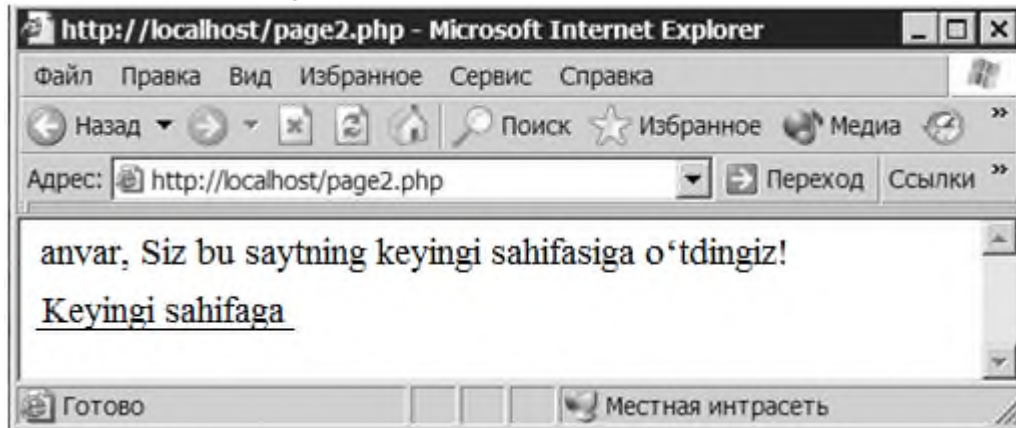
```
echo $_SESSION['username'].', Siz bu saytning keyingi sahifasiga o'tdingiz!';
```

```
echo("<br>");
```

```
?>
```

```
<a href="page3.php">Keyingi sahifaga </a>
```

Natija rasmda keltirilgan:



16.2-rasm. page2.php sahifa

Bu yerda ham foydalanuvchi xavola (ssilka)ni bosib keyingi page3.php sahifaga o'tadi, shu bilan birga seansli o'zgaruvchilarni ro'yxatdan o'tkazish va sessiyani o'chirish boshlanadi.

```
<?
```

```
session_start();
```

```
unset($_SESSION['username']);
```

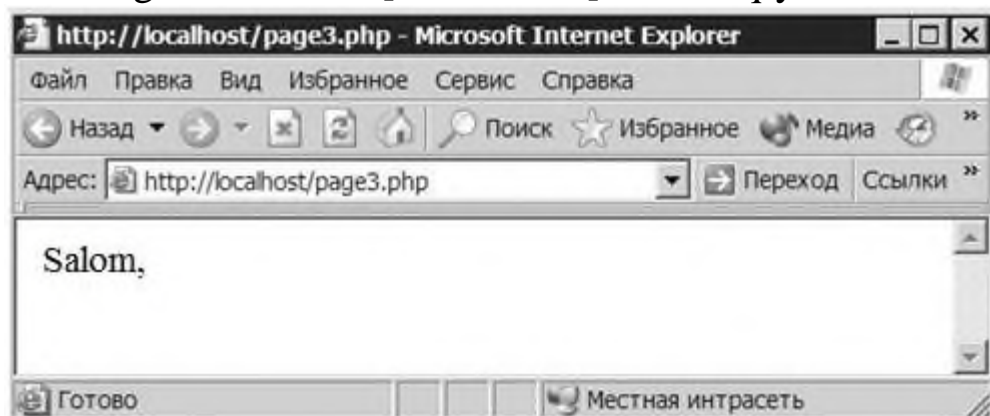
```
echo 'Salom, '.$_SESSION['username'];
```

```
/* Endi foydalanuvchi nomi chiqarilmaydi */
```

```
session_destroy(); // sessiyani to'htatamiz
```

```
?>
```

Tasvirdan ko'rinib turibdiki, seansli o'zgaruvchilarni ro'yxatdan o'tkazgandan so'ng \$_SESSION['username'] massiv qiymati shart emas:



16.3-rasm. page3.php sahifa ko'rinishi

Sessiyani tugatish. Sessiya tugatishdan avval birinchi navbatda sessiyaning barcha o'zgaruvchilarini ro'yxatdan o'tkazish, so'ngra unset() funksiyasini chaqirish kerak:

Sintaksis:

```
unset($_SESSION["username"]);
```

NAZORAT SAVOLLARI

1. PHP tilida Cookie ni qabul qilish va o'rnatish?
2. PHP tilida Cookie ni o'qish?
3. Cookie ni sozlashni amalga oshirish jarayoni?
4. PHP da sessiyalar (seanslar) bilan ishlash?
5. Sessiyalarni hosil qilish va o'chirish jarayoni?

17-BOB. PHP TILIDA FAYLLAR BILAN ISHLASH

17.1. Fayllar ustida amallar

PHP tilida fayllar bilan ishlash jarayonida quyidagi holatlarga duch kelish mumkin:

Fayl biror ma'lumot saqlash fizik qurilmasidagi baytlar ketma ketligidir. Har bir fayl joylashuvini ko'rsatuvchi absolyut yo'lga ega. Yo'l ajratuvchisi sifatida Windows da to'g'ri slesh (/) yoki teskari slesh (\) ishlatilishi mumkin. Boshqa operatsion tizimlarda faqat to'g'ri slesh ishlatiladi.

Fayllarni serever fayl tizimida ochish fopen funksiyasi yordamida amalga oshiriladi:

```
int fopen(string filename, string mode [, int use_include_path])
```

Birinchi argument filename – fayl nomi yoki unga olib boruvchi absolyut yo'l. Agar absolyut yo'l ko'rsatilmasa, fayl kattalogda joylashgan bo'lishi kerak.

Ikkinchi argument mode fayl qaysi amallar uchun ochilganligini ko'rsatadi va quyidagi qiymatlarga ega bo'lishi mumkin:

r (faylni faqat o'qish uchun ochish; ochilgandan so'ng fayl ko'rsatkichi fayl boshiga o'rnatiladi);

r+ (faylni o'qish va yozish uchun ochish; ochilgandan so'ng fayl ko'rsatkichi fayl boshiga o'rnatiladi);

w (Yozish uchun Yangi bo'sh fayl yaratish; agar shu nomli fayl mavjud bo'lsa, undagi hamma ma'lumot o'chiriladi);

w+ (Yozuvlarni o'qish uchun Yangi bo'sh fayl yaratish; agar shu nomli fayl mavjud bo'lsa, undagi hamma ma'lumot o'chiriladi);

a (Yozuv qo'shish uchun faylni ochish, ma'lumotlar fayl oxiriga yoziladi);

a+ (Yozuv qo'shish va o'qish uchun faylni ochish, ma'lumotlar fayl oxiriga yozilad);

b (Ikkilik fayl Bilan ishlash usulini(o'qishsh va yozish) ko'rsatuvchi bayroq; faqat Windows da ko'rsatiladi).

Uchinchi shart bo'lmagan argument use_include_path fayllar include_path kattalogida izlash kerakligini belgilaydi. (include_path parametr php.ini faylda o'rnatiladi).

Fayl muvaffaqiyatli ochilganda, fopen funksiyasi fayl descriptorini qaytaradi, aks xolda – false qaytaradi. Fayl Descriptori ochilgan faylga ko'rsatkich bo'lib, operatsion tizim tomonidan shu fayl Bilan amallarni

qo'llash uchun ishlatiladi. Funksiya tomonidan qaytarilgan fayl descriptorini keyinchalik shu fayl bilan ishlaydigan hamma funksiyalarda ko'rsatish lozim.

Quyida keltirilgan kod, C:/WWW/HTML/file.txt faylini o'qish uchun ochadi:

```
<?
$file = fopen("c:/www/html/file.txt","r");
if(!file)
{
    echo("Fayl ochish xatoligi");
}
?>
```

Ikkilik fayl, masalan rasmni ochish shu tariqa b bayrog'i bioan bajariladi:

```
<?
$file = fopen("c:/www/html/river.jpg","rb");
if(!file)
{
    echo("Fayl ochish xatoligi");
}
?>
```

Fayllarni akslantirish. Ochilgan fayldagi ma'lumotlarni brauzerda fpassthru funksiyasi yordamida akslantirish mumkin:

int fpassthru (int file)

Argument file fayl descriptori emasdir.

```
<?
$file = fopen("c:/www/html/pavlovo.jpg","rb");
if(!file)
{
    echo("Fayl ochish xatoligi");
}
else
{
    fpassthru($file);
}
?>
```

Matnli fayllarni akslantirish uchun yana bir readfile akslantirish funksiyasi mavjuddir:

readfile (string filename)

Shuni ta'kidlash lozimki argument sifatida bu funksiya fayl nomini emas, uning descriptorini qabul qiladi:

```
<?
  readfile ("file.txt");
?>
```

Fayllarni berkitish. Fayl bilan ishni tugallagandan so'ng uni yopish kerak, Fayllarni yopish fclose funksiyasi yordamida amalga oshiriladi:

```
int fclose (int file)
Argument file yopish kerak bo'lgan fayl descriptori.
```

17.2. Fayllardan o'qish

Ochiq fayldan qatorni fread funksiyasi yordamida o'qish mumkin:
string fread (int file, int length)

Bu funksiya fayl simvollaridan iborat length uzunlikdagi file descriptorli qatorni qaytaradi.

```
<?
  $file = fopen("c:/www/html/file.txt","r");
  if(!file)
  {
    echo("Fayl ochish xatoligi");
  }
  else
  {
    $buff = fread ($file,100);
    print $buff;
  }
?>
```

Fayldan o'qish uchun fgets funksiyasidan foydalanish ham mumkin:
string fgets (int file, int length)

Bu funksiya length - 1 bayt uzunlikdagi satrni o'qiydi va qaytaradi. Yangi satr yoki fayl oxiriga etilganda o'qish to'xtatiladi. Fayl oxiriga etilganda funksiya bo'sh satr qaytaradi.

HTML teglarini tashlab yuborgan xolda faylni o'qish uchun fgetss funksiyasi qo'llanadi:

```
string fgetss (int file, int length [, string allowable_tags])
```

Shart bo'lmagan uchinchi parametr allowable_tags tashlab yuborilmasligi kerak bo'lgan teglar ro'yxatidan iborat satrni o'z ichiga oladi. Bu satrda teglar vergul bilan ajratiladi.

Agar fayldagi bor ma'lumotlarni massivga yozish kerak bo'lsa, file funksiya si qo'llanadi:

```
array file (string filename [, int use_include_path])
```

Funksiya filename nomli faylni o'qiydi va har bir elementi o'qilgan fayldagi satrga mos keluvchi massiv qaytaradi. Quyidagi misolda funksiya yordamida fayl o'qiladi va undagi bor ma'lumot brauzerga chiqariladi.

```
<?
$file_array = file("file.txt");
if(!$file_array)
{
    echo("Fayl ochish xatoligi");
}
else
{
    for($i=0; $i < count($file_array); $i++)
    {
        printf("%s<br>", $file_array[$i]);
    }
}
?>
```

Bu funksiya qulayligi shundaki uning yordamida fayldagi satrlar sonini hisoblash mumkin:

```
<?
$file_array = file ("file.txt");
if(!$file_array)
{
    echo("Fayl ochish xatoligi");
}
else
{
    $num_str = count($file_array);
    echo($num_str);
}
?>
```

Shuni e'tiborga olish kerakki file funksiyasini faqat kichkina fayllarni o'qish uchun qo'llash kerak.

Kengaytmasi *.csv bo'lgan fayllarni o'qish uchun fgetcsv funksiyasi qo'llanadi:

```
array fgetcsv ( int file, int length, char delim)
```

Funksiya fayldan satr o'qiydi va delim simvoli bo'yicha uni ajratadi. Parametr delim albatta bir simvoldan iborat satr bo'lishi kerak, aks xolda satrlarning faqat birinchi simvoli inobatga olinadi. Funksiya xosil bo'lgan massivni yoki fayl oxiriga etilgan bo'lsa false qiymatni qaytaradi. Bo'sh satrlar tashlab yuborilmaydi, aksincha bir element bo'sh satrdan iborat massiv qaytaradi. Parametr length satrlar maksimal uzunligini qaytaradi, fgets funksiyasidagi kabi.

CSV formati MSEXcel fayllari saqlanuvchi formatlardan biridir. Quyidagi misolda MSEXcel da yaratilgan, foydalanuvchilar parollarini o'z ichiga olgan file.csv o'qiladi.

```
<?
$count = 1;
$file = fopen ("file.csv","r");
while ($data = fgetcsv ($file, 1000, ","))
{
    $num = count ($data);
    $count++;
    for ($i=0; $i < $num; $i++)
    {
        print "$data[$i]<br>";
    }
}
fclose ( $file );
?>
```

17.3. Fayllarga yozish

Fayllarga yozish fputs va fwrite bir xil funksiyalari bilan amalga oshiriladi:

```
int fputs ( int file, string string [, int length ])
int fwrite ( int file, string string [, int length ])
```

Birinchi argument yozuv amalga oshiralayotgan fayl desriptori. Ikkinchi argument faylga yozilishi kerak bo'lgan satr. Uchinchi shart bo'lmagan argument satrda yozilishi kerak bo'lgan simvollar soni. Uchinchi argument ko'rsatilmasa hamma satr yozilishi kerak.

Bu misolda "file.txt" fayliga "Hello, world!" qator yoziladi.

```
<?
$file = fopen ("file.txt","r+");
$str = "Hello, world!";
```

```

if ( !$file )
{
    echo("Fayl ochish xatoligi");
}
else
{
    fputs ( $file, $str);
}
fclose ($file);
?>

```

Fayllardan nusxa olish, nomini o'zgartirish va fayllarni o'chirish. Fayldan nusxa olish copy funksiyasi yordamida amalga oshiriladi:

```
int copy ( string file1, string file2)
```

copy funksiyasi file1 nomli fayldan file2 nomli fayl nusxa oladi.

Agar fayl file2 mavjud bo'lsa, u qaytadan yoziladi.

Fayl nomini o'zgartirish rename funksiyasi yordamida amalga oshiriladi:

```
int rename ( string old, string new)
```

Bu funksiya fayl old nomini new nomiga almashtiradi.

Agar fayl Yangi nomi boshqa fayl tizimida joylashgan bo'lsa rename funksiyasi faylga Yangi nom bermaydi.

Faylni o'chirish unlink funksiyasi yordamida amalga oshiriladi:

```
int unlink ( string filename)
```

Fayllar atributlari. Fayl atributlari haqida qo'shimcha ma'lumot olish uchun quyidagi funksiyalardan foydalanishingiz mumkin.

file_exists funksiyasi fayl mavjudligini tekshiradi va fayl mavjud bo'lsa true, aks xolda false qiymat qaytaradi:

```
bool file_exists ( string filename)
```

fileatime funksiyasi faylga oxirgi murojaat vaqtini qaytaradi:

```
int fileatime ( string filename)
```

filemtime funksiyasi faylning oxirgi o'zgartirish vaqtini qaytaradi:

```
int filemtime ( string filename)
```

file_size funksiyasi fayl xajmini baytlarda qaytaradi:

```
int file_size ( string filename)
```

file_type funksiyasi fayl turini qaytaradi:

```
string file_type ( string filename)
```

Bu funksiya qaytaradigan satr quyidagi fayl turlaridan biriga tegishli bo'ladi:

- char (maxsus simvolli qurilma);
- dir (kattalog);
- fifo (nomlangan kanal);
- link (simvolli ilova);
- block (maxsus blokli qurilma);
- file (oddiy fayl);
- unknown (tur ma'lum emas).

Fayl harakteristikalarini qaytaruvchi funksiyalardan foydalanish ko'p resurslarni talab qilgani uchun, bunday funksiyalarni chaqirishda unumdorlikni yo'qotmaslik uchun, PHP fayl haqidagi ma'lumotni keshlaydi. Bu keshni clearstatcache funksiyasi yordamida tozalash mumkin:

```
<?
clearstatcache();
?>
```

Fayllar bo'yicha ko'chish. Fayldan ma'lumot o'qilganda joriy pozitsiya ko'rsatkichi navbatdagi o'qilmagan simvolga suriladi. Ko'rsatkich xolatini boshqarishga imkon beruvchi bir necha funksiyalar mavjuddir.

Joriy pozitsiya ko'rsatkichini fayl boshiga keltirish uchun rewind funksiyasi qo'llanadi:

```
int rewind ( int file)
file argumenti fayl descriptoridir.
```

Ko'rsatkich joriy pozitsiyasini ftell funksiyasi yordamida aniqlash mumkin:

```
int ftell ( int file)
```

Ko'rsatkichni fayl ixtiyoriy joyiga fseek funksiyasi yordamida o'rnatish mumkin:

```
int fseek ( int file, int offset [, int whence ])
```

Funksiya fseek fayl ko'rsatkichini offset siljishli baytga o'rnatadi (fayl boshidan, oxiridan yoki joriy pozitsiyadan whence paorometri qiymatiga qarab). Argument file fayl descriptoridir. Argument whence qaysi joydan offset siljish hisoblanishi kerakligini aniqlaydi va quyidagi qiymatlardan biriga teng bo'lishi mumkin:

```
SEEK_SET (pozitsiyani fayl boshidan hisoblaydi);
```

```
SEEK_CUR (pozitsiyani ko'rsatkich joriy pozitsiyasidan otschityvaet pozitsiyu otnositelno tekushogo polojeniya ukazatelya);
```

```
SEEK_END (otschityvaet pozitsiyu otnositelno kontsa fayla).
```

Ko'zda tutilgan bo'yicha argument whence qiymati SEEK_SET.

Ko'rsatkich fayl oxirida ekanligini feof funksiyasi yordamida aniqlash mumkin:

```
int feof ( int file)
```

Agar ko'rsatkich fayl oxirida joylashgan bo'lsa, funksiya true qaytaradi aks xolda false.

feof funksiyasini fayl o'qishda ishlatish qulaydir:

```
<?
$file = fopen ("file.txt","r");
if ($file)
{
while(!feof($file))
{
$str = fgets($file);
echo $str;
echo ("<br>");
}
fclose ( $file);
}
else
{
echo("Fayl ochish xatoligi");
}
?>
```

Bu funksiya yordamida faylda satrlar sonini aniqlash qulay:

```
<?
$file = fopen ("file.txt","r");
if ($file)
{
$counter = 0;
while(!feof($file))
{
$str = fgets ($file);
$counter++;
}
echo($counter);
fclose ($file);
}
else
{
```

```
    echo("Fayl ochish xatoligi");
}
?>
```

17.4. Kataloglar bilan ishlash

Joriy kattalogni o'rnatish uchun `chdir` funksiyasi qo'llanadi:

```
int chdir ( string directory)
```

Bu funksiya bilan quyidagicha ishlash mumkin:

```
chdir("/tmp/data"); // absolyut yo'l bo'yicha o'tish
```

```
chdir("./js"); // joriy kattalog ost kattalogiga o'tish
```

```
chdir("../"); // ajdod kattaloga o'tish
```

```
chdir("~/data"); // o'tamiz /home/foydalanuvchi/data (Unix uchun)
```

Joriy kattalogni aniqlash uchun `getcwd` funksiyasidan foydalanish mumkin:

```
string getcwd ( string path)
```

Kattalogni ochish uchun `path` parametri bilan berilgan kattalogni ochuvchi `opendir` funksiyasidan foydalaniladi:

```
int opendir ( string path)
```

Kattalog ochilgandan so'ng, uni `readdir` funksiyasi bilan o'qish mumkin:

```
string readdir ( int dir)
```

Bu funksiya kattalogdagi elementlar nomlarini qaytaradi. Undan tashqari kattaloglarda `"."` va `".."` elementlari mavjud. Birinchi element joriy ikkinchisi ajdod kattaloga ko'rsatadi. Joriy kattalogni nomini `"."` sifatida ko'rsatib ochish mumkin:

```
$dir = opendir (".");
```

Kattalog bilan ish tugagandan so'ng uni yopish kerak. Kattalog yopilishi `closedir` funksiyasi bilan amalga oshiriladi:

```
void closedir ($dir)
```

Quyida joriy kattalogdagi fayllarni o'qish va chiqarishga misol keltirilgan.

```
<?
```

```
$dir = opendir (".");
```

```
echo "Files:\n";
```

```
while ($file = readdir ($dir))
```

```
{
```

```
    echo "$file<br>";
```

```
}
```

```
closedir ($dir);
```

?>

Bu funksiya "." va ".." qiymat ham qaytaradi. Agar bu kerak bo'lmasa, bu qiymatlarni quyidagicha tashlab yuborish mumkin:

<?

```
$dir = opendir (".");
while ( $file = readdir ($dir))
{
  if (( $file != ".") && ($file != ".."))
  {
    echo "$file<br>";
  }
}
closedir ($dir);
```

?>

Ko'rilgan funksiyalarga misol sifatida, c:/temp kattalogidagi bir sutka davomida murojaat qilinmagan hamma fayllarni o'chiruvchi script yaratamiz. Fayllarni o'chirish funksiyasi bu holda rekursiv chaqiriladi.

<?

```
function delTemporaryFiles ($directory)
{
  $dir = opendir ($directory);
  while (( $file = readdir ($dir)))
  {
    if( is_file ($directory."/". $file))
    {
      $acc_time = fileatime ($directory."/". $file);
      $time = time();
      if (($time - $acc_time) > 24*60*60)
      {
        if ( unlink ($directory."/". $file))
        {
          echo ("Fayllar muvaffaqiyatli o'chirilgan");
        }
      }
    }
    else if ( is_dir ($directory."/". $file) && ($file != ".") && ($file !=
".."))
    {
      delTemporaryFiles ($directory."/". $file);
    }
  }
}
```

```

    }
  }
  closedir ($dir);
}
delTemporaryFiles ("c:/temp");
?>

```

Kattaloglarni yaratish mkdir funksiyasi yordamida amalga oshiriladi:
 bool mkdir (string dirname, int mode)

Bu funksiya dirname nomli i mode.murojaat huquqlari bilan kattalog yaratadi. Kattalog yaratilmasa false qaytaradi. Murojaat xquqlari faqat UNIX kattaloglari uchun beriladi, chunki Windows da bu argument ta'sir ko'rsatmaydi. Quyida c:/temp direktoriyasida test v kattalogini yaratish misoli ko'rilgan.

```

<?
$flag = mkdir ("c:/temp/test", 0700);
if($flag)
{
  echo("Kattalog muvaffaqiyatli yaratilgan");
}
else
{
  echo("Kattalog yaratish xatosi ");
}
?>

```

Kattalogni rmdir funksiyasi yordamida o'chirish mumkin:
 bool rmdir (string dirname)
 Endi yaratilgan /test kattalogini o'chiramiz:

```

<?
$flag = rmdir ("c:/temp/test");
if($flag)
{
  echo("Kattalog muvaffaqiyatli o'chirilgan");
}
else
{
  echo("Kattalogni o'chirish xatosi ");
}
?>

```

Funksiya `rmdir` faqat bo'sh kattaloglarni o'chiradi. Bo'sh bo'lmagan kattaloglarni o'chirish uchun, funksiya yaratib `c:/temp` kattalogini undagi papkalar va fayllari bilan birga o'chiramiz:

```
<?
function full_del_dir ($directory)
{
$dir = opendir($directory);
while(($file = readdir($dir)))
{
if ( is_file ($directory."/".$file))
{
unlink ($directory."/".$file);
}
else if ( is_dir ($directory."/".$file) &&
($file != ".") && ($file != ".."))
{
full_del_dir ($directory."/".$file);
}
}
closedir ($dir);
rmdir ($directory);
echo("Kattalog muvaffaqiyatli o'chirilgan");
}
full_del_dir ("c:/temp")
?>
```

Funksiyani rekursiv chaqirganda argumentlar sifatida joriy va ajdod kattaloglarga ko'rsatuvchi `"."` va `".."` yozuvlarni uzatmang, chunki bu holda siz ma'lumotlaringizni yo'qotishingiz mumkin. Bu yozuvlarni shartli operator yordamida o'tkazib yuboring.

NAZORAT SAVOLLARI

1. PHP tilida fayllar bilan ishlash?
2. Fayllar ustida bajariladigan amallar?
3. PHP tilida fayllardan o'qish qanday amalga oshiriladi?
4. PHP tilida fayllarga yozish va kataloglar bilan ishlash?

18-BOB. PHP SINFLARI

18.1. Sinf ma'lumotlar turi misolida

Ob'ektga yo'naltirilgan dasturlashning asosiy tushunchalari – sinflar hamda ob'ektlardir. Bu tushunchalarni quyidagicha tushunish mumkin: ob'ekt – bu dasturda qo'llaniladigan tushuncha yoki biror fizik predmet haqida ma'lumot beradigan strukturalangan o'zgaruvchidir, sinflar esa bu ob'ektlarning tavsifi va ular ustida bajariladigan qarakatlardir.

PHP dasturlash tilida sinflar quyidagi sintaksis yordamida aniqlanadi:

```
class Sinf_nomi{
    var $xususiyat_nomi;
    /* xususiyatlar ro'yxati */
    function metod_nomi(){
        /* usullarning tanasi */
    }
    /*usullar ro'yxati*/
}
```

Sinf ob'ektlari xususiyatlari nomi var kalit so'zi yordamida e'lon qilinadi, berilgan sinf ob'ektlariga qo'llanilgan usullar Funksiya sifatida ishlatiladi. Sinf tanasi ichida this kalit so'zi yordamida taqdim qilinayotgan joriy sinfga murojaatni amalga oshirish mumkin.

Masalan, biz maqola kategoriyasini tasvirlovchi sinf tuzishimiz kerak. har bir maqolaning nomi, muallifi va qisqa mazmuni kabi xususiyatlari bor. Biz maqola bilan qanday amal bajarmoqchimiz? Biz sanab o'tilgan xususiyatlarga ma'no berishimiz, maqolani brauzerda ko'rsatishimiz kerak bo'ladi. Shunda bu sinfnings ifodalanishi quyidagicha holatda bo'ladi:

```
<?
class Articles { //Maqola sinfini tuzamiz
    var $title;
    var $author;
    var $description;
    //maqola atributi ma'nosini o'zlashtiruvchi usul
    function make_article($t, $a, $d){
        $this->title =$t;
        $this->author = $a;
        $this->description = $d;
    }
}
```

```

    }
//sinf nusxasini ifodalash uchun usul
function show_article(){
    $art =$this->title . "<br>" .
        $this->description .
        "<br>Muallif: " . $this->author;
    echo $art;
} } ?>

```

Shunday qilib “maqola” turidagi fizik ob’ektlarni tasvirlash uchun biz uch o’zgartuvchidan tashkil topgan, maqola xarakteristikasini o’zida jamlagan Articles nomli sinf va muayyan maqola tuzish va uni tasvirlash uchun ikkita funksiya tuzdik.

Ma’lumki, PHP bilan ishlash davriy holatda HTML rejimida yuklanishi mumkin. Bu holda dastur bir necha kodning bo’laklari(bloklar)dan tashkil topadi. Sinfni ifodalash php-kodning har xil bloklari bo’yicha va qolaversa har xil fayllar bo’yicha tarqatilmasligi kerak. Ya’ni quyidagicha yozsak:

```

<?php
class Articles { //Sinfni tasvirlashning boshi
    var $title;
?>
<?php //sinfni tasvirlashning davomi
    function show_article(){ //usulning tarkibi
    }
} //sinfni tasvirlashning yakuni
?>

```

bunda dastur tartibli ishlaydi.

Sinf nomi masalasida ayrim narsalarni e’tiborda tutish kerak. Sinfning nomi PHP tilidagi ob’ektlar nomlanishi qoidalariga javob berishi lozim, lekin bir qator nomlar borki, texnik mutaxassislar tomonidan o’z maqsadi uchun zaxira qilinadi. Birinchi navbatda bu nomlar “_” quyi chiziqdan boshlanuvchilardir. Sinflar va funksiyalar tuzish uchun bu kabi nomlarni ishlatmaslik kerak. Bundan tashqari stdClass nomi zaxira qilingan, zero u PHP surilgichi ichida ishlatiladi.

18.2. O’zgaruvchilarni initsiallashtirish

Ba’zan ayrim sinf atributlariga ma’noni sinf ishtirokchisini tuzish bilanoq o’zlashtirish kerak bo’ladi. Biz maqola sinfini tuzganimizda, sinf

atributlari (xususiyatlari) ma'nolarini o'zlashtirish uchun maxsus funksiya `make_article()` dan foydalandik. Umuman olganda, biz to'g'ri yo'l tutmadik, chunki "velosiped ixtirosi" bilan shug'ullandik. Sinf atributlarining boshlang'ich ma'nolarini berish uchun maxsus ikkita standart usul mavjud. PHP4da ma'noni var operatori yoki konstruktor funksiyasi yordamida initsiallashtirish mumkin. var yordamida faqat konstant ma'nolarni initsiallashtirish mumkin. Konstant bo'lmagan ma'nolarni berish uchun ob'ekt sinfdan ajrab chiqqanda o'z-o'zidan ishga tushuvchi konstruktor funtsiyasidan foydalaniladi. Konstruktor-funksiya u ifodalangan butun sinfga mos keluvchi nomga ega bo'lishi kerak.

Misol. "maqola" deb nomlangan ob'ektni tuzishda uning xususiyatlarini quyidagicha belgilash mumkin: mualliflar – "Kamolov" satriga teng, nomlanish va qisqa mazmun - `$_POST` global massivi elementlariga mos, maqola nashri – mazkur sanada.

```
<?
class Articles { //maqola sinfini tuzish
    var $title;
    var $author = "Kamolov";
    var $description;
    var $published; //sinf atributi ma'nosini o'zlashtiruvchi usul
function Articles(){
    $this->title = $_POST["title"];
    $this->description = $_POST["description"];
    $this->published = date("Y-m-d");
} } ?>
```

PHP3 va PHP4 da konstruktorlar har xil ishlashini hisobga olish kerak. Funksiya PHP3 da, agar u sinfniki kabi nomga ega bo'lsa, konstruktorga aylanadi, PHP4 da esa – agar u ifodalangan sinfniki kabi nomga ega bo'lsa shunday bo'ladi. Bir sinf boshqasini kengaytirganda va xususiyatlarning va baza sinflar usullarining ergashishida usullar orasidagi farq ko'rinib turibdi. Lekin biz bu haqida biroz keyinroq gapiramiz. PHP5da sinf konstruktori `_construct` deb nomlanadi. Bundan tashqari, PHPda destruktorga – ob'ektni yo'q qilishda o'z-o'zidan ishga tushuvchi Funksiyalar paydo bo'ldi. PHP5 da Funksiya-destruktor `destruct` deb nomlanishi kerak bo'ladi.

18.3. Sinflar bilan ishlash

Dasturda ayrim sinflar yozilgan deb tasavvur qilamiz. Sinf – bu mazmuni bo'yicha ma'lumotlar turi, bu tur qiymatlarini saqlaydigan o'zgaruvchilar yaratish uchun ayrim mexanizmlarga ega bo'lishimiz kerak.

Ayrim sinflar ob'yektini yaratish. Sandart turlardan foydalanishdagi ishlarni esga oling:

```
// Son-o'zgaruvchini yaratish.
```

```
$number = 10.4;
```

```
// Satrni yaratish.
```

```
$str = "Some string";
```

```
// Massivni yaratish.
```

```
$arr = array(1, 2, 3);
```

Foydalanuvchi ma'lumot turiga ega bo'lgan o'zgaruvchilarni yaratishda, sinf nomi belgilanadigan new kalit so'zi qo'llaniladi:

```
$obj = new Math_Complex;
```

Endi \$obj sinfning barcha ma'lumotlarini saqlaydi – odatda, o'zida \$re va \$im alohida qiymatlarini saqlaydi.

Ob'yekt xususiyatlariga kirish. Avval aytilganidek, har bir ob'yekt o'zining qiymat yoli o'zgaruvchilari va ko'plab usullariga ega. Dasturda ob'yektning har bir xususiyatiga uning nomi bilan kirish mumkin. Xususiyat qiymati yoki uning kengligini o'zlashtirish mumkin:

```
// Math_Complex sinfi yangi ob'yektini yaratamiz.
```

```
$obj = new Math_Complex;
```

```
// $obj ob'yektning $re va $im xususiyatlariga qiymat o'zlashtiradi.
```

```
$obj->re = 6;
```

```
$obj->im = 101;
```

```
// $obj ob'yektning re xususiyati qiymatini chiqaradi.
```

```
echo $obj->re;
```

Ko'rib turganingizdek, xususiyatga kirish - > operatori yordamida amalga oshiriladi.

Usullarga kirish. O'rnatilgan ma'lumotlar turining “usullar” ini chaqirishni eslab ko'ramiz:

```
// Sonlarni qo'shish: + operatori.
```

```
$c = $a + $b;
```

```
// Qism satrni olish: substr() amali.
```

```
$sub = substr($str, 1, 20);
```

Ko'rib turganingizdek, встроенных turlar uchun “usul” chaqirishni yozish operatori (masalan, so'shish), yoki bo'lmasam funksional chaqirish

(funksiya chaqirilgandek) dan foydalaniladi. PHP da ayrim ob'yektlarni chaqirish usuli uchun "strelka" operatoridan foydalaniladi.

```
<?php ## Ob'yekt usulini chaqirish.  
// includejpath kutubxonasiga kataloglarni ulash.  
require_once "lib/config.php";  
// Sinfni yuklash.  
require_once "Math/Complex.php";  
// Math_Complex sinfi yangi ob'yektini yaratamiz.  
$obj = new Math_Complex;  
// Xususiyatlarga boshlang'ich qiymatni o'zlashtiramiz.  
$obj->re = 16.7;  
$obj->im = 101;  
// $obj ob'yektining (18.09, 303) o'lchamlar bilan add() usulini  
chaqirish.  
$obj->add(18.09/ 303);  
// Natijani chiqaramiz:  
echo "({$obj->re}, {$obj->im})";  
?>
```

Bir nechta ob'yektlarni yaratish. Biz yuqoridagi misolda add usuliga oddiy sonni kiritdik. Odatda, funksiya parametri sifatida istalgan narsani ko'rsatish mumkin, masalan, boshqa sinf ob'yektini.

add() usulini qisman o'zgartirgan holda Mathcomplexi yana bir sinfni yaratamiz. Bundan tashqari, har safar qo'lda ko'rsatmaslik uchun murakkab sonni satrli olish uchun usul qo'shdik.

```
<?php ## Usulli sinfga misol,  
class Math_Complex1 {  
public $re, $im;  
// Mavjud murakkab songa qo'shadi,  
function add(Math_Complex1 $y) {  
$this->re += $y->re;  
$this->im += $y->im;  
} // Sonni satrga o'zgartiradi.  
function toStringO {  
return "({$this->re}, {$this->im})";  
}  
}  
?>
```

Qarang, \$y parametrdan oldin Math_complexi turini ko'rsatdik. Bu PHP ga shuni bildiradiki, berilgan funksiyaga faqatgina shu ob'yekt

sinflarini uzatishimiz mumkin, boshqasini emas. Masalan, \$y o'rniga butun sonni ko'rsatilganda, dastur bajarilishi davomida xatolik yuzaga keldi:

```
$obj->add(1);  
Fatal error: Argument 1 must be an object of class Math_Complex1  
Bu sinfdan foydalanishning to'g'ri variant quyida keltirilgan.  
<?php ## Ob'yekt usulini chaqirish.  
require_once "lib/config.php";  
require_once "Math/Complex1.php";  
// Birinchi ob'yektni yaratamiz.  
$a = new Math_Complex1;  
$a->re = 314;  
$a->im = 101;  
// Ikkinchi ob'yektni yaratamiz.  
$b = new Math_Complex1;  
$b->re = 303;  
$b->im = 6;  
// Bitta qiymatni boshqasiga qo'shamiz.  
$a->add($b);  
// Natijani chiqaramiz:  
echo $a-> toString();  
?>
```

NAZORAT SAVOLLARI

1. PHP tilida sinflar bilan ishlash?
2. O'zgaruvchilarni inisiallashtirish?
3. PHP tilida sinflar va ular bilan ishlash?

19-BOB. MA'LUMOTLAR BAZASI. MA'LUMOTLAR BAZASINI BOSHQARISH TIZIMI

19.1. Ma'lumotlar bazasi asoslari. Ma'lumotlar bazasini boshqarish tizimi (MBBT)

Ushbu bo'lim PHP va MySQL MBBT o'rtasidagi hamkorlik usullari bilan tanishishga mo'ljallangan. Asosiy e'tibor ma'lumotlar bazasi bilan bog'lanishni o'rnatish, so'rovlar jo'natish funksiyalari va javoblarni (mysql_connect, mysql_query, mysql_result, mysql_num_rows, mysql_close) qayta ishlashga qaratiladi. Misol sifatida virtual tarix muzeyi ma'muriyati uchun web-interfeys tuzish masalasini ko'raylik. PHP distributivida MySQL ma'lumotlar bazasi bilan ishlash uchun mo'ljallangan funksiyalar mavjud. Bunda bu funksiyalarning MySQL dagi ba'zi bir ma'lumotlar bazasini tasvirlash va to'ldirish maqsadida web-interfeyslarni tuzish imkonini beruvchi funksiyalar bilan tanishamiz. Ma'lumotlar bazasiga ma'lumotlarni qo'shish uchun web-interfeys bilan ishlashda bu ma'lumotlarni shunchaki html-formaga kiritish va ularni serverga jo'natish kerak bo'ladi.

Namoyish etishda bu interfeysni virtual muzey eksponatlari haqidagi ma'lumotlar saqlanadigan Artifacts jadvallari uchun tuzamiz. Artifacts kolleksiyasidagi har bir eksponat quyidagi tavsif yordamida tasvirlanishini eslatib o'tamiz:

- nom (title);
- muallif (author);
- ifoda (description);
- o'rindosh nom (alternative);
- tasvir (photo).

Nomlanish va o'rindosh nomlanish uzunasiga 255 belgidan kam satr (ya'ni VARCHAR(255)), tasvirlash – matnli maydon (TEXT turiga mansub) hisoblanadi, “muallif” va “tasvir” maydonlarida esa Persons kolleksiyasidan muallifning identifikatorlari va Images kolleksiyasidan eksponat tasvirlariga muvofiq mavjud bo'ladi.

Ma'lumotlar bazasi bilan bog'lanish (MySQL dasturi misolida)

Aloqa o'rnatish. Ma'lumotlar bazasi bilan aloqa o'rnatish uchun mysql_connect funksiyasidan foydalaniladi.

mysql_connect sintaksisi

mysql_connect resursi (“server qatori”, “username”, “password”)

Bu funksiya MySQL serveri bilan aloqa o'rnatadi va bu aloqaga ko'rsatkich qaytaradi yoki muvaffaqiyatsiz chiqqanda FALSE ko'rsatadi. Odatda quyidagi parametrlar qiymati e'lon qilinadi:

```
server = 'localhost:3306'
```

```
username = server jarayoni egasidan foydalanuvchi ismi
```

```
password = bo'sh parol
```

Server bilan ulanish, agar u bungacha mysql_close() yordamida yopilgan bo'lsa, skriptni amalga oshirish tugallanishida baza bilan aloqa yopiladi.

Misol:

```
<?
```

```
$conn = mysql_connect("localhost", "nina","123") or  
diye("Ulanishni amalga oshirib bo'lmaydi: ". mysql_error());
```

```
echo "Ulanish amalga oshdi";
```

```
mysql_close($conn);
```

```
?>
```

mysql_connect amali

shell>mysql -u nina -p123 buyrug'i bilan teng kuchli.

19.2. MySQL ma'lumotlar bazasi boshqarish tizimi

MySQL da ma'lumotlar bazasini tanlash use buyrug'i yordamida amalga oshiriladi:

```
mysql>use book;
```

PHP da buning uchun mysql_select_db funksiyasi mavjud.

mysql_select_db: sintaksisi

```
mantiqiy mysql_select_db (database_name qatori);
```

Bu funksiya TRUE qiymatni ma'lumotlar bazasini muvaffaqiyatli tanlanganda qaytaradi va FALSE ni esa – aksincha bo'lganda.

Misol: Book ma'lumotlar bazasini tanlash

```
<?
```

```
$conn = mysql_connect ("localhost","user","123") or diye  
("Ulanishni amalga oshirib bo'lmaydi: ". mysql_error());
```

```
echo "Ulanish amalga oshdi";
```

```
mysql_select_db("book");
```

```
?>
```

Ma'lumotlar bazasiga ma'lumotlar yozish. Ma'lumki, ma'lumotlarni jadvalga yozish uchun SQL tilidagi INSERT buyrug'i ishlatiladi:

```
mysql> INSERT INTO Artifacts SET title='Eksponat nomi';
```

PHP skriptda bunday buyruqdan foydalanish uchun `mysql_query()` funksiyasi mavjud.

`mysql_query` sintaksisi

mysql_query resursi (query qatori)

`mysql_query()` SQL-so'rovni MySQL ma'lumotlar bazasining ma'lumotlar bazasiga jo'natadi. Agar ochiq aloqa bo'lmasa, funksiya parametrsiz `mysql_connect()` funksiyasiga o'xshash holda MBBT bilan bog'lanishga urinadi.

So'rov natijasi buferlanadi.

19.3. MySQL bilan ishlash interfeysi

PHP dan MySQL da interfeys sifatida foydalanishdan maqsad SQL so'rovlar natijalarini formatlash va ularni web sahifada ifodalashdan iborat. O'rnatilgan MySQL tizimga parol va login bilan kirib bajariladigan ishlarni PHP bilan ham amalga oshirish mumkin bo'ladi. Bunda MySQL komandalar satridan foydalanish o'rniga, so'rovlar yozish va uni MySQL ga uzatish kerak bo'ladi. MySQL javobi ma'lumotlar strukturasi ko'rinishida bo'ladi, bu holda PHP uni tanib oladi va qayta ishlashi uchun qulay bo'ladi. Undan keyin PHP qayta ishlaydi va web sahifa formatiga aylantiradi. MySQL dan PHP yordamida foydalanish jarayoni quyidagilardan iborat bo'ladi:

1. MySQL ga bog'lanish.
2. Foydalanmoqchi bo'lgan ma'lumotlar bazasini tanlash.
3. So'rovlar satrini hosil qilish.
4. So'rovlarni bajarish.
5. Natjalarni olish va web sahifaga chiqarish.
6. 3 va 5 punktlarni barcha zaruriy ma'lumotlarni olguncha takrorlash.
7. MySQL ni yopish. Undan keyingi jarayonlar bosqichma bosqich amalga oshiriladi, eng avvalo tizimga xavfsiz kirishni amalga oshirish kerak bo'ladi.

Registratsiya faylini hosil qilish. Ko'pgina saytlar PHP da ishlangan MySQL ma'lumotlar bazasi bilan ishlaydi, unga murojaat qilish uchun parol va login kerak bo'ladi. Shuning uchun alohida fayl kerak bo'ladi, bu fayl nomini odatda `login.php` deb atash mumkin. Quyida shu fayl matni berilgan.

Fayl `login.php`

```
<?php // login.php
$db_hostname = 'localhost';
$db_database = 'publications';
$db_username = 'foydalanuvchi_nomi';
$db_password = 'parol';
?>
```

Bu faylda asosiy rolni qamrab oluvchi teglar o'ynaydi <?php va ?>, bu teglar orasidagi satrlar PHP kodlar deb interpretatsiya qilinadi. Agar qamrab oluvchi teglar qo'yilmasa saytda fayl chaqirilganda barcha kodlar matn ko'rinishida paydo bo'lishi mumkin. Agar hammasi joyida bo'lsa, saytda bo'sh satr paydo bo'ladi. Bu fayllarni hech qanday muammosiz boshqa PHP-fayllarga qo'shish mumkin bo'ladi.

\$db_hostname o'zgaruvchi PHP ga qaysi kompyuter ma'lumotlar bazasi bilan ishlashi haqidagi xabarni jo'natadi. Bunday o'zgaruvchining mavjudligi siz barcha kompyuterlardagi MySQL ma'lumotlar bazasiga ruxsat olishingiz mumkinligini bildiradi.

NAZORAT SAVOLLARI

1. Ma'lumotlar bazasi va ularni yaratish?
2. MySQL ma'lumotlar bazasini boshqarish tizimi bilan ishlash?
3. PHP da MySql bilan bog'lanish funksiyasi?
4. Ma'lumotlar bazalarini tanlash qanday amalga oshiriladi?
5. Ma'lumotlar bazasiga ma'lumotlar yozishni amalga oshirish?
6. PHP da MySql so'rovlarini amalga oshirishga misol keltiring?

20-BOB. MA'LUMOTLAR BAZASIGA SQL SO'ROVLAR TILI

20.1. Jadval yaratish

Jadvallar CREATE TABLE komandasi bilan yaratiladi. Bu komanda qatorlarsiz bo'sh jadval yaratadi. CREATE TABLE komandasi jadval nomini va jadval o'zini ma'lum tartibda ko'rsatilgan ustunlar nomlari ketma – ketligi ta'rifi ko'rinishida aniqlaydi. U ma'lumotlar turlari va ustunlar o'lchovini aniqlaydi. Har bir jadval juda bo'lmaganda bitta ustunga ega bo'lishi kerak.

CREATE TABLE komandasi sintaksisi:

CREATE TABLE <table-name >

(<column name> <data type>[(<size>)],

<column name> <data type>[(<size>)], ...);

Argument qiymati kattaligi ma'lumot turiga bog'liqdir. Agar siz maxsus ko'rsatmasangiz, tizim avtomatik qiymatni o'rnatadi.

Bundan buyon quyida keltirilgan 3 ta jadvaldan iborat ma'lumotlar bazasini ko'ramiz.

20.1-jadval. Sotuvchilar (Salepeople):

| Snum | Sname | City | Comm |
|------|---------|--------|------|
| 11 | Alisher | Jomboy | 0.12 |
| 12 | Rustam | Urgut | 0.13 |
| 14 | Nodir | Toyloq | 0.11 |

SNum – har unikal nomeri,

SName – sotuvchi nomi,

City – sotuvchi adresi (shahar),

Comm – sotuvchilarning o'nli shakldagi komission foydasi.

bir sotuvchi

20.2-jadval. Buyurtmachilar (Customers):

| Cnum | Cname | City | Rating | SNum |
|------|----------|--------|--------|------|
| 21 | Shuxrat | Chelak | 100 | 11 |
| 22 | G'ayrat | Toyloq | 200 | 13 |
| 23 | Ulug'bek | Urgut | 200 | 12 |

CNum – har bir buyurtmachi unikal nomeri,
 CName – buyurtmachi nomi,
 City – buyurtmachi adresi (shahar),
 Rating – buyurtmachining boshqalardan ustunlik darajasini ko'rsatuvchi kod (reyting),
 SNum – shu buyurtmachiga tayinlangan sotuvchi nomeri.

20.3-jadval. Buyurtma (Orders):

| Onum | AMT | Odate | CNum | Snum |
|------|-------------|------------|------|------|
| 38 | 4723.0
0 | 1990/10/05 | 26 | 11 |
| 310 | 1309.9
5 | 1990/10/06 | 24 | 12 |

ONum – har bir sotib olish unikal nomeri,
 AMT – sotib olish summasi qiymati,
 ODate – sotib olish sanasi,
 CNum – sotib oluvchi buyurtmachi nomeri,
 SNum – sotuvchining nomeri.

Misol uchun sotuvchilar jadvalini yaratishni ko'rib chiqamiz:

```
CREATE TABLE Salepeople
```

```
( SNum integer,  

  SName char (10),  

  City char (10),  

  Comm decimal );
```

Jadvallarni o'chirish. Jadvalni o'chirish imkoniga ega bo'lish uchun, jadval egasi (Ya'ni yaratuvchisi) bo'lishingiz kerak. Faqat bo'sh jadvalni o'chirish mumkin. Qatorlarga ega bo'lgan, to'ldirilgan jadvalni o'chirish mumkin emas, Ya'ni jadval o'chirishdan oldin tozalangan bo'lishi kerak. Jadvalni o'chirish komandasi quyidagi ko'rinishga ega:

```
DROP TABLE < table name >;
```

Masalan: DROP TABLE Salepeople;

Jadvalni yaratilgandan so'ng o'zgartirish. Jadvalni o'zgartirish uchun ALTER TABLE komandasidan foydalaniladi. Bu komanda jadvalga Yangi ustunlar qo'shish, ustunlarni o'chirish, ustunlar kattaligini o'zgartirish, hamda cheklanishlarni qo'shish va olib tashlash

imkoniyatlariga ega. Bu komanda ANSI standarti qismi emas, shuning uchun har xil tizimlarda har xil imkoniyatlarga ega.

Jadvalga ustun qo'shish uchun komandaning tipik sintaksisi:

```
ALTER TABLE <table name> ADD <column name>  
      <data type> <size>;
```

Masalan:

```
ALTER TABLE Salepeople ADD Phone CHAR(7);
```

20.2. Jadvallar uchun cheklanishlar

Jadval yaratayotganingizda (yoki uni o'zgartirayotganingizda), siz maydonlarga kiritilayotgan qiymatlarga cheklanishlar o'rnatishingiz mumkin. Bu xolda SQL cheklanishlarga to'g'ri kelmaydigan hamma qiymatlarni rad etadi. Cheklanishlar ikki asosiy turi mavjud: - ustun va jadval cheklanishlari. Ularning farqi shundaki ustun cheklanishi stolbtsa faqat ayrim ustunlarga qo'llanadi, jadval cheklanishi bo'lsa bir yoki bir necha ustunlar guruxiga qo'llanadi. Ustun cheklanishi ustun nomi oxiriga ma'lumotlar turidan so'ng va verguldan oldin qo'yiladi. Jadval cheklanishi jadval nomi oxiriga so'nggi dumaloq verguldan oldin qo'yiladi. Cheklanishlar hisobga olingan CREATE TABLE komandasi sintaksisi:

```
CREATE TABLE < table name >  
  ( <column name> <data type> <column constraint>,  
    <column name> <data type> <column constraint> ...  
    <table constraint> ( <column name>  
    [, <column name> ])... );
```

Maydonga bo'sh (NULL) qiymatlar kiritilishi oldini olish uchun CREATE TABLE komandasida NOT NULL cheklanishi ishlatiladi. Bu cheklanish faqat har xil ustunlar uchun o'rnatiladi.

Masalan, shu narsa aniqki, birlamchi kalitlar hech qachon bo'sh bo'lmasliklari kerak, shuning uchun Salepeople jadvalini quyidagicha yaratish mumkin:

```
CREATE TABLE Salepeople  
  ( Snum   integer NOT NULL,  
    Sname  char (10),  
    city   char (10),  
    comm   decimal);
```

Ko'p xollarda ustunga kiritilgan qiymatlar bir biridan farq qilishi kerak. Agar ustun uchun UNIQUE cheklanishi o'rnatilsa, bu ustungsha mavjud qiymatni kiritishga urinish rad etiladi. Bu cheklanish bo'sh

bo'lmaydigan (NOT NULL) debe'lon qilingan maydonlarga qo'llanishi mumkin.

Masalan:

```
CREATE TABLE Salepeople
  ( SNum integer NOT NULL UNIQUE,
    SName char (10),
    City char (10),
    Comm decimal);
```

Unikalligi talab qilinadigan maydonlar(birlamchi kalitlardan tashqari) kandidat kalitlar yoki unikal kalitlar deyiladi.

Jadval cheklanishi UNIQUE maydonlar guruhiga o'rnatilishi mumkin. Bu bir necha maydonlar qiymatlari kombinatsiyasi unikalligini ta'minlaydi. Bixning ma'lumotlart bazamizda har bir buyurtmachi bita sotuvchiga biriktirilgan. Ya'ni Buyurtmachilar jadvalida buyurtmachi nomeri (cnum) va sotuvchi nomeri (snum) kombinatsiyasi unikal bo'lishi kerak. Bu cheklanishni UNIQUE (cnum, snum) yordamida, Customers jadvalini yaratishda kiritish mumkin. Bu ustunlar uchun NOT NULL cheklanishini kiritish zarurdir.

Birlamchi kalitlar cheklanishlari. SQL birlamchi kalitlarni to'g'ridan to'g'ri birlamchi kalit (PRIMARY KEY) cheklanishi orqali ta'riflaydi. PRIMARY KEY jadvalni yoki ustunlarni cheklashi mumkin. Bu cheklanish UNIQUE cheklanishi kabi ishlaydi, faqat jadval uchun faqat bitta birlamchi kalit (ixtiyoriy sondagi ustunlar uchun) aniqlanishi mumkin bo'lgan xoldan tashqari. Birlamchi kalitlar NULL qiymatga ega bo'lishi mumkin emas.

Misol:

```
CREATE TABLE Salepeople
  ( SNum integer NOT NULL PRIMARY KEY,
    SName char (10),
    City char (10),
    Comm decimal);
```

PRIMARY KEY cheklanishi qiymatlar unikal kombinatsiyasini tashkil qiluvchi bir necha maydonlar uchun qo'llanishi mumkin. Masalan PRIMARY KEY cheklanishini juftliklar uchun qo'llash mumkin:

```
CREATE TABLE Namefield
  ( firstname char (10) NOT NULL,
    lastname char (10) NOT NULL
    city char (10),
```

PRIMARY KEY (firstname, lastname));

20.3. SQL tilida maydon qiymatlarini tekshirish (CHECK cheklanishi)

Maydon qiymatlarini tekshirish (CHECK cheklanishi) - CHECK cheklanishi jadvalga kiritilayotgan ma'lumot qabul qilinishidan oldin mos kelishi lozim bo'lgan shart kiritishga imkon beradi. CHECK cheklanishi CHECK kalit so'zi ko'rsatilgan maydondan foydalanuvchi predikat ifodapdan iboratdir.

Misol: Salepeople jadvali Comm ustuniga kiritilayotgan qiymat 1 dan kichik bo'lish sharti.

```
CREATE TABLE Salepeople  
( SNum integer NOT NULL PRIMARY KEY,  
  SName char(10) NOT NULL UNIQUE,  
  City char(10),  
  Comm decimal CHECK ( Comm < 1 ));
```

CHECK cheklanishidan maydonga ma'lum qiymatlarini kiritishdan ximoya qilib, xatolar oldini olish uchun foydalanish mumkin. Masalan maxsulotni sotish shaxobchalariga ega bo'lgan shaharlar faqat London, Barselona, San Xose va Nu York bo'lsin.

```
CREATE TABLE Salepeople  
( SNum integer NOT NULL PRIMARY KEY,  
  SName char(10) NOT NULL UNIQUE,  
  City char(10) CHECK (City IN ('London', 'New York',  
'San Jose', 'Barselona')),  
  Comm decimal CHECK ( Comm < 1 ));
```

CHECK jadval cheklanishi sifatida kelishi mumkin. Bu shartga bir necha maydon kiritishga imkon beradi.

Masalan:

```
CREATE TABLE Salepeople  
( SNum integer NOT NULL PRIMARY KEY,  
  SName char(10) NOT NULL UNIQUE,  
  City char(10),  
  Comm decimal,  
  CHECK (Somm < .15 OR City = 'Barcelona'));
```

Ko'zda tutilgan qiymatlarni o'rnatish-biror bir maydon uchun qiymat ko'rsatmagan xolda jadvalga satr qo'shsangiz, SQL bunday maydonga kiritish uchun ko'zda tutilgan qiymatga ega bo'lishi kerak, aks xolda

komanda rad etiladi. Eng umumiy ko'zda tutilgan qiymat NULL qiymatdir. CREATE TABLE komandasida ko'zda tutilgan qiymat DEFAULT operatori orqali, ustun cheklanishi sifatida ko'rsatiladi. Masalan:

```
CREATE TABLE Salepeople
( SNum   integer NOT NULL PRIMARY KEY,
  SName  char(10) NOT NULL UNIQUE,
  City   char(10) DEFAULT 'New York',
  Comm   decimal CHECK ( Comm < 1 ));
```

Ma'lumotlar yaxlitligini ta'minlash- jadval bir maydonidagi hamma qiymatlar boshqa jadval maydonida aks etsa, birinchi maydon ikkinchisiga ilova qiladi deyiladi. Bu ikki maydon orasidagi bog'liqlikni ko'rsatadi. Masalan, buyurtmachilar jadvalida har bir buyurtmachi, sotuvchilar jadvalida o'ziga biriktirilgan sotuvchiga ilova qiluvchi SNum maydoniga ega. Bir maydon ikkinchisiga ilova qilsa tashqi kalit, u ilova qilayotgan maydon ajdod kalit deyiladi. Buyurtmachilar jadvalidagi SNum Maydoni tashqi kalit, sotuvchilar jadvalidagi SNum - ajdod kalitdir.

Tashqi kalit bita maydondan iborat bo'lishi shart emas. Birlamchi kalit kabi, tashqi kalit bita modul sifatida qayta ishlanuvchi bir necha maydonlarga ega bo'lishi mumkin. Maydon tashqi kalit bo'lsa ilova qilayotgan jadval bilan ma'lumi usulda bog'liqdir. Tashqi kalit har bir qiymati (satri), ajdod kalitning bita va faqat bitta qiymatiga(satri) ilova qilishi kerak. Bu xolda tizim ilovali yaxlit xolatda deyiladi

Shu bilan birga ajdod kalit qiymati tashqi kalit bir necha qiymatlariga ilova qilishi mumkin.

Cheklanish FOREIGN KEY - SQL ilovali yaxlitlikni FOREIGN KEY yordamida ta'minlaydi. Tashqi kalit vazifasi ajdod kalitda ko'rsatilmagan qiymatlarni tashqi kalitmaydonlariga kiritmaslikdir. FOREIGN KEY cheklanishi sintaksisi:

```
FOREIGN KEY <column list> REFERENCES
      <pktable> [<column list>]
```

Birinchi ro'yxat komanda tomonidan o'zgartiriluvchi ustunlar ro'yxatidir. Pktable - bu ajdod kalitli jadval. Ikkinchi ustunlar ro'yxati bu ajdod kalitni tashkil qiluvchi ustunlardir.

Misol uchun Sotuvchilar jadvaliga ilova qiluvchi tashqi kalit sifatida e'lon qilingan SNum maydoniga ega bo'lgan Buyurtmachilar jadvalini yaratamiz:

```
CREATE TABLE Customers
```

```
( CNum integer NOT NULL PRIMARY KEY,  
  CName char(10),  
  City char(10),  
  SNum integer,  
  FOREIGN KEY (SNum) REFERENCES Salepeople (SNum) );
```

*

NAZORAT SAVOLLARI

1. Jadvallar va ularni yaratish ketma-ketligi?
2. Yaratilgan jadvallarga qanday cheklanishlar mavjud?
3. SQL tilida maydon qiymatlarini tekshirish qanday amalga oshiriladi?

21-BOB. WEB SAYTLARNI YARATISHDA CMS TEKNOLOGIYASINI QO'LLASH

21.1. CMS lar turlari va ularning paydo bo'lishi

Barchaga ma'lumki, butun jahon tarmog'idagi web-saxifalar kundan kunga ko'payib bormoqda. Bu xolat O'zbekistonda, ya'ni Uznetda ham jadallik bilan amalga oshmoqda. Kimdir o'z shaxsiy saxifasini yaratsa, kompaniya va firmalar korporativ saytlarini yaratmoqda. Yana kimdir o'z kundaliklarini, ya'ni bloglarini olib boradilar.

Shu bilan birga, yirik portallar faoliyati ham rivojlanmoqda. Bularning hammasi esa biror dasturiy vosita yordamida yaratiladi. Kimdir HTML tilida, kimdir PHP da, yana kimlardir CMS deb ataluvchi tizimlar yordamida Web- saytlarni yaratmoqda. Xo'sh, CMS degani o'zi nima. Bu tizimni nima ekanini tushunish uchun, keling, avvalo, web-saxifalar qanday yaratilishi haqida bilib olamiz.

1989-yilda Tim Berners-Li tomonidan World Wide Web, ya'ni butunjaxon o'rgimchak to'ri ixtiro qilinganidan boshlab uning har bir foydalanuvchisi oddiy bo'lsa-da, o'z web-saxifasini yarata olish imkoniyatiga ega bo'ldi. Lekin, buning uchun HTML (Hyper Text Markup Language) tilini bilish shart edi. Turli xil IT- kompaniyalari tomonidan bu ishni osonlashtiruvchi bir qancha dasturiy paketlar, jumladan Adobe Dreamweaver va Microsoft FrontPage kabilar yaratildi. Bunday dasturlardan ba'zilar sayt yaratishning barcha bosqichlarini (HTML-kodni yozishdan bu kodni serverga yuklashgacha) boshqarishni o'z zimmasiga oladi. SHu bilan birga, ko'pgina xollarda (aynan oddiy saxifalarni yaratishda) foydalanuvchi bu kodni bilishi shart bo'lmagan. Sababi bu dasturlarda web- saxifalarni yaratishning vizual usullaridan foydalanilgan.

Biroq, HTML-redaktorlarining barcha afzalliklariga qaramay, saytni boshqarish vaqt o'tgani sayin va yangi ma'lumotlar qo'shilishi bilan qiyinlashib boraverdi. Chunki axborot texnologiyalari rivojlanishi bilan web-saytlar ham rivojlanib bordi. Sayt haritasini yaratishdek oddiy funksiyani qo'shish uchun ham web-yaratuvchilar tomonidan ancha-muncha mexnat talab etilar edi. Axir buning uchun barcha ilovalarni ko'rib chiqish, saxifalar orasidagi bog'liqliklarni tekshirish, menyuni yaratish, forumni qo'shish kabi ishlarni bajarish lozim edi. Bu esa doimiy diqqatni va juda ko'p vaqtni talab etardi.

CMS "Content Management System" deb yoyib yoziladi, ya'ni kontentni boshqarish tizimi yoki saytni boshqarish tizimi. Bu saytni tez

yaratish, Bitrix (Saytni Boshqarish Tizimi) ishini soddalashtirish uchun xizmat qiladi. Birinchi: saytni boshqarish tizimi buyurtmachiga saytni mustaqil ekspluatatsiya qilishda, ya'ni bizning mutaxassis bilan emas, balki o'zingizning kompaniya xizmatchingiz (menedjer, kotib, marketolog) bilan saytdagi ixtiyoriy ma'lumotni va uning strukturasi o'zgartirishingizga imkon beradi.

Agar siz saytingiz marketingli uskuna va undagi har kungi ma'lumotida kompaniyangizning oxirgi marketingli aksiyalari aks etishini xoxlasangiz u holda CMS ni sizning saytingizga qo'llanilishi shart. Bu - saytingizni rivojlantirish yoki modernizatsiyalash ishlarida o'ylanmay qilingan Xarakatlar yoki shunchaki xatolardan asraydi.

Ikkinchi: Saytni boshqarish tizimi yordamida saytni rivojlanishi sezilarli soddalashadi va jonlanadi. Saytda yangi bo'lim yaratish, ixtiyoriy saxifada reklama bloklarini, ichki bannerlarni yoki tovarlarning maxsus takliflarini joylashtirishni buyurtlachi mustaqil va daqiqalar bilan o'lchanadigan vaqt mobaynida bajarishi mumkin. Bundan tashqari, buyurtmachi mustaqil, bizning yordamimizsiz, CMS yordamida saytni yangi interfaol bo'limlarini yaratishi (masalan, forum) yoki u yoki bu saxifalarni parol bilan yopib va ularga faqat tanlangan foydalanuvchilargagina kirishga imkon berish mumkin.

Uchinchi: Bizning saytni boshqarish tizimimiz (CMS, Bitrix) sizning saytingiz asosida yotuvchi bo'lgan, boshqa proektarda ko'p marotaba o'rnatilgani (3500 martadan ko'prok), uchun u saytingizni uni ekspluatatsiya qilish jarayonida dasturiy xatolardan asrashiga kafolat beradi.

CMS - bu Content Management System dan olingan kengaytma, u so'zma-so'z tarjima qilinganda "Sayt kontentini Boshqarish tizimi" yoki "Saytni Boshqarish Tizimi" larni anglatadi. Ba'zida CMS ni saytning "Xarakatlantiruvchisi" deb atashadi. CMS - bu dinamik ma'lumotli web-saxifalarni ishlab chiqarish va ta'minlashga imkon beruvchi dasturiy ta'minotdir. Har xil CMS har xil murakkablikdagi saytlarni proektlashga imkon beradi. Internet magazinlardan to informasion portallargacha, hammasidan ham, CMS axborotli va kontentli saytlarni yaratish uchun mos keladi.

CMS ning ikkita asosiy turi bor:

1. Internetda ishlovchi va joylashuvchi CMS.
2. KompyutYerga o'rnatilgan va sayt bilan avtomatik yoki fayllarni yangilash interfeysi orqali ulangan CMS. CMS ning sayt bilan integratsiyasidan keyin, kompaniyangizning xizmatchisi sayt tarkibi va

strukturasini boshqarishlari, yangiliklarni chop qilishlari, matnlarni taxrir qilishlari va grafikani joylashtirishlari, katalogdagi maxsulotni uzgartirishlari va maxsulotlarni qo'shishlari yangi saxifalarni, bo'limlarni yaratishlari va ko'pgina boshqa ishlarni bajarishlari mumkin. CMS asosida qurilgan dinamik saytlarning ustunligi dizaynning axborotli to'ldirishdan ajratilganida bu xujjat aylanmasini, biznes jarayonlarni shaxsiylashtirish mexanizmlarini avtomatlashtirishga imkon beradi. Bundan tashqari, CMS sayt tarkibini oson, sodda va tez boshqarishga imkon beradi. Bu CMS dan foydalanishning iqtisodiy foydasi sezilarli darajada bo'lishini ta'minlaydi.

CMS ning maqsad va vazifalari - mijoz tomonidan ma'lumot va grafikalarini boshqarish, to'ldirish va taxrir qilish. Soddaroq qilib aytganda, bu tizim - sizga bo'limlarni o'zgartirish va to'ldirish, sayt tarkibini taxrir qilish, sayt grafika va dizaynini o'zgartirishga imkon beradi. CMS siz bunday saytni boshqarish masalalari qimmat turuvchi dasturchilar aralashishmasa hal qilinmaydi.

CMS sayt kontentni boshqarish tizimi web - saytlarni yaratish va ta'minlash narxini pasaytiradi, Shuningdek materiallarni yangilash tezligini oshiradi, ya'ni sayt xo'jaynida material paydo bo'lganidan to saytda chop etilganigacha ketgan vaqt CMS ni sharofati bilan minimal bo'ladi. Buning ustiga CMS da ma'lumotlarni chop etishga tayyorlash jarayoni juda ham oddiy. CMS kontenti boshqarish tizimining shubxasiz ijobiy tomoni shundaki, saytni boshqarish va taminlash narxi pasayadi. Bu web master oylik, maoshiga xujjatlarni izlashga ikkilanish va xatoliklarni oldini olishga ketgan yo'qotishlarning pasayishi, sheriklar va mijozlar bilan bog'lanish tezligining ortishi xisobiga yuz beradi. Sizga loyixani ta'minlash va rivojlantirish uchun mutaxassis yollashingizga to'g'ri kelmaydi.

CMS yordamida siz saytga joylashtirilgan ma'lumotlarning xajmi sayt narxiga tasir qilmasligiga guvox bo'lasiz. CMS kontentni boshqarish tizimi sizga saytda ixtiyoriy xajmdagi ma'lumotlarni joylashtirishga imkon beradi. Odatda CMS ga asoslangan sayt saxifalari butunligicha saqlanmaydi, balki ularga murojat qilishda juda tez shakllanadi, bu saxifalarni yuklashdagi tezlikni oshirishga imkon beradi, Chunki saxifa shabloni (dizaynerlik rasmiylashtirish) faqat bir marta yuklanadi va keyinchalik har bir yangi saxifa so'rovida foydalanuvchi kompyuteridan ochilaveradi. Saytni boshqarish tizimida odatda keshlashtirish mexanizmi amalga oshiriladi, bu qurilmalarga tushadigan yuklamalarni kamaytirishga imkon beradi va ishlashning va saytning kiritish kobiliyatining tezligi oshishiga imkon beradi. Bu ishlar ma'lumotlar bazasi va statik

materiallarga so'rov natijalarini saqlash evaziga sodir bo'ladi. Odatda CMS tizimini saytning kompaniya ishlab chiqaruvchisi o'rnatishini taklif qilishadi. Bu ancha to'g'ri variant Chunki tizim avval boshidanok sizning saytingizga joylashib, moslashib ketadi. Shuningdek, CMS tizimlari qo'shimcha funksiyalarni taklif etadi va oxir oqibat saytda forum, mexmon kitobi, savollar va javoblar va xokazolarni joylashtirishga imkon beradi. CMS ning tekinlari ham mavjud, ular sayt tarkibini boshqarish uchun yetarlicha katta imkoniyatlarni namoyish qiladi. Tekin CMS lar ko'p foydalanuvchilarga ega, ular tekin kontentni boshqarish tizimi yordamida saytni boshqarish muammolarini hal qilishga yordam beradi. Tekin CMS larga katta miqdordagi kengaytma va qo'shimchalar mavjud, ular saytni boshqarish jarayonini ancha qulay va funksional qilishga imkon beradi. Ammo tekin CMS larning salbiy tomoni rasmiy texnik qo'llab kuvvatlashning yo'qligi va sayt tarkibini boshqarish tizimida kritik xatolarni to'g'rilash, tezligining pastligidadir.

Qaysi CMS dan foydalangan ma'qul?

Hozirgi davrga kelib, CMS asosiga qurilgan bir qancha tizimlar yaratilgan. Ularga misol qilib, Drupal, PHP-Nuke, WebDirector, Joomla!, NetCat, Slaed, Microsoft CMS, WordPress, PHPShop kabilarni keltirish mumkin. Bu tizimlarning ba'zilar umuman tekin bo'lsa, ba'zilaridan esa xaqqini to'labgina foydalanish mumkin. CMS ni tanlash bir tomondan oson va jo'ndek ko'rinsada, bu juda jiddiy masala. Internetda ularning turli yo'nalishdagi o'nlab (agar yuzlab bo'lmasa): yuz dollarlab turadiganidan to ochiq kodli bepul turigacha, ma'lum bir soxagagina taalluklisidan ixtiyoriy murakkablikdagi sayt kurish imkonini beradiganigacha, oson boshqariladiganidan administrator uchun katta qiyinchiliklar keltiradiganigacha turini topishingiz mumkin. Xullas, agar CMS lar bilan ishlash bo'yicha hech qanday tajribangiz bo'lmasa, keraklisini tanlash qiyinlashib ketishi tayin. Dastlab keling, pullik CMS larni ishlatmaslikka Xarakat qilamiz. Axir katta pul to'lab, CMS ni ishlatishdan besamara foydalanishga naxojat. Shu yerda bir narsani aytib o'tish kerak: bu bilan o'zimizga qiyin qilamiz. Boshqa tarafdin esa, qanchalik murakkab bo'lsa, shunchalik qiziq bo'lishi aniq, Shunday emasmi? Endi CMS larga yana bir talab qo'yamiz: universallik. Bu degani, biz ishlatmoqchi bo'lgan CMS ixtiyoriy konfiguratsiyadagi kompyuterlarda ham ishlay olishidir. Shuningdek, saytni lokal serverdan xaqiqiy serverga ko'chirganimizda yoki bevosita serverning o'zida yaratayotganimizda CMS bilan hech qanday qiyinchiliklar tug'ilmasligi kerak. Bu talabga esa, afsuski, ko'pchilik bepul CMS lar javob bera olmaydi.

Barcha dasturiy ta'minotlarga qo'yiladigan talablardan yana biri - xavfsizlik (axir siz xaftalab-oylab yaratgan saytingizni shunchaki qiziqish uchun bir kunda buzishlarini xoxlamasangiz kerak?). Shuning uchun CMS larning xavfsizlik darajalari doimiy yangilanib turishi shart. Natijada, talablarimizga deyarli to'liq javob beradigan quyidagi: PHP Nuke, Drupal, Joomla!, Slead CMS tizimlarigina qoladi. Endi ana shu CMS larni batafsilrok ko'rib chiqamiz va o'zimizga kerakli ba'zi xulosalar chiqaramiz.

PHP Nuke. Eng birinchi CMS lardan biri bo'lib, afsuski, ancha vaqtlardan buyon yangilanmay kelmoqda. Biroq, uning kodi ko'plab adashlari uchun asos bo'lib xizmat kilmoqda (masalan, DotNetNuke, PostNuke). Bu bir-biriga nom jixatdan yaqin bo'lgan tizimlar foydalanuvchi diqqatiga navbatma-navbat chiqib turadi. Funkzionallik jixatidan tizim ancha mukammal bo'lib, uning uchun ko'plab qo'shimcha modullar yaratilgan. Masalan, xabarlar lentasi, forum, maqolalar, so'rovlar tashkil qilish, statistika olib borish va boshqalar. Ammo, saytlarning tuzilishi jixatidan bir xillik kuzatiladi. SHu bilan birga, bu tizim yuklanishi qiyin bo'ladi, ya'ni qo'shimcha modullar saxifa yuklanishini sekinlashtiradi. Bu esa ko'plab foydalanuvchilarga yokmasligi tayin. Yuklanish nafaqat saxifaga, Shuningdek, serverga ham tushadi, bu esa xosting xizmatini taqdim etuvchilarga yoqmaydi. Agar sizni Shunday vaziyat qoniqtirsa, aynan shu CMS ni tanlashingiz mumkin.

Drupal. Turli xil o'zgartirishlar orqali o'zimizga moslab olish bo'yicha eng mukammal CMS deb aytish mumkin. SHu yerda Shuni aytish kerakki, "Tomchi" (bu tizim nomi ingliz tilidan aynan Shunday tarjima qilinadi) aslida, CMS emas, balki "CMS ni tuzuvchi tizim". Uning yordamida siz tayyor bloklardan kerakli CMS ni yig'asiz. Shuni aytish mumkinki, funksioallik va dizayn jixatidan o'zingizni kanoatlantiruvchi CMS ni xosil kilasiz. Standart modullar to'plamiga blog, forum, xabarlar lentasi, fayllarni yuklash, ovoz berish, qidiruv kabilar kiradi. Bunda ko'rinadigan kamchilik bitta: PHP va MySQL ni mukammal bilish lozim. Agar bu qisqartmalarni tushunmagan bo'lsangiz, keyingi CMS ga o'tavering, Aks holda esa bu tizimga jiddiyrok e'tibor bering.

Slead CMS. Bugungi kunda ancha ommabop CMS bo'lib, turli xil saytlarda ishlatiladi. Ikki xil varianti mavjud: pullik va tekin (OpenSlead). Ikkovi ham bitta yadroda qurilgan (bu degani xavfsizlik darajasi bir xil) va faqatgina funksionalligiga tasir qiluvchi modullar soni bilan farqlanadi. Bepul variantida biror mavzudagi nisbatan kichik portalni yaratishingiz mumkin. Ammo, bundan ortig'ini emas. Dizayni ham unchalik katta

qiyinchiliklarsiz o'zgartirilishi mumkin.

Tizim aslida Rossiyada yaratilgani uchun rus tilidagi yordam (slead.net) haqida qayg'urmasa ham bo'ladi.

Joomla! Bugungi kunda eng mukammal dvijoklardan biri (agar bepullari ichida eng mukammali bo'lmasa) bo'lib, uning uchun shunchalik ko'p qo'shimcha modullar yaratilganki, ular yordamida o'zingizning saytingizga deyarli ixtiyoriy funksionallikni berishingiz mumkin. Bu dizaynga ham tegishli bo'lib, uni o'zgartirish uchun ham juda ko'p shablonlar mavjud. Biroq, bu yerda bir kamchilik ham bor: shablonlar tizimi Shunday tuzilganki, saxifalarni yaratish va to'ldirish jarayonida sayt bo'limchalari kichik-kichik qismlar bo'lib, ajralib qolishi mumkin. Ya'ni, bu qismlar jadvallar asosida yaratilgan.

21.2. CMS larning afzalliklari va qiyinchiliklari

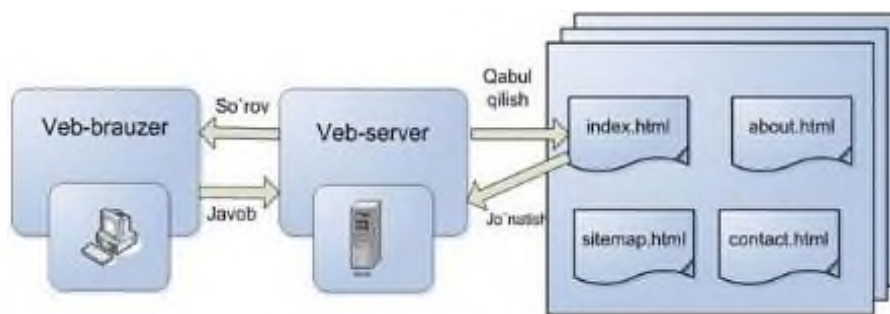
Statik va dinamik Web-sahifalar. Endi savol tug'ilishi mumkin: oddiy HTML kodida yozilgan saxifa bilan CMS nimasi bilan farq qiladi? Avvalombor, HTML da yozilgan sayt statik holda bo'lsa, zamonaviy saytlar dinamik ko'rinishga ega.

Oddiy sayt qanday prinsip asosida ishlashini ko'rib chiqamiz. Foydalanuvchi biror-bir, masalan, page.html saxifasiga murojaat qildi deylik. Web-server o'zidagi Web-saxifalar katalogini ko'rib chiqadi va agar page.html bo'lsa foydalanuvchiga uni ko'rsatadi, Aks holda esa - 404-chi xabarni ("Fayl topilmadi") yetkazadi.

Dinamik sayt esa biroz boshqacha ishlaydi. Foydalanuvchi biror saxifaga murojaat qiladi, biroq bu murojaat diskdagi aniq bir faylga emas, balki sayt dvijokiga (agar dvijok PHP tilida yozilgan bo'lsa, odatda bu fayl index.php bo'ladi) karatilgan bo'ladi. Dvijok foydalanuvchiga kerakli ma'lumotni biror manbadan (masalan, ma'lumotlar bazasi yoki diskdagi fayldan) oladi, uni HTML- saxifaga aylantiradi va web-serverga uzatadi, u esa o'z navbatida, foydalanuvchi brauzeriga jo'natadi.

Unda yana bir savol tug'iladi: agar bitta faylnigina so'rab, uni brauzerga uzatish mumkin bo'lsa, buncha murakkablashtirishning o'zi nimaga kerak? Shuning uchunki, bu saytni boshqarish qulay bo'lsin. Aytaylik, siz 100 saxifalik (bu unchalik ko'p emas) oddiy sayt yaratdingiz. Ertami-kech siz bu sayt dizaynini, ko'rinishini o'zgartirmoqchi bo'lasiz. U holda esa siz shu 100 saxifani qo'lda o'zgartirib chiqishingizga to'g'ri keladi. Ustiga-ustak, siz avvaliga bu 100 saxifani kompyuteringizdagi lokal serverga yuklab olishingiz, undan so'ng taxrirlab, kerakli

o'zgartirishlarni kiritishingiz va oxirida yana serverga yuklashingiz kerak bo'ladi.

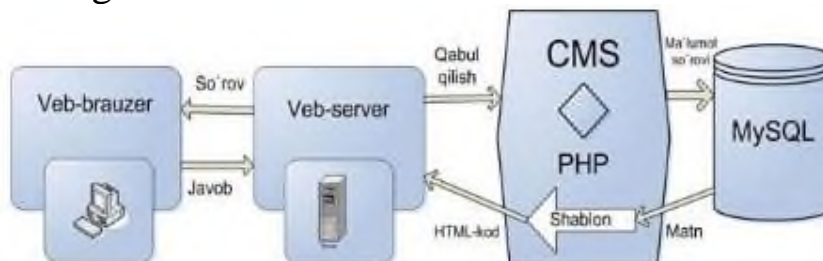


Veb-serverning CMS siz holda qanday ishlashi

21.1-rasm. Web-serverning CMS siz holda qanday ishlashi.

Ana endi dinamik sayt qanday ishlashini ko'rib chiqaylik. Asosiy voqea - "HTML-saxifani yaratish". Ya'ni dvijok kontentni - matnli ma'lumotni (u rasm va jadvallar bilan ham bo'lishi mumkin) oladi va uni dizaynga "kiyintiradi". Demak, saytning dizaynini o'zgartirish uchun siz faqatgina tayyor grafik mavzuni o'rnatishingiz yoki bir-ikkita faylni - sayt shablони fayllaini o'zgartirishingiz kerak.

CMS tizimi saxifalarni dinamik ko'rinishga keltiradi: ular foydalanuvchi saxifani ko'rmoqchi bo'lgandagina amalda yaratiladi. Bu degani, saxifalar foydalanuvchi tomonidan bevosita ko'rilayotganda yangilanishi yoki o'zgartirilishi mumkin.



Veb-serverning CMS bilan qanday ishlashi

21.2-rasm. Web-serverning CMS bilan qanday ishlashi.

CMS afzalliklari. Bu tizimlarning asosiy xislati shundaki, dinamik web-saxifalarni oson yarata olish va ularni turli xil axborot bilan to'ldirish imkoniyatini berishdir. CMS tizimlari odatda murakkab bo'lib, ular o'z ichiga xabarlarini tarkatish xizmatlari (RSS), forumlar va internet-do'konilarni oladi. Shu bilan birga, ularni o'zgartirish ham oson. CMS nafaqat saytdagi turli ma'lumotlarni boshqarishni avtomatlashtiradi, balki shu bilan birga, sayt tuzilishining texnik tomonlari bilan tanish bo'lmaganlar, ya'ni yozuvchi va jurnalistlarga saytda o'z materiallarini

bevosita foydalanuvchi interfeysi orqali qo'shish imkoniyatini beradi. SHu narsa tushunarli bo'ldiki, endi foydalanuvchilarga saytda o'z materiallarini ko'rish uchun HTML ni ham, saytdagi ma'lumotlar yangilanishining texnologiyasini ham bilishi shart emas.

CMS tizimi, shuningdek, Web-sayt yaratishni turli xil mutaxassislar orasida taqsimlab chiqish imkoniyatini beradi. Shunga ko'ra, Web-dizayner o'z diqqatini saytning ko'rinishiga (dizaynni mukammallashtirish va shablonlarni o'rnatishga) qaratsa, texnik yo'nalishda bo'lmagan mutaxassislar uni ma'lumotlar (matn, tasvir va animatsiyalar) bilan to'ldirishga Xarakat qiladi. Bu ham hammasi emas. Ma'lumki, dvijok CMS ning bir qismi xolos. U dizayn shabloni asosida sayt ma'lumotlarini chiqarib beradi. Bundan tashqari, CMS da saytdagi ma'lumotlarni boshqarish (ular sayt bo'limlarini va kategoriyalarini yaratishga va ma'lumotlarni ular orasida bo'lib chiqishga imkon beradi) va bu ma'lumotlarni taxrirlash vositalari ham mavjud. SHu bilan birga, foydalanuvchi sayt ma'lumotlari aynan kaerda (ma'lumotlar bazasida, diskda HTML-fayl ko'rinishida yoki umuman tashki serverda) saqlanayotganini bilishi shart emas.

CMS qo'llanilishi bilan ko'pgina qiyinchiliklar bartaraf etilishi mumkin. Yuqorida misol sifatida ko'rilgan sayt haritasi ham avtomatik ravishda tuziladi, buning uchun foydalanuvchidan hech qanday mexnat talab etilmaydi. SHu bilan birga, CMS saytdagi qo'shimcha funksiyalarni, masalan, suratlar galereyasi yoki ro'yxatdan o'tish oynasini qo'llash imkonini beradi - shu kabi funksiyalar CMS kengaytmalari orqali amalga oshiriladi. Faqatgina u yoki bu kengaytmani o'rnatish kerak xolos.

CMS ga o'tishning o'ziga xos qiyinchiliklari. CMS larning shuncha afzalliklari bo'lganda nima uchun barcha sayt egalari Joomla! yoki Drupal kabi CMS larga otishga intilmayapti? Bunga bir qancha sabablarni ko'rsatish mumkin. Birinchi sabab, inertsiya bo'yicha ketish yoki boshqacha aytganda, o'zi bilgan narsaga o'rganib qolish. Inson HTML ni ishlatishga o'rganib qoladi va CMS ga o'tish saytni boshqarishni avtomatlashtirish bo'yicha qancha qulayliklar keltirmasin, biror narsani o'zgartirishni xoxlamaydi. HTML dan CMS ga o'tish-yillar ichida amalga oshishi mumkin.

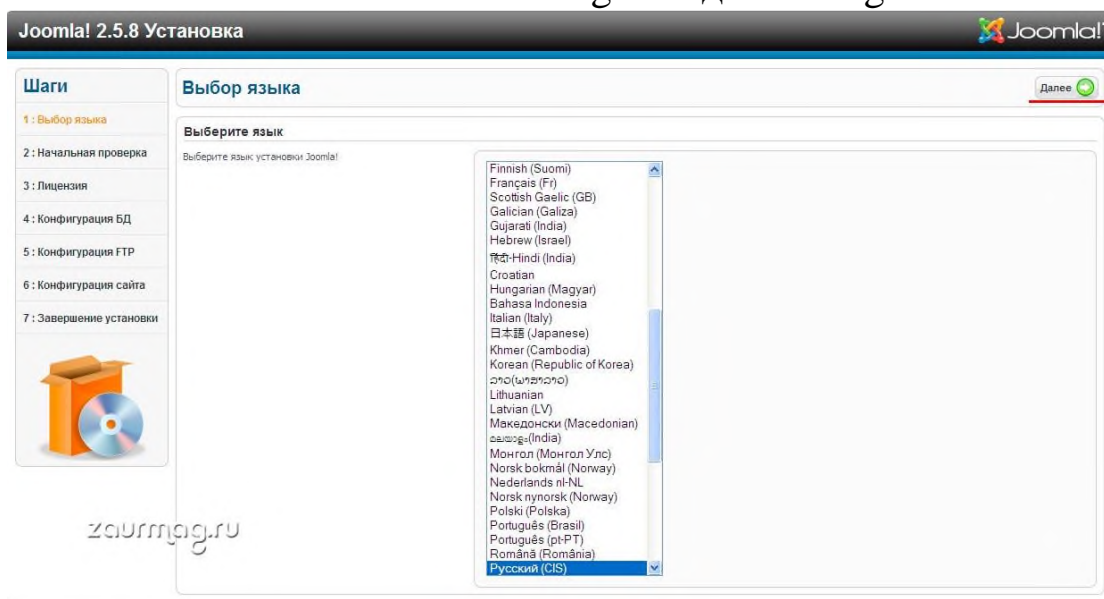
Ikkinchi sabab, mavjud minglab HTML-saxifalarni CMS ma'lumotlar bazasiga moslab o'zgartirish kerak bo'ladi. Aksariyat xollarda buni ko'lda bajarish kerak bo'ladi. Katta sayt uchun bu jarayon ancha ko'p vaqtni oladi, bu esa saytni vaqtinchalik faoliyatining to'xtashiga va balki, kompaniya tomonidan ko'riladigan moliyaviy zararga olib keladi. Albatta,

CMS ga o'tish saytni boshqarish bo'yicha ancha vaqt tejalishini xisobga olsa, kelajakda bu o'zini oklaydi.

Endi, CMS ga o'tishga halaqit beruvchi faqatgina bitta to'siq qoladi - texnik xolat. Deyarli barcha zamonaviy CMS ishlashi uchun xostingda PHP interpretatori va MySQL ma'lumotlar bazasi serveri bo'lishi kerak. Besh olti yil avval shu kabi funksiyalarni taqdim etuvchi xosting-provayderlar kam edi. Hozirga kelib, vaziyat yaxshi tarafga o'zgardi va PHP hamda MySQL ni qo'llaydigan xostingni O'zbekistonning o'zida topish qiyinchilik tug'dirmaydi.

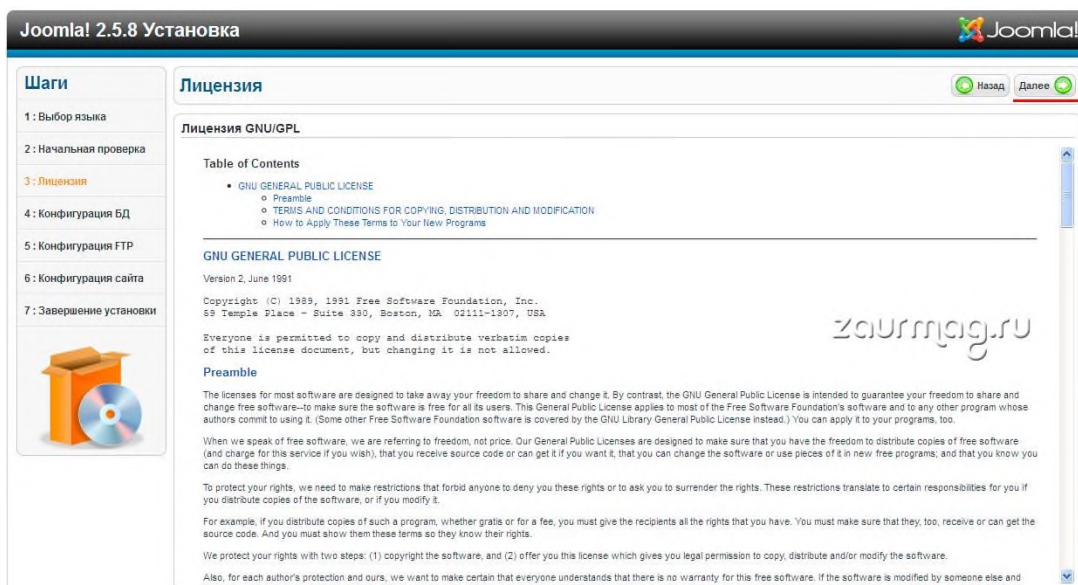
21.3. CMS texnologiyasi - Joomlada sayt yaratish

CMS texnologiyalaridan biri bo'lgan Joomlada sayt yaratish uchun dastlab virtual server Denwer ni o'rnatamiz. Undan so'ng xosil bo'lgan W diskka kirib ichidagi home papkasiga sayt nomi ostida papka ochamiz, uning ichiga WWW papka ochamiz va uning ichiga Joomla saytni tashlaymiz va brauvzerdan saytimizni chaqiramiz. Bu oynada o'rnatish tilini tanlab oldinga tugmasi bosiladi. Keyingi xosil bo'lgan oynamizda esa tekshirish ishlari boshlanadi. Tanishib bo'lgach "далее" tugmasi bosiladi.



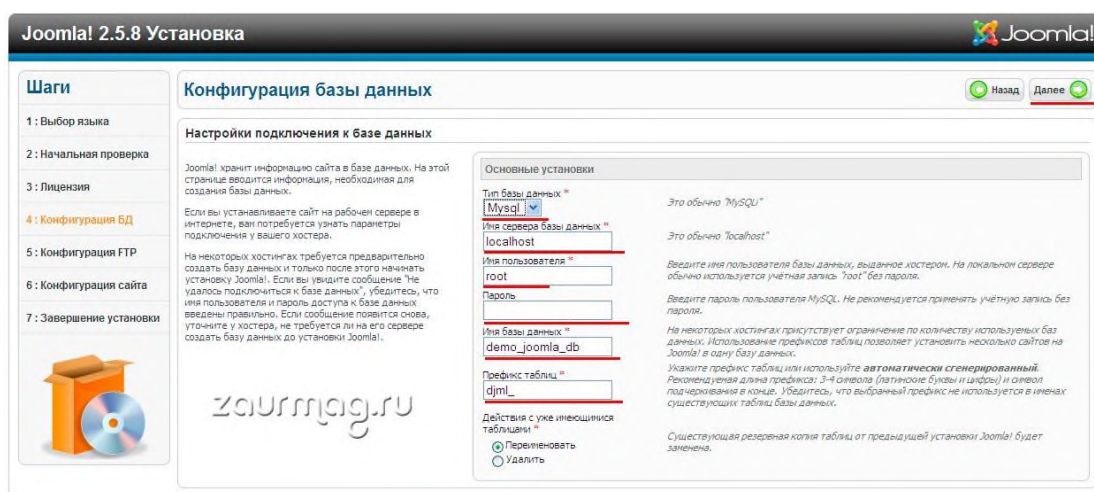
21.3-rasm. Joomlani o'rnatishda dastlabki qadam.

Ushbu oynada esa litenziya haqida ma'lumot beriladi. So'ngra "далее" tugmasi bosiladi.



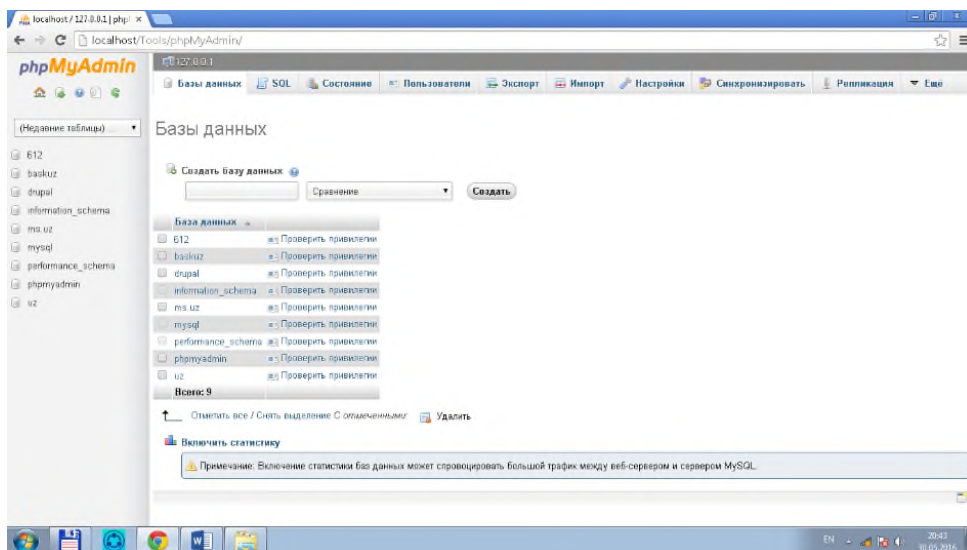
21.4-rasm. Lizenziya haqida ma'lumot beriladi.

Quyidagi oynada esa ma'lumotlar bazasining tuzilishi kiritiladi. Bizni holatimizda u MYSQL bazasi bo'lib xostning va foydalanuvchining nomi berilgan. Oxirgi qismda esa baza nomi kiritiladi. Shuning uchun yangi baza yaratamiz:

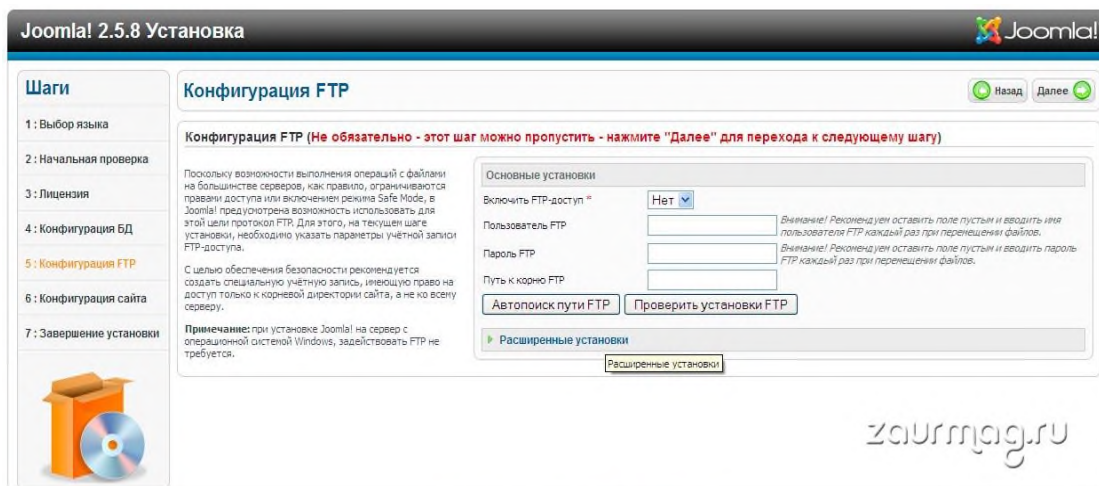


21.5-rasm. MYSQL bazasi bo'lib xostning va foydalanuvchining nomi beriladi.

Navbatdagi oynada MySQL bazasining oynasi berilgan bo'lib yangi baza nomini yozamiz va yaratish tugmasi bosiladi. So'ngra yuqoridagi oynada oldinga tugmasi bosiladi.



21.6-rasm. Yangi baza yaratiladi Ushbu oyna esa o'zgarishsiz qoldirilib “далее” tugmasi bosiladi.




21.7-rasm. Konfiguratsiya FTP ko'riladi.

Quyidagi oynada esa saytning nomi, pochta manzili va parol takroran kiritiladi. So'ngra Demo variantini o'rnatish uchun “далее” tugmasi bosiladi.

Шаги

- 1: Выбор языка
- 2: Начальная проверка
- 3: Лицензия
- 4: Конфигурация БД
- 5: Конфигурация FTP
- 6: Конфигурация сайта
- 7: Завершение установки



Конфигурация сайта

Назад Далее

Название сайта

Введите название вашего сайта.

Дополнительно вы также можете указать мета-теги (Meta Description и Meta Keywords) для вашего сайта. Кроме того, вы можете выбрать, будет ли сайт доступен посетителям сразу после завершения установки или нет.

Основные установки

Название сайта *

[Расширенные настройки - Дополнительно](#)

E-mail и пароль администратора.

Введите e-mail адрес. Это будет e-mail адрес Суперадминистратора сайта.

Введите новый пароль и подтверждение пароля в соответствующие поля ниже. Вы можете изменить логин **admin**. Введенные вами данные будут логином и паролем, которые вы можете использовать для авторизации в Панели управления после завершения установки. Если вы осуществляете миграцию с предыдущей версии, вы можете пропустить этот шаг. В этом случае будут использованы ваши текущие настройки авторизации.

Загрузка демо-данных

ВАЖНО! Начиная с пользователя настоятельно рекомендуется установить демо-данные! Чтобы это сделать, нажмите на кнопку справа прежде, чем переходить на следующий шаг установки.

1. Демо-данные: Перед тем, как завершить процесс установки Joomla!, вы можете заполнить базу данных сайта демо-данными. Это можно сделать двумя путями:

1а. Стандартные демо-данные. Убедитесь, что в выпадающем меню выбраны "Стандартные English (GB) демо-данные", а затем нажмите на кнопку "Установка демо-данных".

1б. Пользовательские демо-данные. Для этого выберите в выпадающем меню демо-данные из доступных вариантов, а затем нажмите на кнопку "Установка демо-данных".

Ваш E-mail *

Логин администратора *

Пароль администратора *

Подтверждение пароля *

Демо-данные

Блог English (GB) демо-данные
 Визитка English (GB) демо-данные
 Стандартные English (GB) демо-данные

[Установка демо-данных](#)


Начиная с пользователя рекомендуется установить демо-данные. Они помогут легче освоиться с основными функциями системы.

21.8-rasm. Pochta manzili, login, parol kiritiladi.

So'ngi oyna quyidagicha bo'lib unda installation papkasini o'chirib tashlash kerakligi ko'rsatilgan. Aytilgan papkani o'chirib bo'lgach сайт tugmasi bosladi va jarayon yakullanadi.

Шаги

- 1: Выбор языка
- 2: Начальная проверка
- 3: Лицензия
- 4: Конфигурация БД
- 5: Конфигурация FTP
- 6: Конфигурация сайта
- 7: Завершение установки



Завершение установки

Сайт Панель управления

Поздравляем, вы установили Joomla!

Нажмите кнопку «Сайт», чтобы перейти к сайту на Joomla! или «Панель управления» для перехода к административной панели.

На сайте Сообщества Joomla! вы найдёте список со ссылками на доступные языковые пакеты Joomla! и инструкции по установке языковых пакетов.

Нажмите на кнопку справа, чтобы открыть сайт Сообщества Joomla! в новом окне.

ВНИМАНИЕ: НЕ ЗАБУДЬТЕ ПОЛНОСТЬЮ УДАЛИТЬ ДИРЕКТОРИЮ INSTALLATION.

Установка Joomla! не будет завершена, пока Вы не удалите данную директорию. Это требование безопасности Joomla!.

[Удалить директорию 'installation'](#)

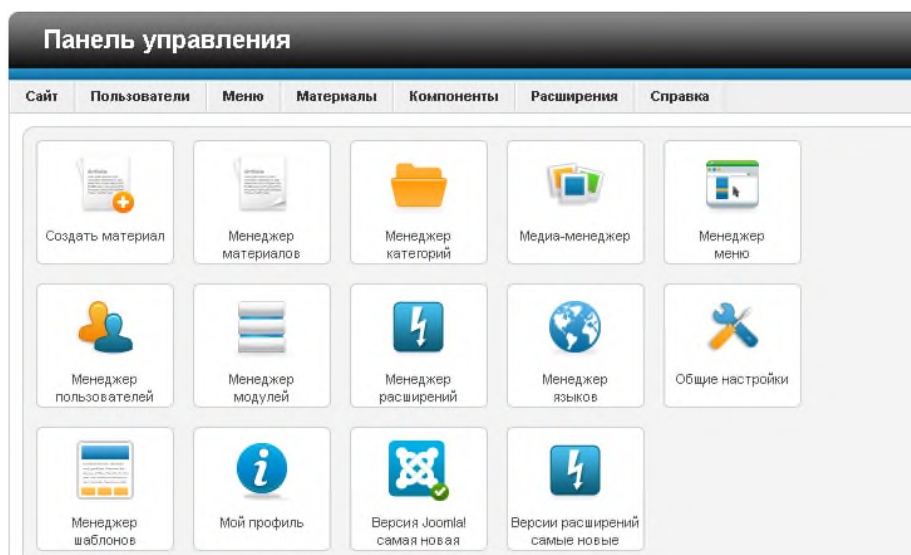
Подробности учетной записи администратора:

Имя пользователя : **admin**

Желаете переключить интерфейс Joomla! на ваш родной язык? Посетите сайт Сообщества Joomla! для загрузки дополнительных языковых пакетов.

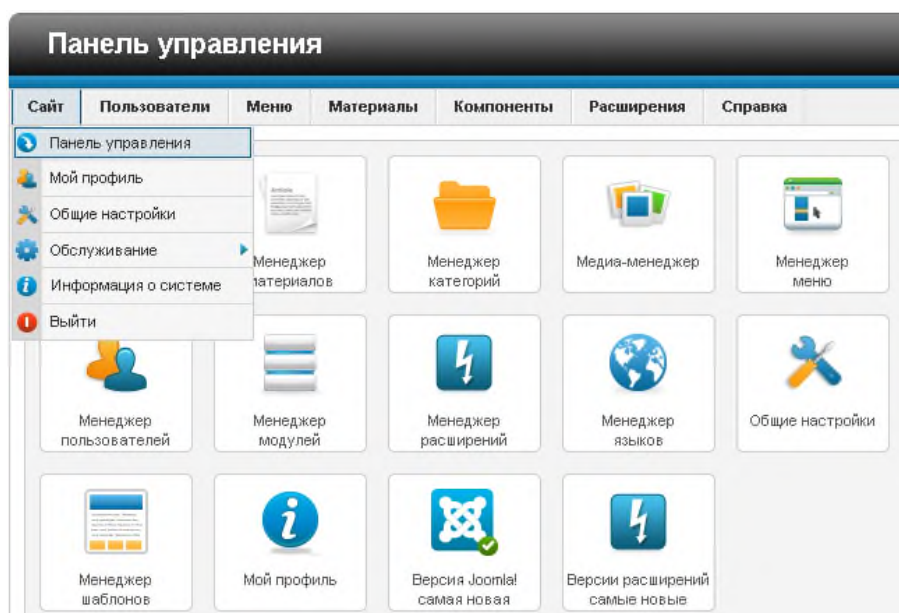
21.9-rasm. Yakunlanuvchi oyna.

Saytning administrator oynasi quyidagicha bo'lib u 7 bo'limdan iborat.



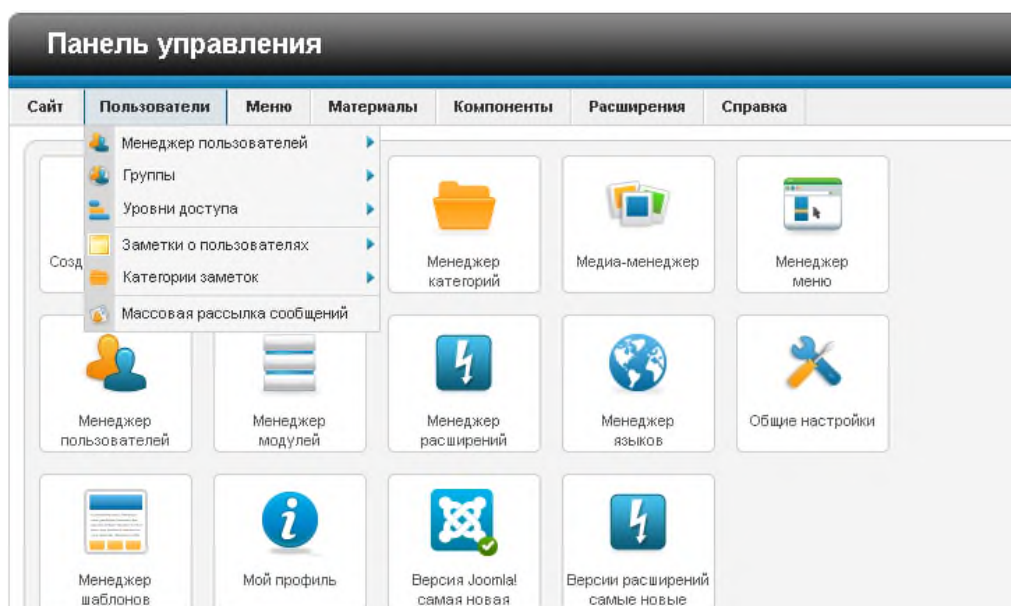
21.10-рasm. Boshqaruv paneli oynasi.

“Сайт” bo’limida boshqaruv paneli, foydalanuvchilarni boshqarish, global konfiguratsiya va paneldan chiqish bo’linmalari joylashgan.



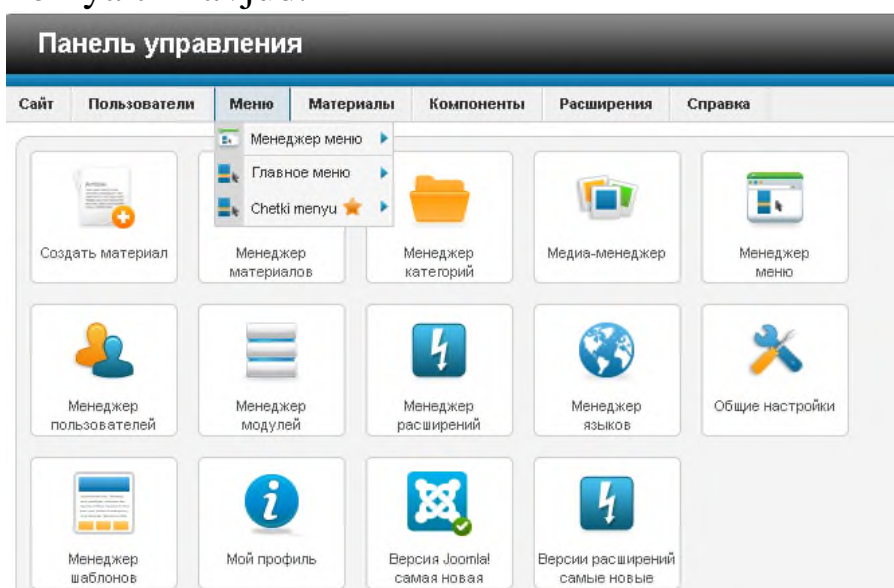
21.11-рasm. Sayt bo’limi imkoniyatlari.

“Пользователи” bu bo’limda foydalanuvchi yaratish, guruhlar yaratish, eslatmalarni yaratish va boshqanarsalarni o’z ichiga oladi.



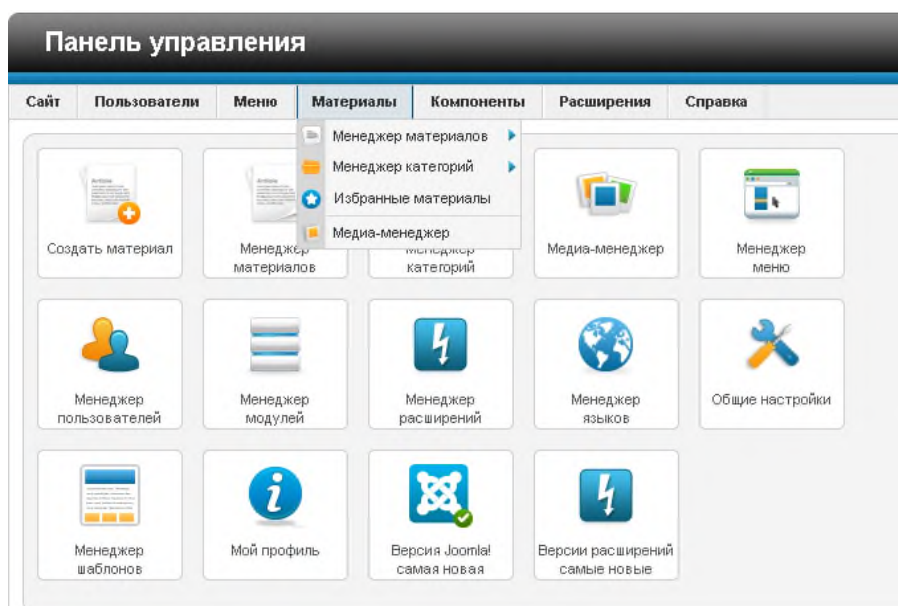
21.12-rasm. “Пользователи” bo’limi imkoniyatlar oynasi.

“Меню” bo’limida saytning barcha qismidagi menyo’larni boshqarish imkoniyalti mavjud:



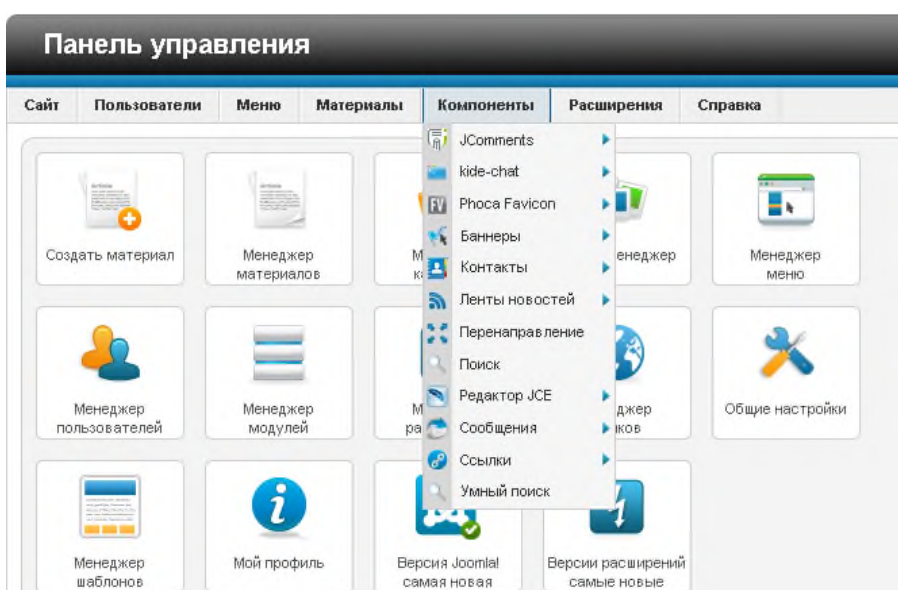
21.13-rasm. “Меню” bo’limi imkoniyatlar oynasi.

“Материалы” - bo’limida biz qilayotgan web saytimizni katigoriyasi va materiallarini kiritib chiqamiz. Biz yana asosiy sahifada ko’rinib turishi uchun “избранны материалы” qilib qo’shgan ma’lumotlarni asosiy qilib qo’yishimiz mumkin.



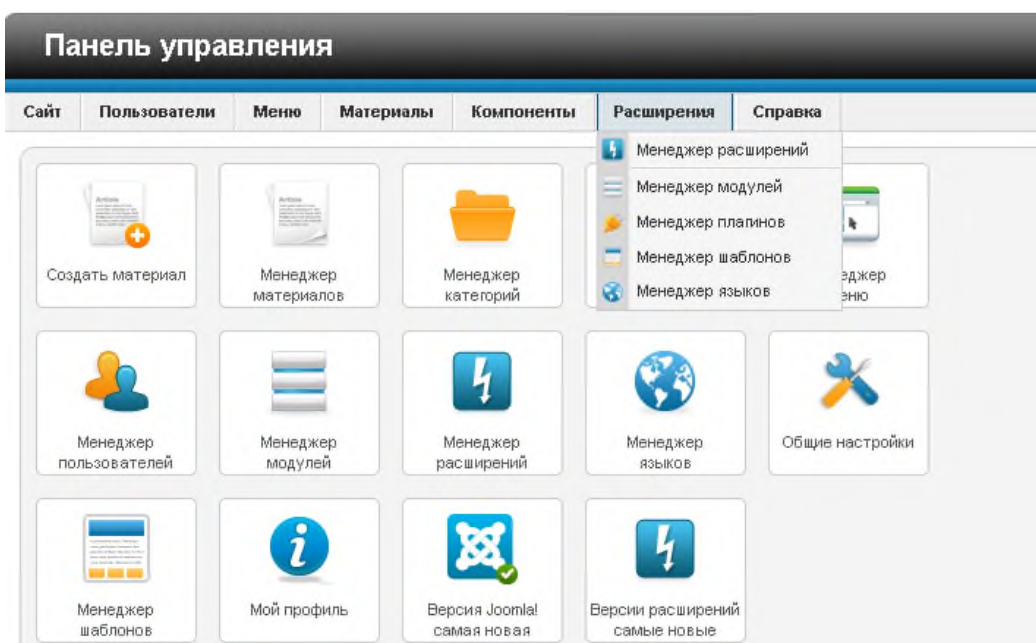
21.14-рasm. “Материалы” bo’limi imkoniyatlar oynasi.

“Компоненты” bo’limida biz qilayotgan saytimizda barcha komponentlarni boshqarish imkoniyatiga ega bo’lamiz.



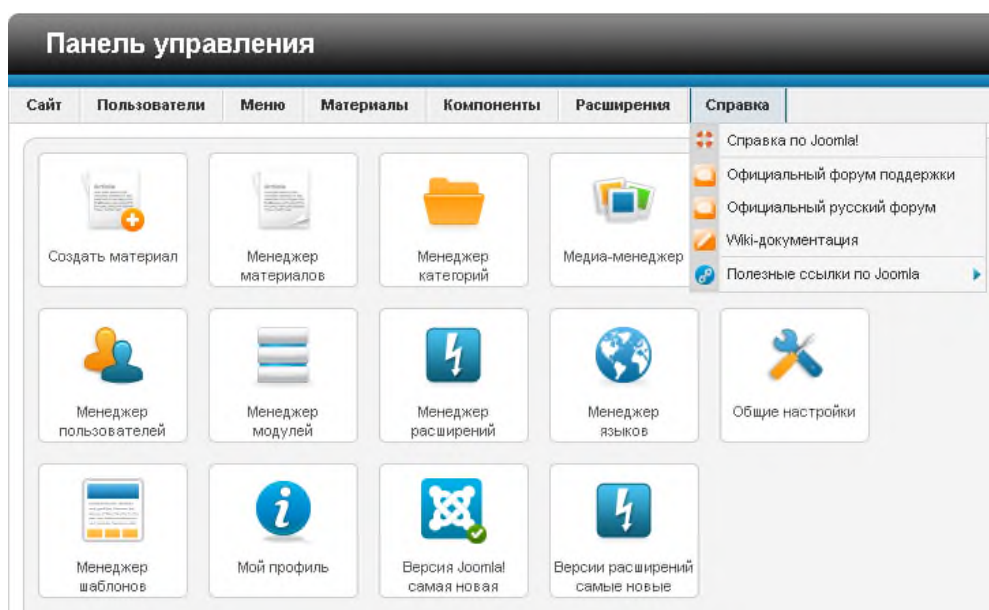
21.15-рasm. “Компоненты” bo’limi imkoniyatlar oynasi.

“Расширения” bu bo’limda biz saytimiz uchun kerak bo’ladigon barcha modul, plaginlar, shablonlarni yuklab olish imkoniyatiga ega bo’lamiz va bu yerda biz yaratayotgan saytimizni tilini ham shu yerda o’zgartirishimiz mumkin bo’ladi.



21.16-rasm. “Расширения” bo’limi imkoniyatlar oynasi.

“Справка” bo’limida joomla haqida so’rovlarga javoblar bo’ladi ma’lumotlarga ega bo’lamiz.

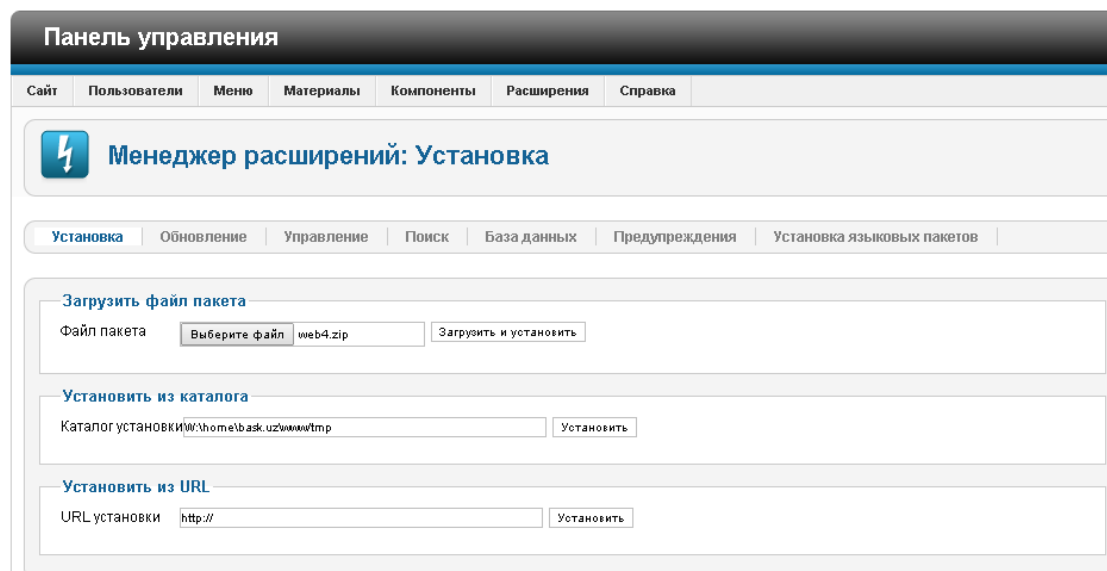


21.17-rasm. “Справка” bo’limi imkoniyatlar oynasi.

Boshqaruv paneli bilan tanishib chiqdik endi biz saytni yaratish uchun CMS Joomlaдан foydalanib ish jarayonini olib boramiz. Joomla ni ishga tushurganimizda u standart holatda boladi. Biz ta’lim sohasiga tegishli sayt qilganligimiz uchun o’zimizga mos keladigon shablon o’rnatamiz. Shablonni iloji boricha ochiq ranglardan foydalanganimiz maqsadga muofiq bo’ladi. Yorqin ranglar esa saytda odamni ko’zini ham charchatib qo’yadi.

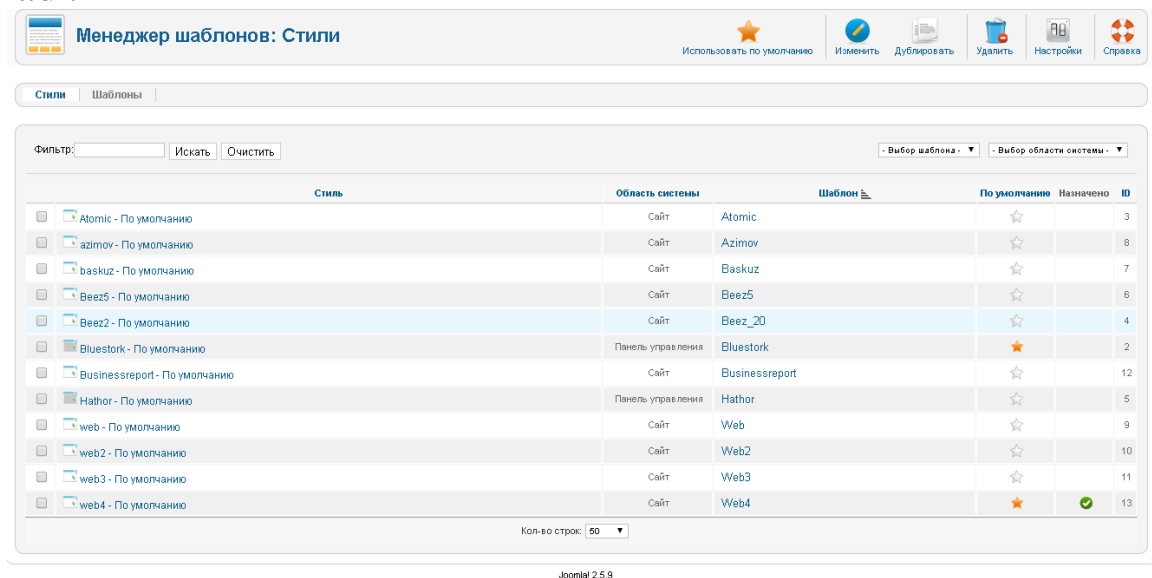
Endi biz shablonni oʻrnatib saytimizni menyularini yaratishni, modullardan foydalanib dizaynini chiroyli koʻrinishga olib kelamiz.

Shablonni oʻrnatish uchun biz avvalo “Панел управления” boʻlimidan “Расширения” boʻlimidan “Менеджер расширения” boʻlimini tanlanadi “Выберите файл” ni tanlab shablon joyi koʻrsatiladi va “Загрузить и установить” tugmasi boʻsiladi va shablonni yuklab olamiz.



21.18-rasm. Shablon oʻrnatish.

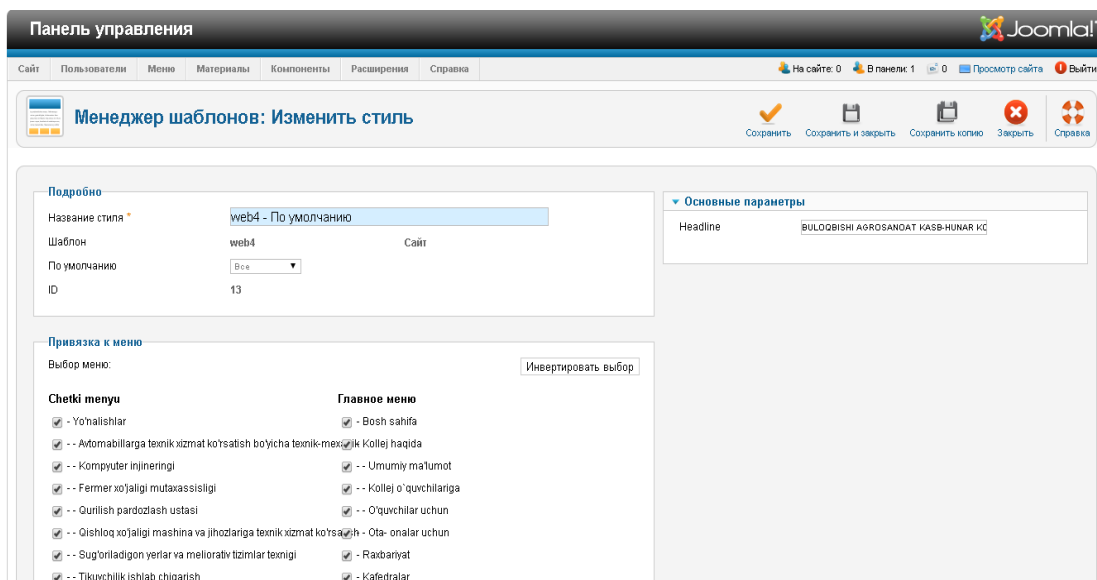
Yuklab olingan shablonni faol holatga otgazish uchun biz “Панел управления” boʻlimidan “Расширения” boʻlimidan “Менеджер шаблонов” boʻlimini tanlanadi va keyin esa yuklab olingan shablonni tanlanadi.



21.19-rasm. Shablonni faol holatga keltirish.

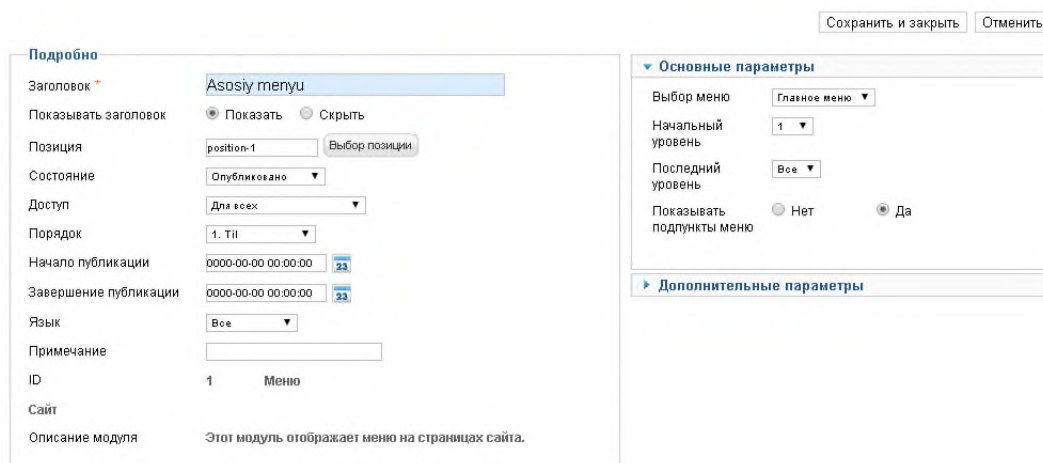
Endi biz oʻrnatilgan shablonimizga yaratgan boʻlimlarimiz va

oʻrnatgan modullarimiz koʻrinishi uchun yuklangan shabolga kursokni olib borib chap tugmasini bir marta bosiladi va quydagi oyna bosiladi “инвертировать выбор” tugmasi bosiladi va “Сохранить и закрыть” tugmasi bosilib chiqib ketiladi.



21.20-rasm. Oʻrnatgan shablonimizda barcha boʻlimlar koʻrinishi uchun.

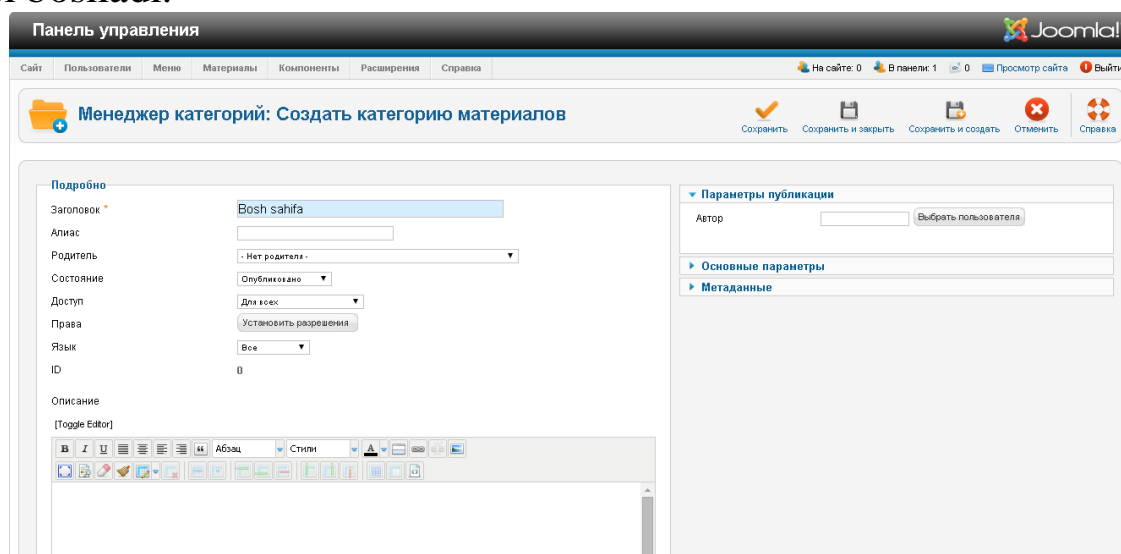
Asosiy menyu boʻlimini yaratamiz va qolgan boʻlimlar shu yaratgan menyumiz ichida joylashadi. Shablonimizni qaysi joyida turish keraligini yaʼni (position) ni koʻrsatib qoʻyamiz. Buning uchun biz “Менеджер меню” boʻlimiga kirib yaratgan menyumizni joylashish oʻrnini koʻrsatib qoʻyamiz.



21.21-rasm. Asosiy menyuni joylashish oʻrni.

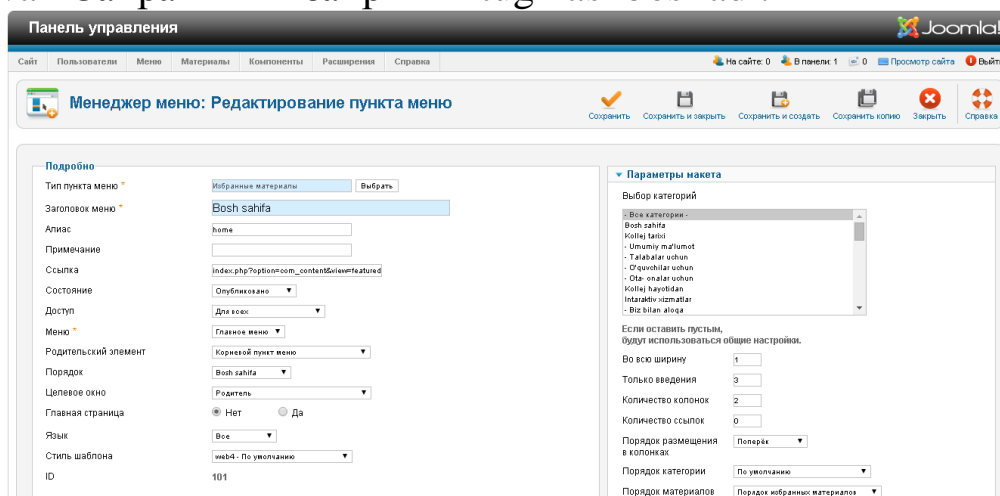
Saytimizga Bosh sahifa qismini yaratishni koʻrib chiqamiz. Buning uchun biz bosh sahifani “категория” sini yaratamiz.

“Материалы” bo’limidagi “Менеджер категорий” bo’limidan “Создать категорий” tanlanib yaratamiz va “Сохранить и закрыть” tugmasi bosiladi.



21.22-rasm Bosh sahifasini katigoriyasini yaratish.

Biz bosh sahifa menyusini yaratish uchun “Меню” bo’limidan “Менеджер меню” dan “Создать меню” bo’limidan bosh sahifa bo’limi yaratamiz va “Сохранить и закрыть” tugmasi bosiladi.



21.23-rasm Bosh sahifa menyusini yaratish.

Bosh menyuga ma’lumot kiritilmaydi unga qo’yilgan materiallar asosiy qilib chiqiladi va bosh sahifamizda ko’rinib turadi.

NAZORAT SAVOLLARI

1. CMS lar turlari va ularning paydo bo’ishi?
2. CMS larning afzalliklari va qiyinchiliklari?
3. CMS texnologiyasi – Joomlani yuklash bosqichlari?

22-BOB. FREYMVORK PLATFORMADA WEB DASTURLASH

22.1. Framework dasturlash asoslari

MVC (Model View Controller, Model Ko'rinish Nazorat). Model bu - ma'lumotlar bazasidagi biror bir jadval (table) bilan ishlaydigan obyekt (class) dir.

View bu - HTML kodlarni yozasidan fayl yoki foydalanuvchiga (browserda) ko'rinadigan kod.

Controller bu - foydalanuvchi so'roviga mos holda kerakli model bilan ishlab, kerakli viewni foydalanuvchi jo'natadi.

Eng birinchi frameworklardan biri - bu CakePHP. Biroq CakePHP o'sha vaqtda hozirgi Frameworklardan anchagina farq qilar edi. Bu Frameworkda OOP (Object Oriented Programming – Ob'yektlarga Asoslangan Dasturlash) dan deyarli foydalanilmagan. Ular MVC konsepsiya asosida ishlamas edi, shuning uchun ham ortiqcha yuklanish talab etuvchi kodlari ko'p bo'lgan. Undan so'ng 2007-yilda yana bir kuchli - Zend Framework 1.0 yaratildi. Zend Framework 1.0 funksionalligini Framework Zend 1.11 ga tenglashtirish mumkin. Keyinchalik bu Frameworklardagi kamchiliklar bartaraf etildi, ortiqcha kodlar olib tashlandi. Hozir esa Ajax yordami bilan sahifalarni ortiqcha yuklanishlar va qotib qolishlarsiz yuklab olish imkoniyati tug'ildi. 2011-yildan Frameworklarning yangi davri boshlandi. Chunki bu paytga kelib, Frameworklar PHP5.3 versiya asosida tuzila boshlandi (PHP 5.3 2009-yil avgust oyida chiqarilgan).

2013-yilga kelib, Frameworklarga talab va ulardan foydalanuvchilar soni yanada oshdi.

PHP dasturlash tiliga talabning ko'payishi esa o'z navbatida, PHP Frameworklarning qulayligi, ishonchliligi, funksionalligi oshishiga sabab bo'ldi.

2014-yil avgust oyining oxirlariga kelib, PHP 6.0 versiya yaratildi. Bu esa web-loyihalarni yaratishdagi xatoliklarni kamaytirish va ish unumdorligini oshirish uchun Frameworklarni yanada takomillashtirish imkonini yaratdi.

Hozirda Web-dasturchilar tomonidan CodeIgniter, Yii, Medoo, Flight, Pop PHP, Laravel, Symfony, Zend, CakePHP, Kajona va boshqa ko'plab qulay va ko'p funksiyali PHP Frameworklardan foydalanilmoqda.

CodeIgniter - eng ko'p qo'llaniladigan MVC tizimli Frameworklardan biri. EllisLab kompaniyasi, Paul Burdick va Rick Ellis

tomonidan yaratilgan. Birinchi marotaba 2006-yil 28-fevralda ommaga havola etilgan. 2011-yil 28-yanvarga kelib, CodeIgniter2.0 yaratildi. Bu Framework ikki bo'limli: CodeIgniter Core va CodeIgniterReactor dan iborat.

2014-yil 6-mayda CodeIgniter2.2.0 versiya yaratildi.

Mazkur versiya boshqa frameworklardan ko'ra ancha tez va yengil hisoblanadi. MySQL, PostgreSQL, MSSQL, SQLite, Oracle kabi ma'lumotlar bazasini quvvatlaydi. O'rganish juda oson. Sozlash va o'rnatish uchun ko'p vaqt talab qilmaydi. Deyarli mikroframework desak ham bo'ladi. Biroq ma'lumotlarni bazadan Modelga avtomatik generatsiya qilish imkoni mavjud emas.

22.2. Symfony, Laravel va Yii freymworklar

Symfony - 2005-yilda ishlab chiqilgan Framework bo'lib, juda kuchli MVC Framework hisoblanadi. Dasturchilar orasida anchagina ommalashgan. 2011-yil lider Frameworklar qatoriga kirgan va hozirgacha mashhurligini yoqotmagan.

Laravel - deyarli barcha professional Web-dasturchilar bu Framework haqida xabardor. Hozirgi kunda eng ommabop Frameworklardan biri bo'lgan Laraver dastlab yaratilgan versiyalari MVC konsepsiya asosida ishlamagan. Ma'lumotlar bazasi bilan ishlash, kesh bilan operatsiyalar, sessiyalar, modul va kutubxonalar, sodda mexanizmlar formalar va HTML kodlar ham ORM (Object Relational Mapping) struktura bo'yicha ish 2011-yil 24-noyabrga kelib, 2-versiya, 2013-yil 28-mayda esa Laravel 4 yaratildi.

Yii Frameworkning asoschisi Qiang Xue bo'lib, u Yii Frameworkni 2008-yil 1-yanvarda yaratdi. Oradan bir yil o'tib, dekabr oyida Yiiframework 1.0 ni ommaga havola qildi. Yii'ni yaratishdan oldin u ko'p ishlatiladigan barcha frameworklarni o'rganib chiqdi. Natijada, qolgan Frameworklarda yo'l qo'yilgan kamchiklarni o'z Frameworkida bartaraf etdi. Hozirda Yii Framework'ga talab kuchli va u o'z o'rnida boshqa Frameworklar bilan raqobatlasha, ulardan bimalol ustunlik qila oladi ham. Yii yuqori samarador, komponentlarga asoslangan PHP framework. Yii Web-dasturlarni yaratishni sezilarli darajada tezlashtiradi.

Yii - Yes It Is (ha bu o'sha) degan so'zning qisqartmasidir. Yii'da yaratilgan Web-dasturlarning ishlashi uchun kamida PHP 5.1.0 versiyasi o'rnatilgan Web server kerak bo'ladi. Bundan tashqari, Yii'da ishlash uchun OOP (Object Oriented Programming - Obyektlarga asoslangan

dasturlash) ni tushunish dasturchi uchun juda muhim, chunki Yii to'liq OOP'ga asoslangan Frameworkdir. Yii umumiy Framework hisoblangani uchun ham u yordamida ixtiyoriy web-dasturlarini yaratish mumkin. Yii yengil, ko'p resurs talab qilmaganligi va murakkab keshlash mexanizmlariga ega bo'lganligi uchun ham katta trafikli dasturlarni, masalan, portallar, forumlar, kontentni boshqarish tizimlari (CMS), elektron kommersiyani yaratishga mos keladi.

Yii ning yordamchi funksiyalari bilan ishlash. Biz odatda kattagina ish qilayotganimizda bir hil funksiyalar yozishimizga tog'ri keladi va bu ishni qayta qayta takrorlanishiga duch kelamiz. Misol uchun biror bir joyda 2 ta DateTime ning o'zaro ayirish masalasi bo'lsin. Bu ishni bir nechta joyda ishlatiladigan bo'lsa unda xar bir shu ishni qiladigan funksiyani yozib ketaveramiz. Uning o'rniga bir yordamchi funksiya sifatida bir joyda saqlasak o'sha funksiyani kerak bo'lganda o'sha nom ostida chaqirishimiz mumkin bo'ladi. Bunda funksiyani har safar yozib o'tirish shart bo'lmaydi. Bunda har bitta dasturlash tilining bunday imkoniyati mavjud. Shu kabi yii da ham bu imkoniyat mavjud. Keling aslida funksiyani qanday yaratish va undan qanday foydalanish haqida gaplashsak. Yordamchi classning ko'rinishi

```
<?php
// protected/components/MyClass.php
class MyClass {
...
}
```

Ushbu holat Classning standart ko'rinishi: Bunda ushbu faylni "protected/components/MyClass.php"

Ko'rinishdagi joyga saqlash kerak bo'ladi. Ya'ni "components" papkasining ichiga.

Yordamchi funksiya ushbu ko'rinishda elon qilinadi

```
<?php
//protected/components/Helpers.php NOT RECOMMENDED!
class Helpers {
// does the first string start with the second?
public static function startsWith($haystack, $needle)
{
return strpos($haystack, $needle) ===0;
}
...
}
```

Bunda birinchi bo'lib class nomi e'lon qilinmoqdi keyin ehtiyojdagi funksiya yozilmoqda. Shu yordamchi funksiyaning chaqirish yo'li quyidagicha

...

```
if (Helpers::startsWith(...))
```

Ya'ni class nomi, funksiya va funksiya beriladigan qiymat.

Endi yordamchi fayl e'lon qilish va uni ishlatish usulini bir kichik misol bilan ko'rib chiqamiz.

Yordamchi fayl nima? Yordamchi fayl bu helpers.php nomi ostida yaratiladigan fayl bo'lib unda bir qancha funksiyalar yozishingiz mumkin bo'ladi.

```
"protected/components/helpers.php"
```

```
<?php
```

```
function time_difference($t1, $t2)
```

```
{
```

```
list($hh1,$mm1,$ss1) = explode(':', $t1);
```

```
list($hh2,$mm2,$ss2) = explode(':', $t2);
```

```
$time2 = mktime((int)$hh2, (int)$mm2, (int)$ss2);
```

```
$time1 = mktime((int)$hh1, (int)$mm1, (int)$ss1);
```

```
$duration = $time2 >= $time1? $time2 - $time1: $time2 + 86400 - $time1;
```

```
$hh3 = floor($duration / 3600);
```

```
$mm3 = floor(($duration - $hh3 * 3600) / 60);
```

```
$ss3 = $duration - 3600 * $hh3 - 60 * $mm3;
```

```
$result = date('H:i:s', mktime($hh3, $mm3, $ss3));
```

```
return $result;
```

```
}
```

Ko'rib turganingizdek bu yerda class nomi yo'q shunchaki funksiya yaratilib ketmoqda. Bu holatda hozir biz ikkita vaqt orasidagi vaqtni topmoqdamiz.

Bu faylni ham odatdagidek "components" papkasiga tashlanadi va nomini "helpers.php" saqlash kerak bo'ladi. Endi bu funksiyaning chaqirilishi quyidagicha chaqiriladi.

Ko'rib turganingizdek bu yerda class nomi yo'q shunchaki funksiya yaratilib ketmoqda. Bu holatda hozir biz ikkita vaqt orasidagi vaqtni topmoqdamiz

Bu faylni ham odatdagidek "components" papkasiga tashlanadi va nomini "helpers.php" saqlash kerak bo'ladi. Endi bu funksiyaning chaqirilishi quyidagicha chaqiriladi.

```
<?php echo time_difference($date1,$date2); ?>
```

Bu ishimiz xozirda to'g'ridan to'g'ri ishlaymaydi. Chunki bu funksiyani qayerdan olishni bilmaydi

Shuning uchun quyidagicha ish qilamiz:

```
<?php
// protected/config/main.php
require_once(dirname(__FILE_). '/../components/helpers.php');
```

...

Config papkamizdagi main.php faylimizga yuqoridagi yo'lni ko'rsatib qo'yishimiz kerak bo'ladi.

Shaxsiy widgetimizni yaratishning kichik ko'rinishi.

UserMenu sinfini yaratish.

Foydalanuvchi menu sini portlet qismi ostida UserMenu sinfini yaratamiz. Bu sinfni biz/protected/components/UserMenu.php da saqlaymiz.

```
<?php
Yii::import('zii.widgets.CPortlet');
class UserMenu extends CPortlet
{
public function init()
{
$this->title=CHtml::encode(Yii::app()->user->name);
parent::init();
}
protected function renderContent()
{
$this->render('userMenu');
}
}
```

UserMenu sinfini CPortlet clasi vorisligi ostida yaratamiz, CPortlet zii kutubxonasiidagi sinf. U init() va renderContent() larni CPortlet sinfida qayta aniqlaydi.

Birinchi navbatda foydalanuvchining nomini o'rnatadi menu tepasidagi titlega, keyin render buyrug'i orqali shu sinfning viewiga murojaat qiladi, ya'ni userMenu.

userMenu Viewini yaratish:

Biz hozir userMenu viewini yaratamiz va quyidagicha joyga joylaymiz.

```

/protected/components/views/userMenu.php
<ul>
<li>
<?php echo CHtml::link('Yangi qayd yaratish',array('post/create'));
?>
</li>
<li>
<?php echo CHtml::link('Qaydlarni
boshqarish',array('post/admin')); ?>
</li>
<li>
<?php
echo CHtml::link(
'To'g'ri deb topilgan sharxlar',
array('comment/index')) . ' (' .
Comment::model()->pendingCommentCount . ')';
?>
</li>
<li>
<?php echo CHtml::link('Chiqish',array('site/logout')); ?>
</li>
</ul>

```

Ushbu view faylni components dagi papkani ichida view papka yaratib o'sha papkani ichida saqlash kerak bo'ladi. UserMenu Sinf shundagina topib oladi.

UserMenu Portletidan foydalanish:

```
/protected/views/layouts/column2.php
```

```

...
<div id=«sidebar»>
<?php
if(!Yii::app()->user->isGuest)
$this->widget('UserMenu');
?>
</div>
...

```

Bu yerda mehmon bo'lsa UserMenu unga ko'rinmaydi. columns2 faylining sidebar qismiga joyladik, bizga endi bu menu o'ng tomonimizda ko'rinadi. Tayor dizayn ko'rinishida ko'rindai. Buning boisi CPortletda bir qancha stillar birlashtirilgan ma'lum taglar uchun, o'sha taglar uchun maxsus

o'zining qolipiga solmoqda. Bunda o'zingiz stillar berishingiz mumkin. Buning uchun

```
/protected/views/layouts/main.php
```

fayli ichiga o'zimizning CSS faylimizni chaqiramiz va taglarimizga sinflar beramiz. Bu sinflar umumiy yii sinflari bilan ustma ust tushmasligi kerak. Endi biz yaratilgan menuga o'zimiz hohlagan stillarni berishimiz mumkin.

Yana bir kichik misol:

```
/protected/components/MyWidget.php
```

```
<?php
```

```
Yii::import('zii.widgets.CPortlet');
```

```
class MyWidget extends CPortlet
```

```
{
```

```
public $testVar;
```

```
public $labels = array();
```

```
public function init()
```

```
{
```

```
parent::init();
```

```
$this->render(myWidget, array(
```

```
'testVar' => $this->testVar,
```

```
'labels' => $this->labels,
```

```
));
```

```
}
```

```
}
```

```
/protected/components/views/myWidget.php
```

```
<?php
```

```
echo '<h1 style=«color:red;»>'.$testVar.'</h1>'; // tagga still
```

berishning qo'pol usuli

```
echo '<div class=«class_name»>'; // agar sinf berishni hohlasangiz
```

```
echo '<ul class=«class_name»>';
```

```
foreach($labels as $label)
```

```
{
```

```
echo '<li>';
```

```
echo $label;
```

```
echo '</li>';
```

```
}
```

```
echo '</ul>';
```

```
echo '</div>';
```

```
?>
```

Widgetni ishlatish ko'rinishi.

```
<?php
$testVar = 'Salom men h1 man';
$label = array('qizil', 'sariq', 'yashil');
$this->widget(MyWidget, array('testVar'=>$testVar,
'labels'=>$label));
?>
```

22.3. Freymwork platformada ma'lumotlar ba'zasi bilan ishlash

Yii DAO PHP Data Objects (PDO) ustiga qurilgan bo'lib, ko'pgina ma'lumotlar omboriga kirish xuquqi uchun foydalaniladi, misol uchun MySQL, PostgreSQL va shu kabilar. Shu sababli Yii DAO dan foydalanishda PDO (dinamic library) dinamik kutubxonasi o'rnatilgan bo'lishi va foydalanilayotgan ma'lumotlar omboridagi PDO driveri bilan mos bo'lishi kerak. (Misol uchun PDO_MYSQL)

Yii DAO 4 calassdan tashkil topgan:

- CdbConnection: ma'lumotlar omboriga bog'lanishni ifodalaydi.
- CdbCommand: ma'lumotlar omborida amalga oshishi kerak bo'lgan so'rovni ifodalaydi.
- CdbDataReader: So'rovga javob qaytaruvchi bir yo'nalishdagi ma'lumotlar satri oqimini ifodalaydi..
- CdbTransaction: Ma'lumotlar ombori tranzaksiyasini ifodalaydi.

Keyinchalik turli xil misollarda Yii DAO dan foydalanishni ko'rib chiqamiz.

Ma'lumotlar ombori bilan bog'lanishni o'rnatish uchun CdbConnection sinfini hosil qilishimiz va faollashtirishimiz kerak bo'ladi. Qo'shimcha ma'lumot : ma'lumotlar omboriga bog'lanish uchun (host, port, foydalanuvchi nomi, attribute, va boshqalarini) va DSN (Data Source Name) ni ko'rsatib qo'yish kerak bo'ladi.

```
$connectionqnew CdbConnection($dsn,$username,$password);
// bog'lanishni o'rnatamiz
// try...catch mumkin bo'lgan qoidalarni to'xtatish uchun
konstruktoridan foydalanamiz
$connection->active=true;
...
$connection->activeqfalse; // bog'lanishni yopish
```

DSN formati foydalanilayotgan PDO driveriga bog'liq. DSN ni qurishda foydalanilayotgan driver bog'lanishi sintaksisiga mos PDO

attribute nomi va ikki nuqtadan keyin bog'lanish parametrlari ko'rsatiladi. Batafsil PDO bu yerdan tanishishingiz mumkin. Quyida bir qancha asosiy DSN formatlari ko'rsatilgan:

- SQLite: `sqlite:/path/to/dbfile`
- MySQL/MariaDB: `mysql:host=localhost;dbname=testdb`
- PostgreSQL: `pgsql:host=localhost;port=5432;dbname=testdb`
- SQL Server: `mssql:host=localhost;dbname=testdb`
- Oracle: `oci:dbname=//localhost:1521/testdb`

Haqiqiy ilova komponentlarida undan foydalanishimiz mumkin, shu boisdan `CdbConnection` `ApplicationComponent` sinf vorisidir. Bu uchun db komponentini quyidagicha holatda ilova konfiguratsiyasini qilish kerak:

```
array(  
    ...  
    'components'=>array(  
        ...  
        'db'=>array(  
            'class'=>'CdbConnection',  
            'connectionString'=>'mysql:host=localhost;dbname=testdb',  
            'username'=>'root',  
            'password'=>'password',  
            'emulatePrepare'=>true, // MySQL ning bir qancha ilovalari uchun  
kerak  
        ),  
    ),  
)
```

Endi biz `Yii::app()->db` orqali ma'lumotlar ombori bilan bog'lanishga kirish xuquqini olishimiz mumkin. Faol bo'lmagan bog'lanishlar uchun `CdbConnection::autoConnect` da false belgisini o'rnatishimiz kerak. Bu usul kodning ixtiyoriy joyidan ma'lumotlar ombori bilan bog'lanishga imkon beradi.

SQL so'rovlardan foydalanish

Qachonki biz ma'lumotlar omboriga ulansak unda biz `CdbCommand` dan foydalanib SQL so'rovini bajarishimiz mumkin. Buning uchun `CdbCommand` sinf nusxasini hosil qilib `CdbConnection::createCommand()` orqali SQL so'rovini ko'rsatishimiz mumkin bo'ladi:

```
$connection=Yii::app()->db; // attribute orqali "db" bilan bog'lanish  
// Ochiqdan ochiq murojaat uchun bu usulni ham qo'llash mumkin  
// $connection=new CdbConnection($dsn,$username,$password);
```

```
$commandq$connection->createCommand($sql);
```

```
// bizga kerak bo'lgan SQL so'rovlar berishimiz mumkin
```

```
// $command->text=$newSQL;
```

CdbCommand dan foydalanish bilan SQL so'rovlarni bajarishning ikki xil usul mavjud:

- execute(): INSERT, UPDATE va DELETE SQL so'rovlarini ishlatish uchun.

- query(): qaytuvchi ma'lumotlar oqiminig SQL so'rovlarini bajarishda, misol uchun SELECT so'rovi. Agar so'rov muvofaqiyatli bajarilsa CdbDataReader sinfi nusxasi yordamida qaytgan ma'lumotlarni olish xuquqiga ega bo'lamiz.

Agar jarayon paytida qandaydir xatolik bo'lsa unda bu jarayonni amalga oshirmaydi. Ya'ni jadvalda sizga halaqit qiluvchi va dasturni bizuvchi qoldiqlar qoldirmaydi.

```
$rowCount=$command->execute(); // 'INSERT', 'UPDATE'  
yoki'DELETE' turidagi so'rovlarni bajarish uchun
```

```
$dataReader=$command->query(); // 'SELECT' turidagi so'rovlarini  
bajarish uchun
```

```
$rows=$command->queryAll(); // so'rov natijasidagi barcha satrlarni  
olish
```

```
$row=$command->queryRow(); // so'rov natijasidagi birinchi satrni  
qaytaradi.
```

```
$column=$command->queryColumn(); // so'rov natijasidagi birinchi  
ustunni qaytaradi.
```

```
$value=$command->queryScalar(); //birinchi maydondagi birinchi  
satrni qaytaradi ya'ni birinchi katakcha qiymatini.
```

So'rov natijalari bilan ishlash

CdbDataReader sinfi nusxasidagi CdbCommand::query() orqali natijalarni yig'amiz va CdbDataReader::read() usuli orqali natijalarni satrma satr o'qiymiz. Ma'lumotlarni satrma satr o'qib olishimiz uchun shuningdek foreach orqali CdbDataReader dan foydalanamiz.

```
$dataReaderq$command->query();
```

```
// ko'p martalik read() murojaat bo'lib toki false qiymat  
qaytarmaguncha ishlaydi. Ya'ni olingan barcha ma'lumotlar hammasi  
tugaguncha
```

```
while(($rowq$dataReader->read())!qqfalse) { ... }
```

```
//ma'lumotlarga satrma satr murojaat uchun foreach dan  
foydalanamiz.
```

```
Foreach($dataReader as $row) { ... }
```

```
// barcha satrlarni massivga yig'amiz
```

```
$rowsq$dataReader->readAll();
```

Izoh: Barcha queryXXX() larning query() dan farqi to'g'ridan to'g'ri ma'lumot qaytarishidadir.

Misol uchun, queryRow() metodi joriy satr natijasi bo'yicha massiv qaytaradi.

Tranzaksiyadan foydalanish

Bir qancha so'rovlarni bir vaqtning o'zida bajarilishini, ya'ni hamma so'rovlar aniq bajarilishi kerak, biri bajarilmay qolgan holatda qolgan barcha bajarilgan so'rovlarni ortga qaytarib asl holiga qaytarib qo'yish kerak. Bunday holatda tranzaksiya ishlatiladi va CdbTransaction sinfidan foydalaniladi. CdbTransaction sinfini ishlash usuli quyidagicha:

- Tranzaksiya boshlanadi.

- So'rovlar birma bir amalga oshiriladi. Bunday holatda tranzaksiya o'zgargan ma'lumotlarni keshida saqlaydi, ma'lumotlar omborida xech qanday o'zgarish bo'lmaydi. Lekin ma'lumotlar ombori qaytarishi kerak bo'lgan ma'lumotlarni qaytaradi. Ya'ni siz saqlayotgan \$sql1 satrning id si kerak bo'lsa va bu attribute \$sql2 jadvalingiz uchun tashqi kalit sifatida kerak bo'lsa sizga ushbu idni qaytara oladi garchi ma'lumotlar omboriga saqlanmagan bo'lsa ham.

- Agarda tranzaksiyadagi so'rovlar muvofaqiyatli bo'lsa tranzaksiya amalga oshadi va ma'lumotlar omboridagi ma'lumotlarga o'zgarishlar kiradi.

- Agarda so'rovlar paytida qandaydir xatolik ro'y bersa, barcha so'rovlarni ortga qaytarib qo'yadi.

Bir kichik ko'rinishi quyidagicha:

```
$transaction=$connection->beginTransaction();
```

```
try
```

```
{
```

```
    $connection->createCommand($sql1)->execute();
```

```
    $connection->createCommand($sql2)->execute();
```

```
    // SQL-so'rovlari
```

```
    $transaction->commit();
```

```
}
```

```
catch(Exception $e)
```

```
{
```

```
    $transaction->rollback();
```

```
}
```

Model - foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni

doimo rolda nazorat qilib boradi.

Biz modelni ikki xil ko'rinishda hosil qilishimiz mumkin. Biz ma'lumotlarni olish, qayta ishlash, keyin attribute uchun CformModel dan foydalanamiz; agar ma'lumotni olib uni bazada saqlash bo'ladigan bo'lsa CActiveRecord dan foydalaniladi. Ikkala tur ham bir Cmodel sinfi asosida ishlaydi.

“attribute” – termini sinfning ayrim xususiyatlarini boshqa xususiyatlaridan ajratish uchun yaratilgan. Bunday xususiyat (attribute) asosan o'zida ma'lumotlar omboridan olingan yoki foydalanuvchi tomonidan kiritilgan ma'lumotni saqlashda qo'llaniladi.

Model sinfini aniqlaymiz:

Quyida biz LoginForm modelidan foydalanib, foydalanuvchining ro'yhatdan o'tish pageni yaratamiz.

```
class LoginForm extends CformModel {  
    public $username;  
    public $password;  
    public $rememberMeqfalse;  
}
```

Biz uchta o'zgaruvchi e'lon qilamizi, bular: \$username, \$password va \$rememberMe. Ular foydalanuvchi nomini, passwordini va shu bilan birgalikda username va passwordni eslab qolish belgisini (\$rememberMe). Bu belgi boshida false qiymatga ega bo'lmoqda.

Tekshiruv qoidalarini aniqlash: Foydalanuvchi formadan ma'lumotlarni yuborganda, model ularni oladi, biz ularni amalga oshirayotganda ma'lumotlar to'g'ri ekanligiga ishonch hosil qilishimiz kerak. Bu ishni amalga oshirayotganda qoidalar to'plami bilan ma'lumotlarni tekshiramiz. Tekshirish qoidalarini amalga oshirshda bizga rules() usuli yordam beradi.

```
class LoginForm extends CformModel {  
    public $username;  
    public $password;  
    public $rememberMeqfalse;  
    private $_identity;  
    public function rules()  
    {  
        return array(  
            array('username, password', 'required'),  
            array('rememberMe', 'boolean'),  
            array('password', 'authenticate'),  
        );  
    }  
}
```

```
);  
}  
public function authenticate($attribute,$params)  
{  
    $this->_identity=new UserIdentity($this->username,$this->password); if(!$this->_identity->authenticate())  
        $this->addError('password','Неправильное имя пользователя или пароль.');
```

NAZORAT SAVOLLARI

1. Framework dasturlash asoslari?
2. Symfony, Laravel va Yii freymworklar va ular bilan ishlash?
3. Freymwork platformada ma'lumotlar ba'zasi bilan ishlash?

23-BOB. WEB DASTURI MUHARRIRI ILOVALARI DREAMWEAVER MISOLIDA

23.1. Macromedia Dreamweaver 8 dasturi

Macromedia Dreamweaver 8 - bu web sahifa yaratish uchun eng qulay dastur hisoblanadi. Bu dastur yordamida siz nimalarni amalga oshira olishingiz mumkin.

1. Bu dastur avvalo sizga HTML kodlarini o'zi kiritadi ya'ni siz kod kiritib o'tirishingizning hojati yo'q.

2. Bu dasturni siz nafaqat HTML yoki CSS lar uchun balki boshqa bir qancha web-sayt uchun ishlov beriladigan dasturlar, ya'ni ColdFusion, PHP, ASP VBScript, ASP.NET, C#, JavaScript, XML, XSLT, kabilar uchun ham foydalanishingiz mumkindir.

3. Siz shuningdek bu dastur yordamida Internetdagi web-saytingizni nazorat qilishingiz mumkindir.

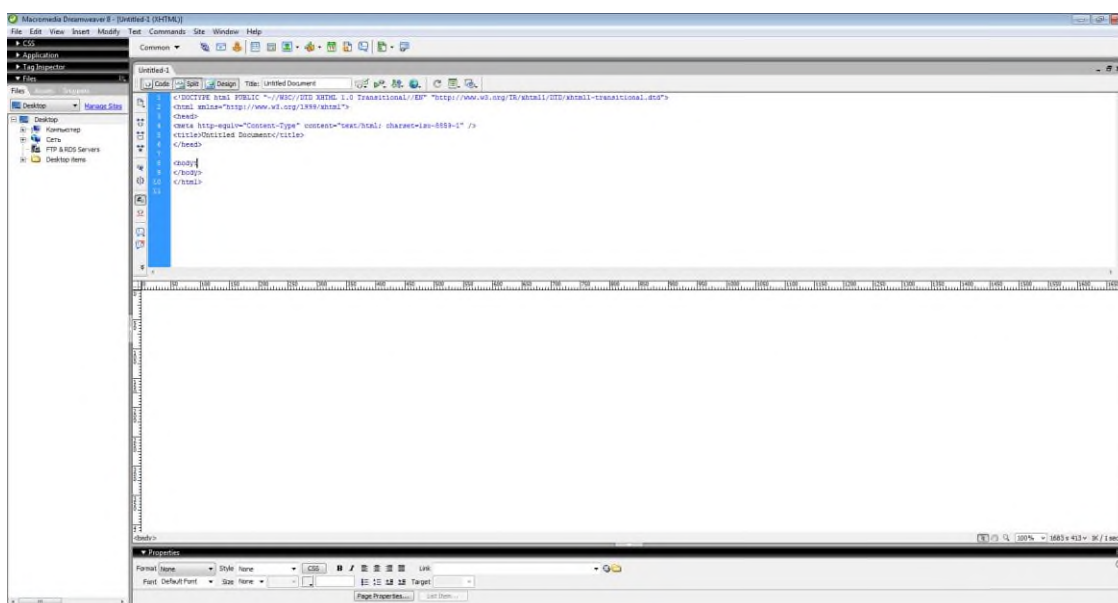
4. Hullas siz web-saytingiz uchun hamma-hamma narsani shu dastur yordamida bajarasiz.

Aksariyat professional web masterlar shu dasturdan foydalanishadi, albatta bunga o'xshash boshqa web page editor dasturlari ko'p masalan: Microsoft FrontPage, Adobe Pagemill, Adobe GoLive, NetObjects Fusion, CutePage, HotDog Proffesional, Netscape Composer va Arachnophilia, biroq kompyuter tehnologiyalariga ihtisoslashgan chet el unversitetlarida ham birinchi o'rinda Macromedia Dreamweaver o'qitiladi. Bu dasturni o'rganishning eng oson yo'li bu biror web-sayt yaratib o'rganishdir. Biroq avvalo bu dastuning interfeysi bilan tanishib chiqsak. Dreamweaver 8 ning asosiy ko'rinish quyidagichadir: Asosiy oyna uchga bo'lingan va o'ng tarafda yordamchi panel joylashgan. Yuqorida HTML kodlar joylashgan, markazda web-sahifa uchun ishlov berish joyi va quyida Properties joylashgan. Siz asosan Markaziy va Quyi qismlardan foydalanasiz, shuning uchun ushbu tugmalardan "Design" bo'limini tanlasangiz ham bo'ladi. Code bo'limi faqat HTML kod uchun, Split esa ikkalasi hamdir.

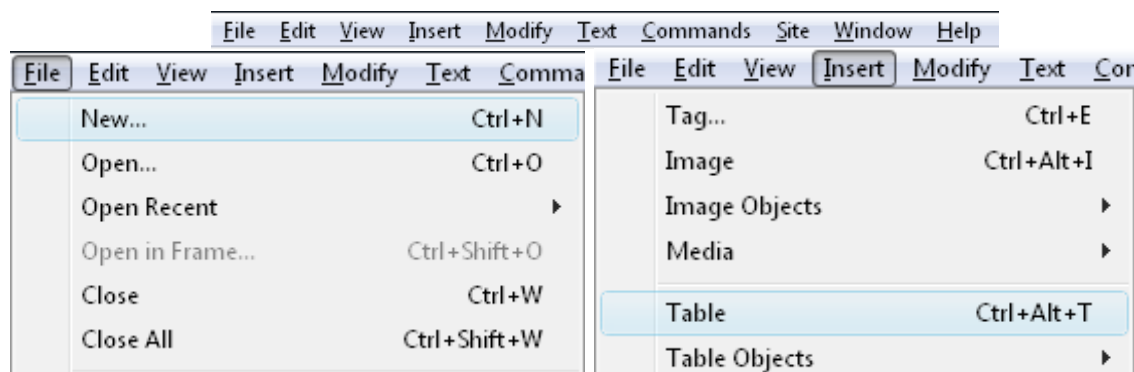
23.2. Macromedia Dreamweaver 8 dasturining ishchi muhiti

Eng yuqorida Menyu qatori. Asosiy buyruqlar kiritiladi. Har bir menyu MS Offise (Word, Exsel) dasturlari kabi o'z vazifalariga egadir.

Masalan yangi hujjat ochish kerak bo'lsa File→New, biror jadval yaratish kerak bo'lsa Insert→Table va hokazo.



23.1-rasm. Macromedia Denwer ishchi oynasi.



23.2-rasm. File→New va Insert→Table.

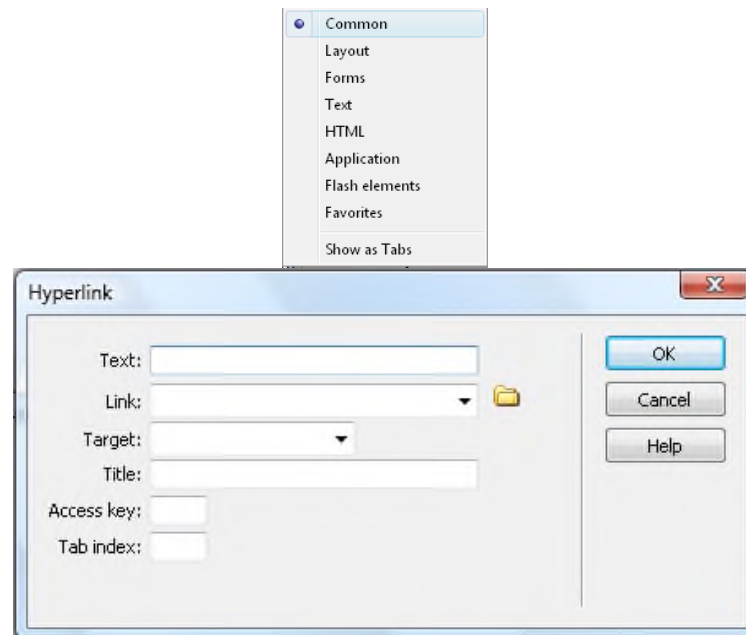
Insert Bar – bu qator sakkiz hil buyruqlar kiritish bo’limlaridan tashkil topgan.

Har birining o’ziga yarasha vazifalari bor. Bularning aksariyat buyruqlari Menyu bo’limi orqali kiritilishi ham mumkin lekin, Insert bar orqali osonroqdir.. Har bir buyruqlar bo’limini tanlashdagi pastga yo’naltirilgan strelkani bosish kerak bo’ladi. Va quyidagicha ro’yhat chiqadi.

Ushbu sakkiz hil buyruqlar kiritish bo’limini birma bir eng muhumlarini ko’rib chiqamiz. Siz ham Dreamweaver 8.0 dasturini ochib olib birga buyruqlarni kiritishni bir paytning o’zida mashq qilib boring. Quyida ko’rsatilingan rasmlarni aksariyatini bosishingiz bilan, ma’lum bir oynachalar shiqib keladi. O’sha oynachalarni to’ldirishingiz bilan, to’ldirgan ma’lumotlaringizga binoan HTML Kodlar bo’limida o’z o’zidan qo’yiladi.

Masalan quyidagi birinchi tugmani olaylik. Unga bosganimizda

ushbu oyna chiqadi.



23.3-rasm. Giper murojaatlar oynasi.

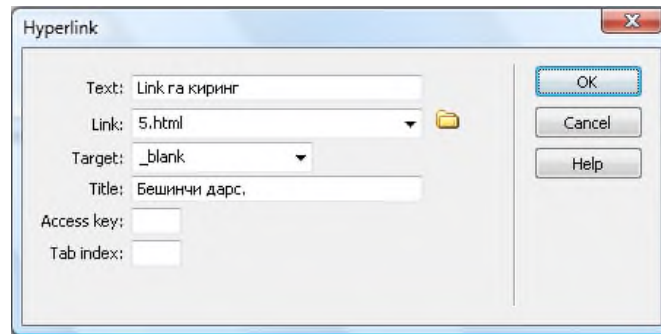
Va uni quyidagicha to'ldiramiz.

Va siz uchun HTML dokumentimizning kodlar bo'limida quyidagicha kodlar paydo bo'ladi. `Link ga kiring`

Common – eng ko'p qo'llaniladiganlarini o'zida mujassam etgandir.

Layout – asosan jadval tuzishda va unga ishlov berishda kerak bo'ladi. Forms – yuqorda ko'rsatilingan tugmalar faqat forma yaratishda ishlatiladi.

Text - yuqorda ko'rsatilingan tugmalar faqat oddiy matnga ishlov berishda ishlatiladi. Ularning vazifalarini HTML darslaridan juda yahshi bilamiz. Biz o'zi mavjud bo'lmagan, bir kafeni websaytini yaratish orqali Dreamweaver 8.0 ni yanayam yahshiroq o'zlashtiramiz degan umiddaman. Masalan Kafe web saytini yaratish uchun quyidagi vazifalar bajariladi, kafe nomini “Cafe Townsend” deb nomlanadi. Biz bu websaytning barcha sahifalarini o'zimizning kompyuterga saqlab va to'plab boramiz, va darsimiz so'ngida bu tayyor sahifalarni Internetga joylashni o'rganamiz. Darslarni o'rganish mobaynida Internetga ulangan bo'lishingiz shart emas. Biroq Macromedia Dreamweaver 8.0 kompyuteringizga o'rnatilgan bo'lishi shart. Har bir bajarishingiz kerak bo'lgan jarayonni son ketma ketligida → bilan belgilab boraman.



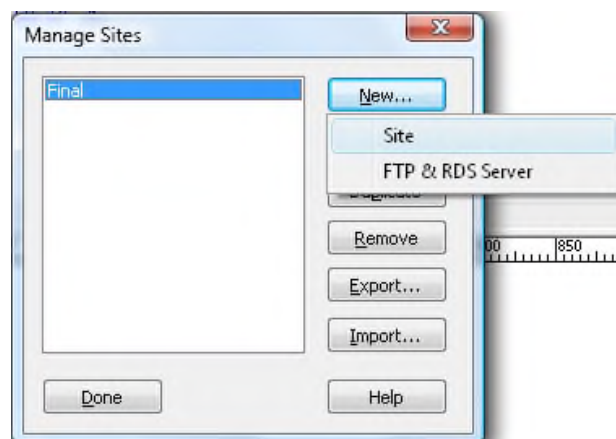
23.4-rasm. Giper murojaat oynasini to'ldirish.

1→ Kompyuteringizning HARD DISSCI (D) ida local_sites deb nomlanga yangi papka oching. Mening kopyuterimda bu papka ushbu manzilda joylashgan D:\local_sites 2→Dreamweaver 8.0 ni o'zini kompyuteringizga o'rnatilingan papkalari ichida cafe_townsend deb nomlangan yana bitta papka bor, faqt o'sha papkani local_sites papkangiz ichiga ko'chirib oling (kesib emas). Menda bu papka ushbu manzilda joylashgan.

S:\Program Files \ Masromedia \ Dreamweaver 8 \ Tutorial_assets \ cafe_townsend.

3→Navbatdagi bajarishimiz kerak bo'lgan narsa "Manage site" orqali saytimizni faqat o'zimizgina ko'rishimiz mumkin bo'lgan holatda joylashdir. Buni sayt yaratishdan oldin birinchi bo'lib bajarib olish kerak. Agar bunday qilinmasa Dreamweaver ning ba'zi funksiyalari bajarilmasligi mumkin. Shuningdek bu holda websayt yaratish ancha qulay. Buning uchun quyidagilarni bajaring.

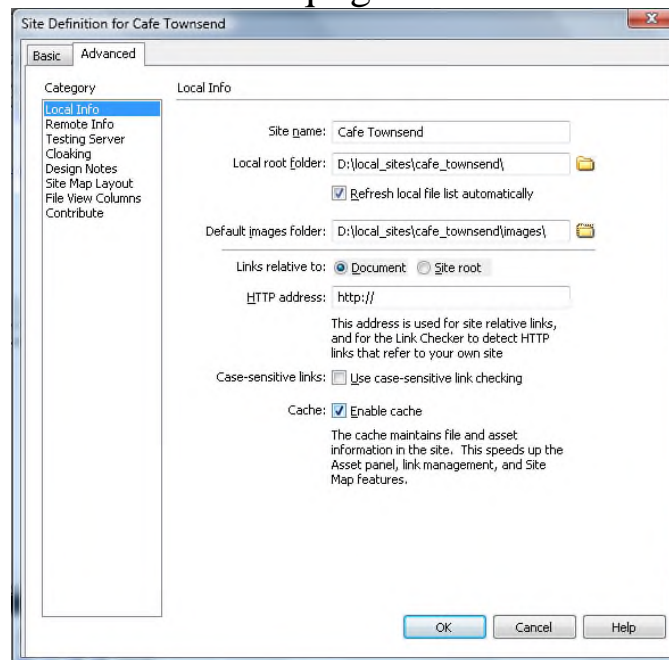
1. Menyu qatoridan Site > Manage Sites ni tanlang. Va quyidagi oynachadan New > Site ni tanlang. Quyidagi rasmdagi Final ga e'tibor bermang.



23.5-rasm. Saytni boshqarish oynasi.

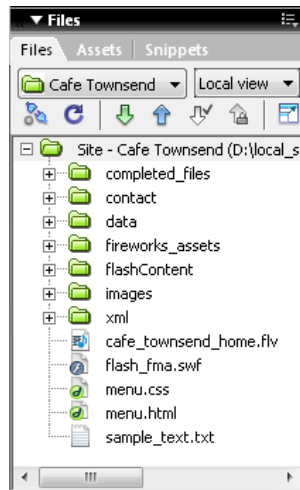
2. Quyidagi oynada boshida e'tibor bergan bo'lsangiz Basis (oddiy)

va Advanced (mukammal) turibdi bizga Advanced bo'limi kerak. Va uni quyida ko'rsatilingandek to'ldirib chiqing.



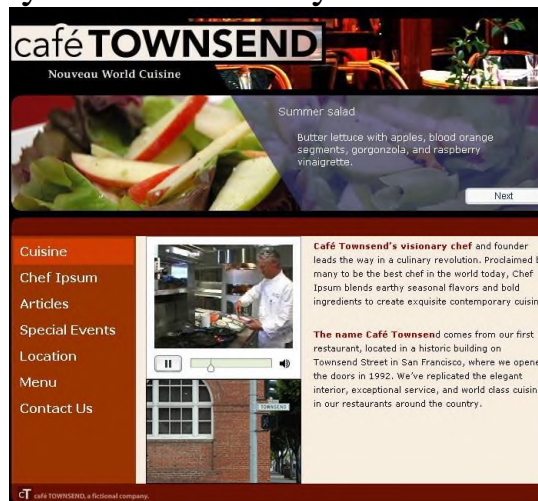
23.6-rasm. Advanced bo'limini tanlash va uni to'ldirish.

- Local Info
- a. Site name: Sayt nomi: cafe Townsend bo'lsin.
 - b. Local root folder: cafe_townsend papkangiz joylashgan joyini ko'rsating.
 - c. Default images folder: rasmlar joylashgan papka manzili.
 - d. Links relative to: Dosument da tursin.
 - e. HTTP address: Hozirsha bo'sh tura tursin, qandoq bo'lsa shundoq turadi.
 - f. Cache: Enable cache.
 - g. Ohirida OK ni bosing.
- Remote Info, Testing Server, va hokazolarga hozirsha tegmay turamiz.
3. OK bosganingizda ushbu oynacha chiqadi va keyin Done ni bosing.
 4. Files paneli Dreamweaver Dokumentingizning o'ng tarafida quyidagicha ko'rinishda ochiladi.



23.7-rasm. File paneli ko'rinishi.

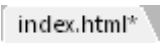
Endi bajarishimiz kerak bo'lgan narsa bu websayt yaratishni boshlashdir. Yodingizda bo'lsin websaytlar asosan jadvallar ichiga yasaladi. Demak websayt yaratishdan oldin websaytimiz qanday strukturada joylashishini biror qog'ozga chizib chiqishimiz kerak. Keyin unga qarab jadval tuzamiz. Darhol Dreamweaver ni ochib websayt tuzib ketilavermaydi. Keyin tayyor bo'lgan jadvalimizning har hil qismlariga har hil ma'lumotlarni o'rnatishimiz kerak bo'ladi. Biz quyida ko'rsatilingan websaytni yaratishni boshlaymiz.



23.8-rasm. Web saytning ko'rinishi.

1 → Buning uchun birinshi navbatda yangi HTML dokument ochamiz va uni index deb saqlab qo'yamiz.

- a. File > New → Basis page > HTML
- b. File > Save As
- c. File name: index deb yozsangiz bo'lgani (index.html) kerak emas.
- d. Ohirida Save.

2→ Keyin Title: bo'limiga Café Townsend deb yozing. Vaqti vaqti bilin o'zingizni bilib klaviatura yordamida (Ctrl+s) dokumentingizni saqlab boring.  bu rasmdagi ushbu yulduzcha dokumentga so'nggi o'zgartirish kiritilgandan buyon saqlanmadi, degan ma'noni beradi.

3→ Yuqorida takidlaganimizdek endi navbat Jadval tuzishga. Jadvalni 2 hil yo'l bilan tuzishimiz mumkin.

Birinshisi: dagi birinchi rasimga bosish orqali. Ikkichisi: Menyu qatoridan Insert > Table orqali.

Jadvalni 3 ta qatorga 1 ta ustun qilib quyidagicha belgilab olamiz. Umumiy kengligini esa 700 piksel qilib olamiz

- a. Rows: jadval qatorlari soni 3 ta
- b. Columns: jadval ustunlari soni 1 ta
- c. Table width: Umumiy kengligini esa 700 piksel.
- d. Qolganlarini o'zgartirmaymiz.
- e. Header: None da tursin
- f. OK

4→ Sizda ushbu ko'rinishdagi jadval tuziladi. O'zingizni websaytingizni yaratayotganingizda biron qator yoki ustun ortiqcha yoki kam bo'lib qolsa, o'chirib yoki qo'shishingiz mumkindir.

Agar kursoringiz jadval ishida tursa yashil shiziqlar bilan, har bir ustunlar kengligi va jadvalning umumiy kengligi ko'rinib turadi. Quyidagi misolda (tuzishingiz shart emas) 700 umumiy jadvalning kengligi, 140, 230, 330 lar esa ustunlar kengligidir. O'lchamlar piksel yoki foiz da bo'lishi mumkin. Kursorni jadvaldan tashqariga chiqarsingiz sichqonchani jadvalning o'ng tamoniga tashqariga bosish orqali, bu o'lcham ko'rsatkichlari yo'qoladi. Bu ko'rsatkichlar shunchaki sizga eslatma sifatidadir. Agar hohlasangiz bunday eslatmani o'chirib qo'yishingiz mumkindir. Uning uchun siz Menyudan View > Visual Aids > Table Widths ga bosishingiz kerak bo'ladi.

5→ Kursorni jadvaldan tashqariga sichqonchani jadvalning o'ng tamoniga bosish orqali chiqaring, klaviaturadan hech qanday tugmani bosmagan holda. Yana bitta ikkinchi jadvalni tuzamiz. Bu gal 1 ta qator, 3 ta ustun qilib belgilaymiz. Va qolgan o'lchamlarni o'zgartirmaymiz. OK bosing.

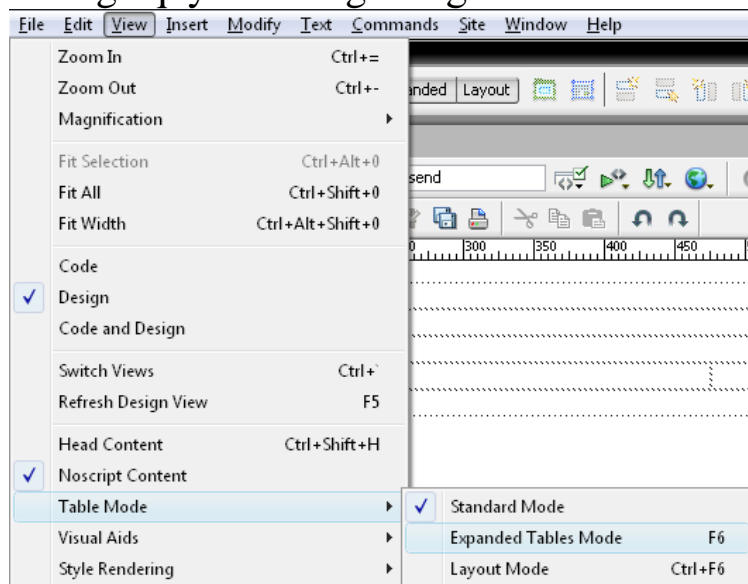
6→ Kursorni jadvaldan tashqariga sichqonchani jadvalning o'ng tamoniga bosish orqali chiqaring, klaviaturadan hech qanday tugmani bosmagan holda. Yana bitta uchunchi jadvalni tuzamiz. Endi esa 1 ta qator, 1 ta ustun qilib belgilaymiz. Va qolgan o'lchamlarni

o'zgartirmaymiz. OK bosing.

Sizni umumiy hama jadvalingiz ushbu ko'rinishga egadir.

7→ Endi vaqtinchalik jadvalimizni o'lchamlarini belgilashni osonlashtirish maqsadida cell padding va cell spacing joylashtiramiz. Uning uchun shunshaki: View > Table Mode > Expanded Tables Mode (F6) yoki quyidagi qatordan Expanded ga bosamiz.

Va quyidagi natijani olamiz. Biroq o'lchamlarni joylab bo'lgach oldingi Standard Mode ga qaytishimizga to'g'ri keladi.



23.9-rasm. View oynasi.

8→ Kursorni yuqoridan eng birinchi qator ichiga joylagan holda, eng quyidagi bo'limidan "H" ya'ni height (balandlik)ni 90 (piksel) qilib belgilaymiz.

9→ Kursorni yuqoridan ikkinchi qator ichiga joylagan holda, eng quyidagi bo'limidan "H" ya'ni height (balandlik)ni 166 (piksel) qilib belgilaymiz.

10→ Kursorni yuqoridan uchinchi qator ishiga joylagan holda, eng quyidagi bo'limidan "H" ya'ni height (balandlik)ni 24 (piksel) qilib belgilaymiz. Spu

bilan birinchi jadvalimiz o'lchamlari tugaydi.

Birinchi Jadvalimizning o'lchamlari o'rnatilganidan so'ng umumiy ko'rinishi quyidagichadir.

11→ Endi ikkinchi jadvalimizning o'lchamlarini kiritamiz. Kursorni chapdan birinchi ustun ishiga joylagan holda, eng quyidagi **▼ Properties** bo'limidan "W" ya'ni width (kenglik)ni 140 (piksel) qilib belgilaymiz.

12→Ikkinchi va uchunchi ustunlar uchun ham W ni 230 va 330 qilib mos ravishda belgilab olamiz. Va bizda umumiy jadvalimiz quyidagicha

ko'rinishga ega bo'lishi kerak bo'ladi. H uchun hech qanday o'lcham kiritmaymiz chunki keyinchalik bu kataklar ichiga joylashtiradigan ma'lumotlarimizga qarab jadval balandligi o'zgarishi mumkindir.

13→ So'ngi jadvalimizning esa (H)ni 24 (pixel) qilib belgilab olamiz.

Eng so'nggida jami jadvallarimizning umumiy ko'rinishi quyidagicha bo'lishi kerak bo'ladi.

14→ Keyin Boshida ta'kidlaganim Standard Mode ga qaytishimiz kerak bo'ladi. Uning uchun ikki hil uslubni qo'llashimiz kerak bo'ladi. Birinchisi: Menyu qatoridan View > Table Mode > Standard Mode

Ikkinchisi: joydan Standard tugmachasiga bosishimizdur. Va Standard holatga o'tkazganingizdan so'ng dokumentingizni saqlang. Image Placeholder va Rang berish

Website yaratishimizda eng yuqori qatorda odatda biror rasm yoki animatsiya bo'lishi kerak bo'ladi. Hatto uni oldindan yaratmagan bo'lsangiz ham u uchun joy ajratib qo'yishingiz kerak, va u joy odatda image placeholder (rasm joyi deb nomlanadi) bo'ladi. 1→ Jadvalimizning eng yuqori qatoriga kursorni joylagan holda dokumentimizning Menyu qismidan Insert > Image Objects > Image Placeholder ga kiramiz.

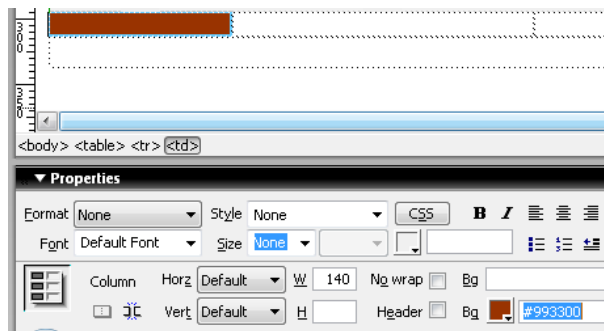
2→Undan so'ng quyidagi oynacha chiqib keladi va uni quyidagicha to'ldirishimiz kerak bo'ladi.

- a. Name: banner_graphic
- b. Width: 700
- c. Height: 90
- d. Color: #993300
- e. Alternative text: esa bo'sh qola tursin.
- f. Va ohirida OK

3→Endigi qilishimiz kerak bo'lgan ish, bu jadvalimizning ba'zi kataklariga rang berib chiqishimizdir. Uning uchun birinchi navbatda uch ustunlik jadvalimizning birinchi ustuniga kursorni joylaymiz.

4→Biz shu holda umumiy katakni tanlashimiz kerak bo'ladi, uning uchun ishlash panelimizning quyi qismidagi <td> belgisini tanlashimiz kerak bo'ladi

5→ Dokumentimizning eng quyi qismidagi Properties bo'limidan background color (bg) dan #993300 rangini tanlang va Enter ni bosing. U quyidagicha ko'rinishga ega bo'lishi kerak bo'ladi.



23.10-rasm. Ranglarni tanlash.

6→ Endi esa ikkinchi ustunni tanlashimiz kerak bo'ladi va u uchun biz yana kursorni katak ichiga joylaymiz va <td> ni tanlaymiz, #F7EEDF rangni bg color uchun belgilaymiz.

7→ Uchunchi ustun uchun ham (6→) ni takrorlaymiz. Va jadvalimiz ushbu ko'rinishga ega bo'ladi.

8→ Endigi qilishimiz kerak bo'lgan narsa bu butun bir sahifamizning o'zini Page Properties yordamida background rangini qora (#000000) qilib belgilab olishdir, siz shuningdek text turi, matn o'lchami va hokazolar uchun ham bu yerda moslab olishingiz mumkindir, biroq hozircha faqat rangini o'zgartiramiz. Page Properties oynasini ikki yo'l bilan ochishimiz mumkindir.

Birinchisi: Albatta Menyu qatoridan Modify > Page Properties.

Ikkinchisi: Dokumentimizning quyi qismidagi Properties bo'limidagi, Page Properties tugmachasi yordamidadir, u tugmacha yuqoridagi rasmda o'ng-quyi burchakda joylashgandir.

9→ Rangni tanlab bo'lganimizdan keyin boshqa hech narsani o'zgartirmagan holda OK ni bosamiz. Va sizning butun bir sahifangiz qora rangga aylandi. Shu yerda dokumentingizni yana bir bor saqlab qo'ying.

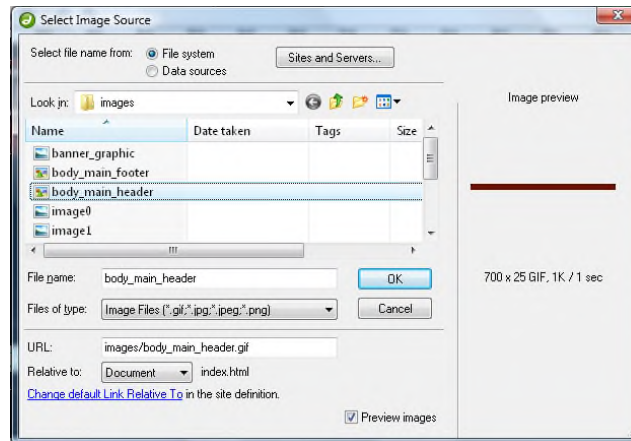
Web sahifamizni Matnini to'ldirish. Bu darsda qilishimiz kerak bo'lgan narsa jadvalimizning ichini asosiy ma'lumotlar bilan to'ldirishdir. Dars so'nggida bizning sahifamiz quyidagicha ko'rinishga ega bo'lishi kerak bo'ladi. Keyin biz CSS document yaratib HTML bilan bog'lab biriktiramiz shunda matnimiz yanayam ko'rkamlashadi.

1→ Jadvalimizning eng yuqoridagi bo'limiga (banner_graphic) ga ikki marta sichqonchani bosing. Va images papkasidan banner_graphic rasmini tanlang, va OK ni bosing. Jadval tashqarisiga sichqonchani bosish orqali umumiy ko'rinishini ko'ring. Agar browserda ko'rish niyatingiz bo'lsa F12 ni klaviaturadan bosing. Va shu yerda yana bir dokumentingizni saqlang.

2→ Birinchi jadvalimizning yuqoridan uchunchi qatori ichiga

kursorni joylang. Menyu qismidagi Insert > Image dan body_main_header.gif faylini tanlang va OK ni bosing. Ushbu holat quyida ko'rsatilingandir.

3→ Quyidagi ushbu oynachaga e'tibor bermagan holda shunchaki OK ni bosing.



23.11-rasm. Fayl tanlash.

4→ Uchinchi jadvalimizning qatori ichiga kursorni joylang. Menyu qismidagi Insert > Image dan body_main_footer.gif faylini tanlang va OK ni bosing. Keyingi chiqqan oynaga ham OK ni bosing. Siz shuningdek Files bo'limidan rasmni sichqoncha yordamida ushlab katak ichiga olib kelib tashlash yordamida ham rasmlarni joylashingiz mumkindir. Ushbu holat quyida ko'rsatilingandir.

5→ Dokumentingizni shu yerda yana bir saqlab oling. Web sahifangiz umumiy ko'rinishi quyidagichadir.

6→ Uch-ustunlik jadvalimizning ikkinchi ustuniga kursorni joylagan holda Assets Bo'limidan rasm joylash uslubini ham bir ko'rib o'tamiz. Buning uchun avval katak ichidagi ma'lumotni vertikal joylashuvini yuqoridan va gorizontal joylashuvini esa markazdan qilib belgilaymiz. Uning uchun kursor katak ichida turgani holda Properties bo'limidan ushbu ko'rsatkichlarni tanlaymiz.



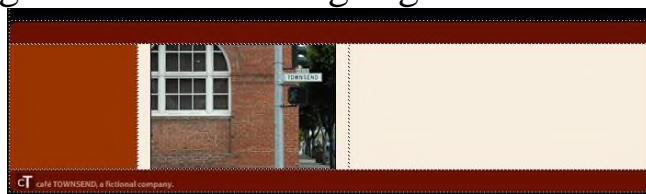
23.12-rasm. Rasm joylashuvini berish.

So'ngra quyidagi ko'rsatilingan rasmni sichqoncha yordamida ushlagan holda shu katakga joylab, chiqib kelgan oynachaga OK ni bosamiz.



23.13-rasm. Rasmni yuklash.

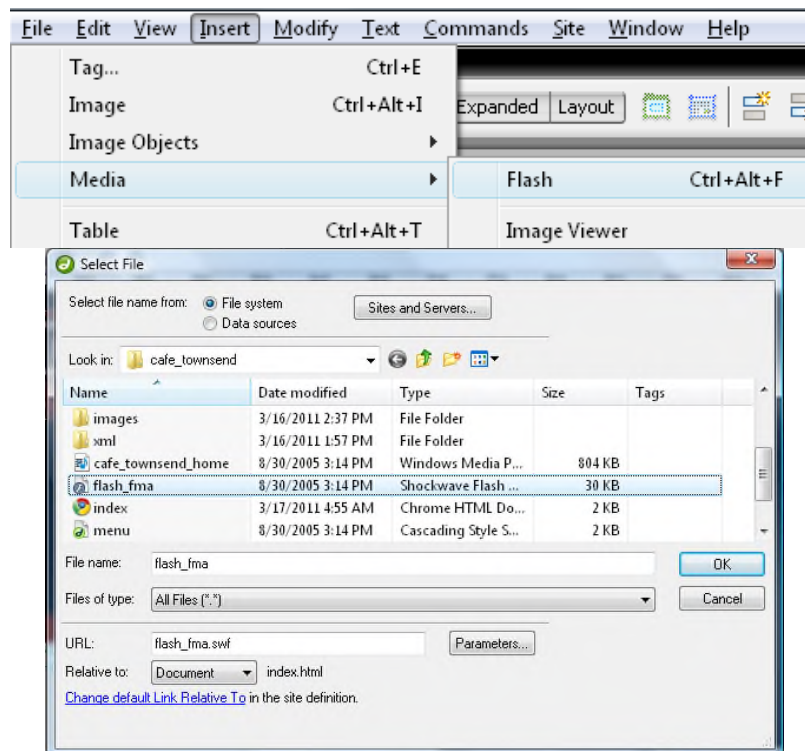
Sizning sahifangiz ushbu ko'rinishga ega bo'lishi kerak.



23.14-rasm. Yuklangan rasm natijasi.

7→ Endigi navbat Flash faylgadir. Flash fayllar odatda websaytlarda juda keng qo'llaniladi, ular rasmlarni o'zi navbatma navbat almashinib turishini ta'minlaydi, bunday flash fayllar tayyorlash Dreamweaverda mumkin emas, mahsus dasturlarda tuziladi, biroq biz hozircha tayyor yasalganini olib joylaymiz, broq vaqti kelib bularni ham o'rganamiz. Buning uchun birinchi jadvalimizning ikkinchi qatorini tanlaymiz va kursorni joylaymiz. Insert > Media > Flash yordamida flash faylni

tanlaymiz.



23.15-rasm. Flash yordamida flash faylni tanlash.

U fayl yuqoridako'rsatilgani kabi bo'ladi va OK ni bosib, keying chiqqan oynachaga OK ni bosib.

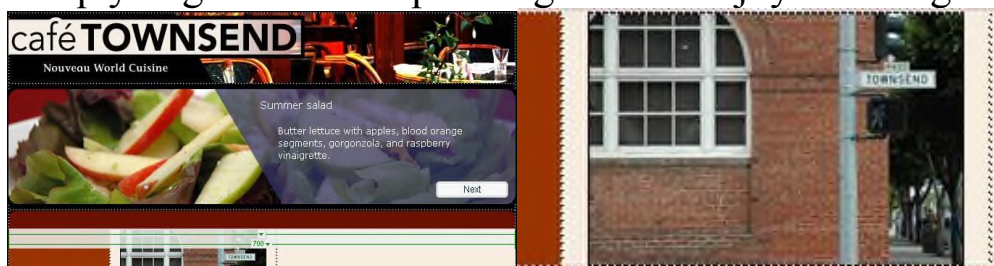
8→ Sizda quyidagicha ko'rinish paydo bo'ldi.



23.17-rasm. Flash fayl yuklangan oyna ko'rinishi.

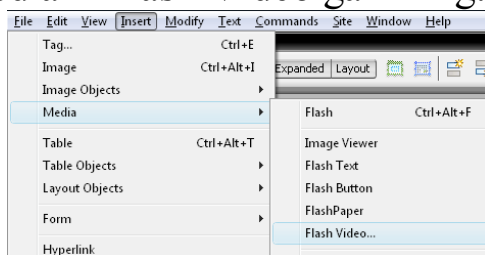
Biroq flash faylni ko'rish uchun uni sichqoncha bilan jadval ichida yuqorida ko'rsatilgan holda tanlab Properties bo'limidan ni bosishingizga to'g'ri keladi. Va quyidagi natijani olasiz.

9→Endi quyidagi rasmni chap tarafiga kursorni joylashtiring.



23.18-rasm. Kursorni joylashtirish.

10→ Va Insert > Media > Flash Video ga kiring.

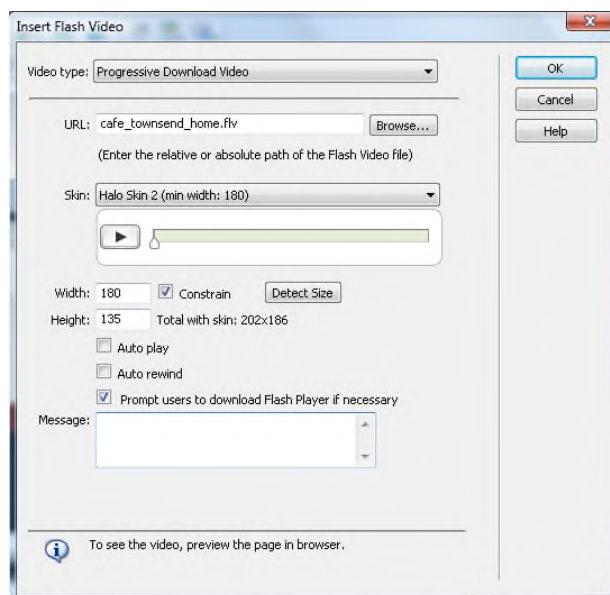


23.19-rasm. Video yuklash oynasi.

11→ Ushbu oyna huddi ko'rsatilingani kabi to'ldirilsin. Va OK

- a. Video type: Progressive Download Video
- b. URL: cafe_townsend_home.flv
- c. Skin: Halo Skin 2
- d. Width: 180
- e. Height 135

Constrain: eni va bo'yi o'lchamining nisbatini belgilash uchun Auto Play: web sahifa ochilganda o'zi ishga tushsinmi yoki websayt foydalanuvchisi o'zi play ni bosgandami degan tushunchani belgilash uchun. Auto rewind: video flash tugaganda yana boshiga qaytarish uchun.



23.20-rasm. Videoni joylash oynasi.

Endi websahifamizga matn kiritishni boshlaymiz.

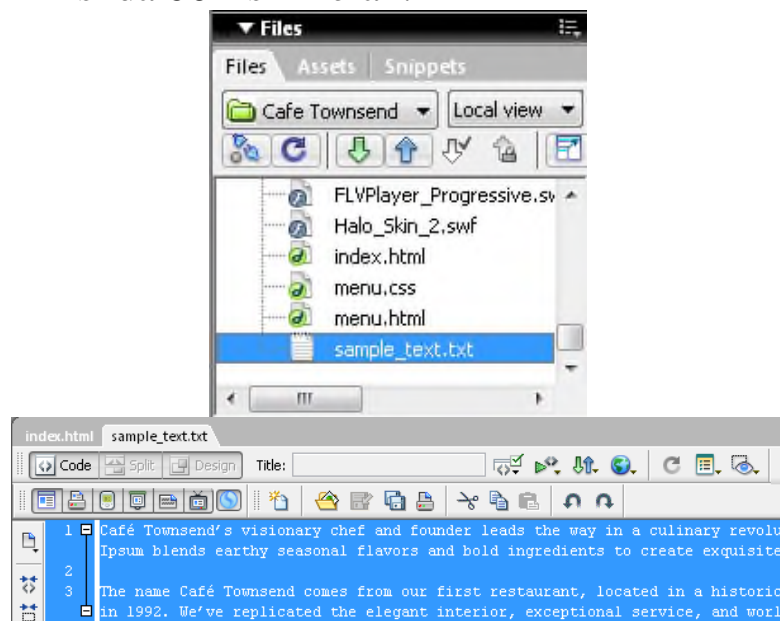
1→ Files panelida sample_text.txt dokumntimi sichqonchni ikki marta bosish orqali Dreamweaverda oching.

2→ Matnni qora qilib ko'chirgan holda uch-ustunlik jadvalimizning uchunchi ustuniga joylang. Va shu yerda yana bir bor dokumentingizni

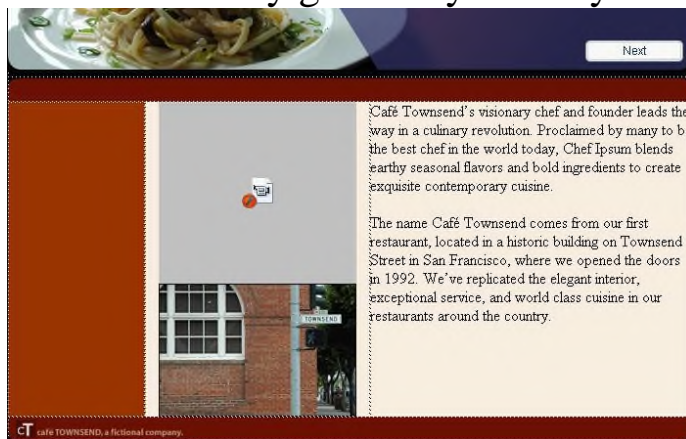
saqlab qo'ying.

3→ Joylangan matnni, vertikal joylashuvini kursorni matn ichida turgani holida Properties bo'limidan Vertikal ni Top qilib oling. Horizontal bo'limiga tegman. Endi F12 yoki boshqa yo'l bilan Internet Explorer da website ko'rinishida ochib ko'ring.

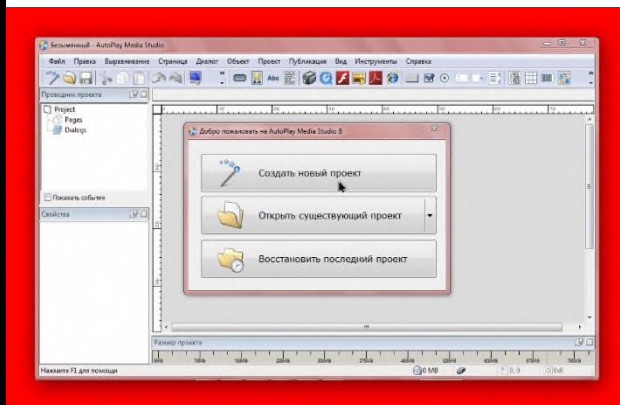
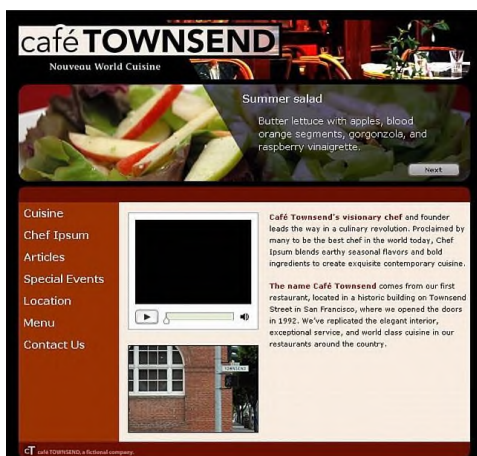
CSS dokumentini HTML dokumentga bog'lash so'ngida siz yaratgan website ushbu ko'rinishda bo'lishi kerak.



23.21-rasm. Saytga matn yozish oynasi.

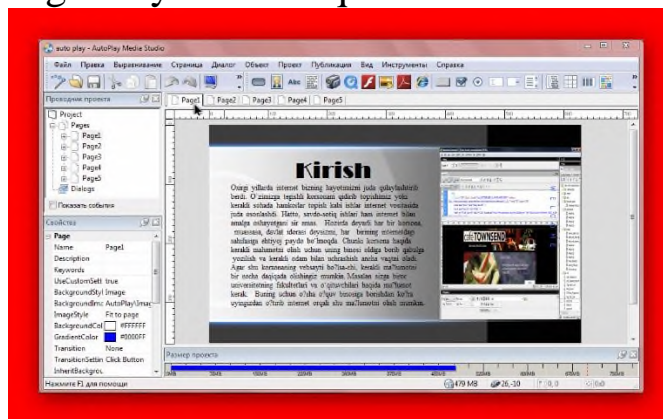


23.22-rasm. Matnni saytda ko'rinishi.



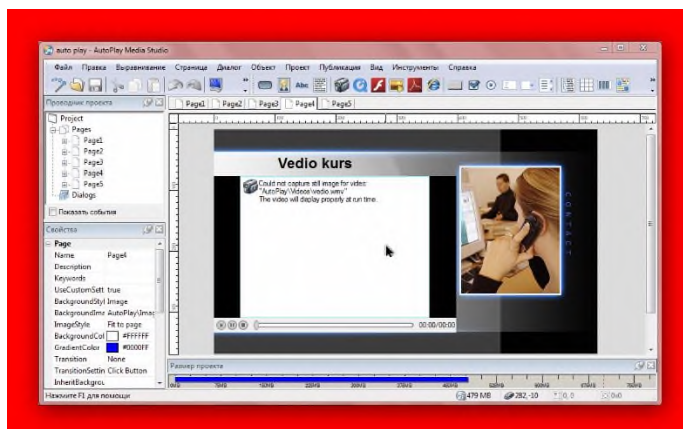
23.23-rasm. Autoplay yaratish jarayoni.

Bu autoplayning yaratish jarayoni. Bunda yangi loyiha yaratishimiz mumkin yoki yaratilgan loyihani chaqirib olishimiz mumkin.



23.24-rasm. Birinchi sahifa ko'rinishi.

Bu birinchi sahifa bo'ladi. Shunga o'xshagan bir nechta sahifalarlar yaratiladi.



23.25-rasm. Video qo'yish jarayoni.



23.26-rasm. Bu autoplayda yaratilgan loyihaning ko'rinishi.

Bunda pdf formatdagi dokumentni yaratish ko'rsatilgan. Buning uchun Adober reader programmasini o'rnatish kerak. Shundagina ishlaydi.

NAZORAT SAVOLLARI

1. Macromedia Dreamweaver 8 dasturi va uni o'rnatish jarayoni?
2. Macromedia Dreamweaver 8 dasturining ishchi muhiti?
3. Web sayt yaratish bosqichlari?

24-BOB. WEB SERVERNI SOZLASH. XAMPP, DENVER, WAMPP, OPEN SERVER PAKETLAR

24.1. XAMPP – krossplatformali web – server jamlanmasi

XAMPP – krossplatformali web – server jamlanmasi, Apache, MySQL, PHP skript interpretatori, Perl dasturlash tili va qoʻshimcha kutubxonalar katta raqamini oʻz ichiga olgan Web server - toʻlaqonli web serverni ishlatish imkonini beradi.

XAMPP deb nomlanishi quyida soʻzlardan olingan

- X – ixtiyoriy operatsion sistema;
- Apache;
- MySQL;
- PHP;
- Perl;

Toʻliq paketi quyidagilarni oʻz ichiga oladi:

- SSL bilan Apache web-server;
- MySQL maʼlumotlar bazasi;
- PHP;
- Perl;
- FileZilla FTP-serveri;
- POP3 / SMTP server;
- phpMyAdmin utilitasi;

XAMPP 32-bit Microsoft operatsion tizimlari bilan ishlaydi (98/2000/XP/2003/ Vista/7), shuningdek, Linux, Mac OS X va Solarislar bilan ham ishlashi mumkin. Dastur erkin litsenziya - GNU General Public Licence litsenziyasi ostida tarqatilgan va dinamik sahifalarni yozishga xizmat qiladi.

Linux uchun XAMPP. Linux uchun Ubuntu, ASILA, RedHat, Mandriva, Debian ustida sinovlar olib borilgan va paket quyidagilarni oʻz ichiga oladi: Apache, MySQL, PHP 5 + PHP 4 va nok, Perl, ProFTPD, phpMyAdmin, OpenSSL, GD, Freetype2, libjpeg, libpng, gdbm, zlib, expat, Sablotron, libxml, Ming, Webalizer, pdf sinf, Ncurses, tartib Perl, FreeTDS, Gettext, Mcrypt, Mhash, eAccelerator, SQLite va IMAP C-Mijoz, FPDF.

Mac OS X uchun XAMPP. Mac OS X uchun paketi quyidagilarni oʻz ichiga oladi: Apache, MySQL, PHP va nok, SQLite, Perl, ProFTPD,

phpMyAdmin, OpenSSL, GD, Freetype2, libjpeg, libpng, zlib, Ming, Webalizer, mod_perl, eAccelerator, phpSQLiteAdmin.

Solaris uchun XAMPP. Solaris 9 sinov, Solaris 8 uchun ishlab chiqilgan va sinov qilingan. Solaris uchun paket quyidagilarni o'z ichiga oladi: Apache, MySQL, PHP va nok, Perl, ProFTPD, phpMyAdmin, OpenSSL, Freetype2, libjpeg, libpng, zlib, expat, Ming, Webalizer, pdf sinfi.

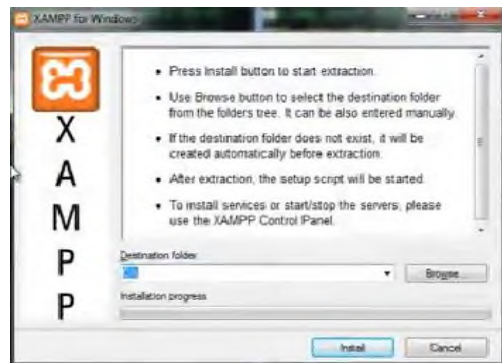
Talablar va xususiyatlari. XAMPP o'rnatish uchun formati .zip, .tar yoki .exe ko'rinishdagi o'rnatuvchisini yuklab olish kerak. Dasturning qismlariga xususiylashtirish shart emas. Dastur muntazam ravishda Apache / MySQL / PHP va Perl ning eng so'nggi versiyasini yangilab boradi. Bundan tashqari, XAMPPda OpenSSL va phpMyAdmin va boshqa modullar bor. Foydalanuvchi interfeysi («dangasa odamning WAMP / LAMP o'rnatuvchisi») oddiy qilib dangasa uchun montaj, deyiladi. XAMPPni o'rnatish alohida har bir komponentining o'rnatishdan ko'ra kamroq vaqt talab etadi. Ushbu web-server (XAMPP Lite sifatida ham tanilgan), to'liq standart va tushgan versiyalarida taqsimlanadi. Barcha qo'shimcha modullarni ham ko'chirib olish mumkin. Qo'shimcha imkoniyatlari biri, kompaniyaning o'zi yangi paketlarni .zip, 7-zip, .tar yoki .exe ko'rinishida hammasini yangilashga imkoniyatini beradi.

Windows tizimiga o'rnatish va olib tashlash. Installer (taxminan 100 MB) yordamida - XAMPPni eng oson o'rnatish mumkin. O'rnatishdan so'ng, XAMPPni Пуск / Все программы / XAMPP yo'lidan topish mumkin. XAMPPni boshqarish paneli yordamida serverlarni yoqish yoki o'chirish mumkin. Bundan tashqari, o'rnatilgan holda yuklash ham mumkin. XAMPPni o'chirish uchun barcha yoqilgan serverlarni, XAMPP dasturini o'chirish va XAMPP papkasini o'chirib tashlash kerak. Agar instalyator yordamida o'rnatilgan bo'lsa o'rnatilgan dasturlar ro'yhatidan o'chirish orqali o'chirib yuborish mumkin.

Foydalanish. Internetdan foydalanmasdan o'z ishini sinab ko'rish uchun web-dizaynerlar va dasturchilar, ishlab chiquvchilari uchun XAMPP vositasi sifatida yaratilgan. XAMPP da ishlayotganda ba'zi imkoniyatlar xavsizlik standart xolatda o'chirilgan bo'ladi. Shu bilan birga, XAMPP ba'zan World Wide Web sifatida ishlatiladi. Dastur, shuningdek, ma'lumotlar bazalari MySQL va SQLite yaratish va boshqarish qo'llab-quvvatlaydi. XAMPP-serverda joylashgan htdocs papkasiga yangi sayt uchun sayt fayllarni ko'chirib olish mumkin.

XAMMP paketi va uning qulayliklari. XAMMP paketi xuddi Microsoft Office paketiga o'xshash bo'lib unda u o'ziga bir kancha web

dasturlashga uchun kerak bo'ladigan dasturlar to'plami. Bu paketda Apache, MySQL, FileZilla, Mercury, Tomcat dasturlaridan iborat. Paketni o'rnatish uning o'rnatuvchi fayliga borib sichkonchani 2 marotaba chertsak. Quyidagi oyna xosil bo'ladi:



24.1-rasm. XAMMP paketi.

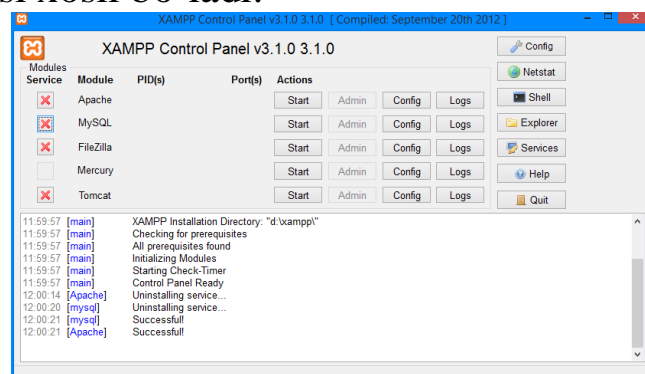
Xosil bo'lgan oynadan biz paketni qaerga o'rnatish kerakligini ko'satishimiz yoki taklif qilingan joyga o'rnatishimiz mumkin. Bu oynanada taklif qilingan joy: C:\. C: diskka o'rnatishni taklif etyapti. Agar biz C: diskka o'rnatadigan blo'sak kompyuterimizni yangi sistema o'rnatganimizda paketdagi malumotlar o'chib ketadi. Paketdagi malumotlarni saqlab qolish uchun uni D:\ diskka o'rnatish ko'proq tavsiya etiladi. Buning uchun yuqorida xosil bo'lgan oynadan Browse.. tugmasini tanlab o'rnatilishi kerak bo'lgan joy yani D:\ diskni ko'rsatamiz.

O'rnatiladigan joyni ko'sartganimizdan so'ng Install tugmasini bosamiz va dastur o'rantilishi boshlanadi.

Bu oynada so'ng paketni yorlig'ini ishchi stoliga o'rnatishni taklif qiluvchi oyna xosil bo'ladi.

Bu oynadan «y» tugmasini tanlaymiz va ishchi stolga yorlig'i qo'yiladi.

Bu oynada antivirus dasturiga ishonchlilik bildirish kerak bo'ladi va dastur ishlashga tayyor xolatga keladi. Paketni ishga tushirish uchun ishchi stolidan uning yorlig'iga sichqonchani olib borib 2 marta bosamiz va XAMPP paketi oynasi xosil bo'ladi.



24.2-rasm. XAMPP paketi oynasi.

Apache va MySQL dasturlari oldidagi «x» tugmalairga bosib ularni o'rnatib, ishga tushiramiz. Buning uchun «x» tugmasini bosgandan so'ng Start tugmasini bosamiz. Bu paket o'rnatilgandan so'ng xampp papkasi kompyuterimizning D diskida xosil bo'ladi. Bu paketning afzalligi shundaki apache va php dasturlarini o'zi sozlab oladi.

Biz apache dasturini o'rnatganimizda Apache papkasi ichida joylashgan conf papkasi ichidagi httpd.conf fayliga o'zgartirishlar kiritishimiz kerak bo'lar edi. Bunda d:\xampp\apache\conf\extra\ papkasi ichida httpd-autoindex.conf fayliga barcha sozlash ishlarini quyidagicha qilib to'g'rilab qo'yadi.

```
<Directory «D:/xampp/apache/icons»>  
Options Indexes MultiViews  
AllowOverride None  
Require all granted  
</Directory>
```

Bu paketda ishlayotgani tekshirish uchun brouzer orqali unig adreslar yoziladigan maydoniga <http://localhost> ni yozib tekshirib ko'ramiz. Agar ishlasa, bundan ko'rinib turibdiki paket muvofaqiyatli o'rnatilgan bo'ladi.

24.2. Denwer dasturi bilan ishlash

Denwer (qisqa D.n.w.r yoki DNWR - Web – dasturchilar uchun Jentelmenlarcha majmua) – Windows sistemasida shaxsiy kompyuter orqali lokal holatda (web-ilovalar, boshqa dinamik tarkib web-sahifalarni) saytlar yaratish va ularni xatolarini (Internetga ulanishlarsiz) topish imkonini beradi. Dastruning asoschilari: Dmitriy Koterov - ham mashhur PHPni o'rgatuvchi qo'llanmalar va ijtimoiy tarmoq “Мой кпыт” asoschisi sifatida tanilgan muallifi, Anton Sushchev – dasturchi, forum moderator, Maykl Livach – foydalanuvchilarning texnik muhandisi, dasturchi.

Tarix. 1999 yilda Dmitriy Koterov “Windows 95/98 uchun Apache + Perl + PHP3 + MySQL o'rnatish uchun qo'llanma” deb nomlangan maqola yozgan. U bergan intervyusida, o'quvchilarining xatlariga javoban, u avtomatlashtirish jarayoni haqida o'ylaganini aytgan. Shunday qilib, Denwer qisqa D.n.w.r yoki DNVR -.Web – dasturchi uchun jentilmenlarning majmuasi ishlab chiqilgan. Avvaliga u faqat zip-arxivda past versiyadagi Apache, Perl, PHP, MySQL va phpMyAdmin + Perl skript bilan hajmi taxminan 1,5 MB va bir vaqtning o'zida bir nechta

virtual kompyuterlar bilan ishlash uchun mo'jallangan edi. Tizimining barcha komponentlar allaqachon qat'i nazar paketi qo'yiladi axborotnamasining, fayl ustida yo'llari bilan faqat ishlash uchun tuzilgan qilingan. 2002 yil 15 mart fayllarni belgilangan joyga ko'chiriladigan va Windowsdagi yuklanuvchi papkasi uchun yorliq qo'shiladigan oddiy o'rnatuvchi yaratilib va u "Denwer" ning birinchi rasmiy versiyasi deb taklif etildi.

Imkoniyatlar. Dastur o'rnatilgandan so'ng to'liqligicha Apache web – serveri bilan ishalashi, lokal kompyuterda ham, saytlar soni cheklanmaganligi, PHPning fayllarini udalyonniy serverdan yuklamasdan, undan samarali ishlar qilish va xatolar bilan ishlash imkoniyati bor. Undan foydalanish uchun Denwerning barcha verisyalarida /denwer (yoki /etc) papkasi joylashgan Run dasturini ishga tushurish kerak. Ishga tushganida barcha projeklarni saqlash uchun virtual disk xosil qilinadi. Denwerning 3-versiyasi fleshka bilan ham ishlash imkoniyatiga ega. Denwer boshqa WAMP-taqsimlash ajratadigan bir xususiyati, hosts avtomatik tuzatish tizimi Fayl bo'lgan katalogda joylashgan papkaga nomi bilan bir xil nom bilan Denwerda ishlayotgan mahalliy saytlarga kirishni beradi. Denwer DNS-server bilan mahalliy analog hisoblanadi.

Denwerning asosiy paketi quyidagicha:

- SSI, SSL, mod_rewrite moduli, mod_php Apache web-server qo'llab-quvvatlaydi;
- GD, MySQL, SQLite dasturlarni PHP interpretatori bilan ishlashini ta'minlaydi;
- MySQL ma'lumotlar bazasi jurnali (mysqld-max);
- Andozalar asosida virtual xostlar boshqaruv tizimi;
- Nazorat tizimi ishga tushirish va o'chirish;
- PhpMyAdmin panel ma'lumotlar bazasini boshqarish uchun;
- Standart kutubxonalarsiz holda Perl interpretatori yadrosi (alohida);
- Emulator sendmail va SMTP serverlar bilan birgalikda PHP, Perl, Parser va boshqalar bilan ishlashi;
- O'rnatuvchi(Installer).

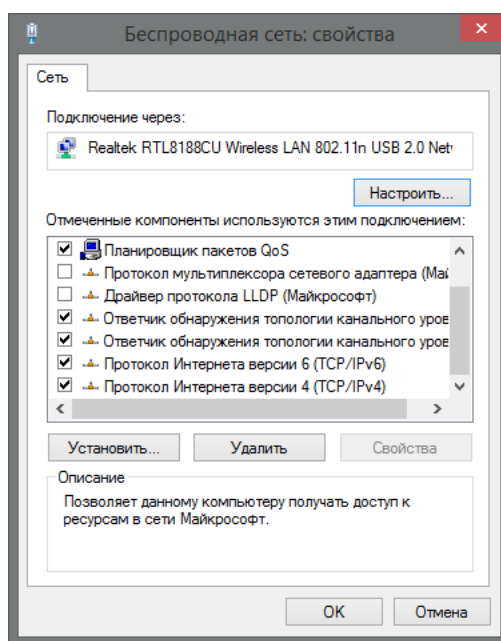
Paketning kengaytirilgan ko'rinishida:

- ActivePerl 5.8 to'liq versiyasi.
- PHP interpretator modul (o'z ichiga olmaydi php_openssl) to'liq majmui bilan 5 versiyasi;
- Python interpretatori;

- DBMS MS SQL, PostgreSQL yoki InterBase / FireBird versiyalar 1 va 2;
- PHP interpretatori versiya 3 yoki 4;
- Tarjimon Parser;
- Konfiguratsiyasi o'zgartirilgan;

Paketi (yopiq .exe bir necha buyruq qatoni yordam dasturlari o'z ichiga oladi) bepul tarqatiladi.

Denwer dasturini local tarmoqlar uchun sozlash. Ko'p web dasturchilar o'zlarining yangi loyihalarini tayyorlanganlaridan so'ng uni foydalanuvchilarga taqdim etishda ba'zi bir muammolar yuzaga keladi. Bu muammolardan biri shuki loyiha uchun Apache, Mysql, PHP server yetishmaslik muammosi. Siz qandaydur korxonaga yoki tashkilot uchun ma'lumotlar saqlaydigan web dastur yaratdingiz va bu dastur butun bir korxonaga bo'yicha ishlatilishi kerak, demak sizda ushbu dasturni barcha foydalanuvchilar uchun taqsimlash muammosi tug'ildi. Agarda siz ushbu dasturni global tarmoqdagi biron bir hostingga joylamoqchi bo'lsangiz bu korxonaga uchun ma'lum bir noqulayliklar tug'dirishi mumkin: hostingga to'lov qilish, har bir kompyuter uchun internet tarmog'i yetkazib berish va bu ish korxonaga rahbari uchun qimmatga tushadi. Muammoni hal etish uchun biz shu korxonadagi biron bir asosiy kompyuterdan foydalanib undan server sifatida foydalanamiz va foydalanuvchilar ushbu serverga LAN yoki WiFi tarmog'i orqali ulana olishadi.



24.3-rasm. Tarmoq bo'limi

Denwer dasturini kerali server kompyuterimizga o'rnatib olamiz va

server ip manzillarini statik manzilga o'rgartiramiz. Buning uchun «Центре управления сетями и общим доступом» dan «Изменение параметров адаптера» bo'limiga kiramiz kerakli adapterni tanlab uning ustiga sichqonchani o'ng tomonini bosamiz va «Свойства» menyusiga kiramiz.

Bu yerdan “Протокол Интернета версии 4 (TCP/IPv4)” bo'limiga kiramiz. “Использовать следующий IP-адрес” qismiga belgi qo'yamiz.

Va ushbu sozlamalarni kiritamiz:

IP-адрес: 192.168.11.1

Маска подсети: 255.255.255.0

Основой шлюз: 192.168.11.1 (Mening internet modemim ip manzili 192.168.11.1 bo'lgani uchun shu manzilni qo'lladim)

“Предпочитаемый DNS-сервер: 192.168.11.1” (Bunga ham internet modem IP manzilini kiritamiz).

Ma'lumotlar kiritilgach “OK” tugmasini bosamiz. Tarmoq adapteriga biz o'zimizga kerakli sozlamalarni kiritdik va keyinchalik boshqa foydalanuvchilar serverga biz kiritgan IP manzil (192.11.1) orqali murojaat qiladi.

Denwer dasturini local tarmoq uchun sozlash. Denwer dasturida hostlar “/home” papkasidagi boshqa papkalar nomiga nisbatan generatsiya qilinadi. Ya'ni agarda biz “/home” papkasiga “mywebsite” deb papka ochib denwerni ishga tushirsak denwer avtomatik tarzda “mywebsite” papkasi nomiga asoslanib yangi “mywebsite” nomi bilan domen ochadi va biz “mywebsite” papkasi ichiga kerakli scriptlarni tashlab brauzer orqali “mywebsite” domeniga kirib scriptlar natijasini ko'ra olamiz. Bu ishni apachi avtomatik tarda qiladi va shu tarda server ip manzilini “127.0.0.1” IP manziliga sozlaydi demak biz generatorni 127.0.0.1 manzili o'rniga tarmoq adapteri uchun kiritgan ip manzilini bo'yicha ishlatadigan qilamiz.

Buning uchun “/usr/local/apache/conf/” papkasidan “httpd.conf” fayliga o'zgartirish kiritamiz. “httpd.conf” faylini bloknout yordamida ochamiz, fayl ichidan “NameVirtualHost \${ip:-127.0.0.1}:\${port:-80}” qismini qidiramiz va ushbu qismni topamiz.

##

НАЧАЛО ШАБЛОНА ВИРТУАЛЬНОГО ХОСТА.

##

Если вы хотите по умолчанию запускать Apache на порту, отличном от 80,

измените номер порта в следующей далее директиве.

```

##
#Listen $&{ip:-127.0.0.1}:$&{port:-80}
#NameVirtualHost $&{ip:-127.0.0.1}:$&{port:-80}
#<VirtualHost $&{ip:-127.0.0.1}:$&{port:-80}>
# DocumentRootMatch «/home/(?!cgi-)(.*)^1/(?!cgi$/cgi-)(.*)»
# DocumentRootMatch «/home/(?!cgi-)(.*)/public_html^1»
# DocumentRootMatch «/home/(?!cgi-)(.*)/public^1»
# DocumentRootMatch «/home/(?!cgi-)(.*)^1/html/(.*)»
# DocumentRootMatch «/home/(?!cgi-)(.*)^1/domains/(?!cgi$/cgi-
)(.*)»
# DocumentRootMatch «/var/www/html/(?!cgi-~)(.*)^1/(?!cgi$/cgi-
)(.*)»
# DocumentRoot «$&»
# ServerName «%&/-www»
# ServerAlias «%&/-www» «%&/-www/www» $&{host:-}
#
# $&{directives:-}
#
# ScriptAlias /cgi/ «$^1/cgi/»
# ScriptAlias /cgi-bin/ «$^1/cgi-bin/»
#</VirtualHost>
Ushbu shablonni yangisiga almashtiramiz.
##
## НАЧАЛО ШАБЛОНА ВИРТУАЛЬНОГО ХОСТА.
##
## Если вы хотите по умолчанию запускать Apache на порту,
отличном от 80,
## измените номер порта в следующей далее директиве.
##
##Listen $&{ip:-127.0.0.1}:$&{port:-80}
#NameVirtualHost $&{ip:-*}:$&{port:-80}
#<VirtualHost $&{ip:-*}:$&{port:-80}>
# DocumentRootMatch «/home/(?!cgi-)(.*)^1/(?!cgi$/cgi-)(.*)»
# DocumentRootMatch «/home/(?!cgi-)(.*)/public_html^1»
# DocumentRootMatch «/home/(?!cgi-)(.*)/public^1»
# DocumentRootMatch «/home/(?!cgi-)(.*)^1/html/(.*)»
# DocumentRootMatch «/home/(?!cgi-)(.*)^1/domains/(?!cgi$/cgi-
)(.*)»
# DocumentRootMatch «/var/www/html/(?!cgi-~)(.*)^1/(?!cgi$/cgi-

```



```

)(.*)»
# DocumentRoot «$&»
# ServerName «%&/-www»
# ServerAlias «%&/-www» «%&/-www/www» $&{host:-}
#
# $&{directives:-}
#
# ScriptAlias /cgi/ «$^1/cgi/»
# ScriptAlias /cgi-bin/ «$^1/cgi-bin/»
#</VirtualHost>

```

Shu fayl ichidan yana bir qismni yangilaymiz. Buning uchun ushbu qismni topamiz.

```

#
# Denwer utilities.
#
Alias /denwer/ /denwer/www/denwer/
Alias /localhost/ /home/localhost/www/
<Location /denwer>
    Deny from all
    Allow from 127.0.0.1
</Location>

```

Va ushbu sozlamalarga o'zgartiramiz.

```

#
# Denwer utilities.
#
Alias /denwer/ /denwer/www/denwer/
Alias /localhost/ /home/localhost/www/
<Location /denwer>
    Allow from all
    Allow from 127.0.0.1
</Location>

```

Denwer dasrugida qayta yuklash “Restart” buyrug’ini beramiz va brauzerimiz orqali tarmoq adapteri uchun kiritgan ip manzilga (192.168.11.1) kiramiz ushbu manzil orqali Denwer daturining localhost papkasidagi ma’lumotlar ko’rinsa hamma sozlamalar bizda to’gri hisoblanadi va “Firewall” sozmalarini o’zgartirsak bo’ldi. “Firewall” uchun biz “80” portiga boshqa tarmoqlardan ulanishga ruxsat berishimiz kerak.

Buning uchun “Панель управлениядан” “Брандмауэр Windows”

bo'limiga kiramiz. Bu yerdan “Дополнительные параметры”ga kirib “Правила для исходящего подключения” qismiga o'tamiz va o'ng tomondagi menyudan “Создать правило”ni bosamiz. Yangi ochilgan oynaga kerakli ma'lumotlarni kiritamiz.

“Тип правила” bo'limidan “Для порта” qismini tanlaymiz va “Далее” tugmasini bosamiz.

“Протокол и порты” bo'limidan “Определенные удаленные порты” qismiga “80” raqamini yozamiz va “Далее” tugmasini bosamiz.

“Действие” bo'limidan “Разрешить подключение” qismiga belgi qo'yamiz va ikki marotaba “Далее” tugmasini bosamiz.

“Имя” bo'limidan “Имя:” qismiga o'zimizga esda qolarni nomni kiritamiz va “Готово” tugamasini bosamiz.

Yana bir marotaba denwer dasturiga qayta yuklash “Restart” buyrug'ini beramiz.

Server sozlamalari ishlashga tayyor.

Local tarmoqdagi boshqa kompyuter sozlamalari. Tarmoqdagi boshqa kompyuterlarni serverga sozlash uchun “c:\Windows\System32\Drivers\etc\” papkasidagi hosts fayliga o'zgartirish kiritamiz.

Ushbu faylni bloknot dasturida ochib eng pastgi qismiga ushbu ma'lumotlarni kiritamiz.

192.168.11.1 mywebsite.

“192.168.11.1” bu yerda serverimizning ip manzili “mywebsite” esa serverdagi bizga kerakli domen nomi. Agarda siz local tarmoq orqali bir necha domenlarga ruxsat bermochi bo'lsangiz yuqoridagi sozlamani yangi qatorlarga yozim “mywebsite” o'rniga domen nomini kiritasiz.

Tarmoq sozlamalari to'liq ishga tayyor tarmoqdagi kompyuterlar brauzeridan kirittgan domen nomi bo'yicha “mywebsite” serverga murojaat qilaverasiz.

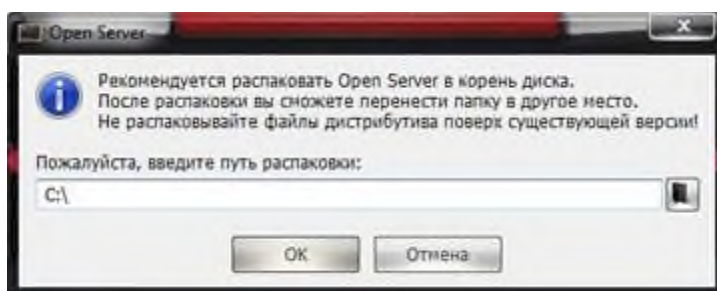
24.3. Open Server ni o'rnatish

Local serverni o'rnatish uchun bizga server texnologiyalari yani Apache, PHP, MySQL kerak bo'ladi. Ularni alohida olib o'rnatib chiqish mumkin, lekin bu bizga o'rtiqcha qiyinchilik va vaqtni oladi. Bularning tayyor yechimi mavjud. Tayyor yechimlar orasida yurtimizda eng ommabopi DENWER hisoblanadi. Lekin DENWER bizning talablarga uncha mos kelmaydi shuning uchun Open Serverdan foydalanamiz. Open Serverda web texnologiyalarning eng so'nggi versiyalari o'rnatilgan

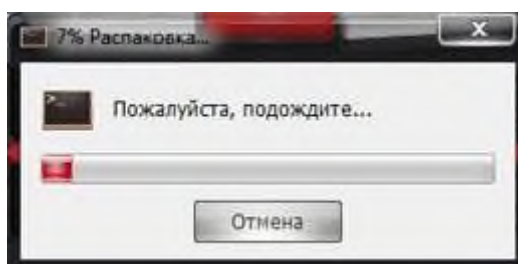
bo'ladi va unda tashqari qo'shimcha imkoniyatlar, kengaytirilgan sozlamalar, qo'shimcha modullar ham mavjud. Masalan Denwerda composer yo'q, Open Serverda esa bor va ishlatish uchun qulay.

Yuklab olganimizdan so'ng uni o'rnatamiz, o'rnatish esa huddi boshqa Windows dasturlar kabi oddiy. Bu sizga qiyinchilik tug'dirmaydi.

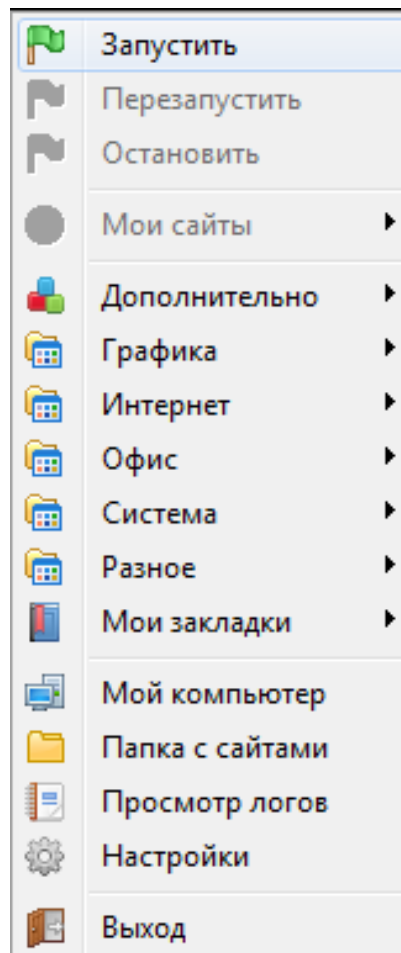
Kompyuterda web sayt yaratish uchun open serverni o'rnatish quyidagicha amalga oshiriladi. Open Server ni C:\ diskga o'rnatish so'raladi bunda siz ixtiyoriy diskni tanlashingiz mumkin Rasm 1. Va Ok tugmasini sichqoncha bilan bosing .



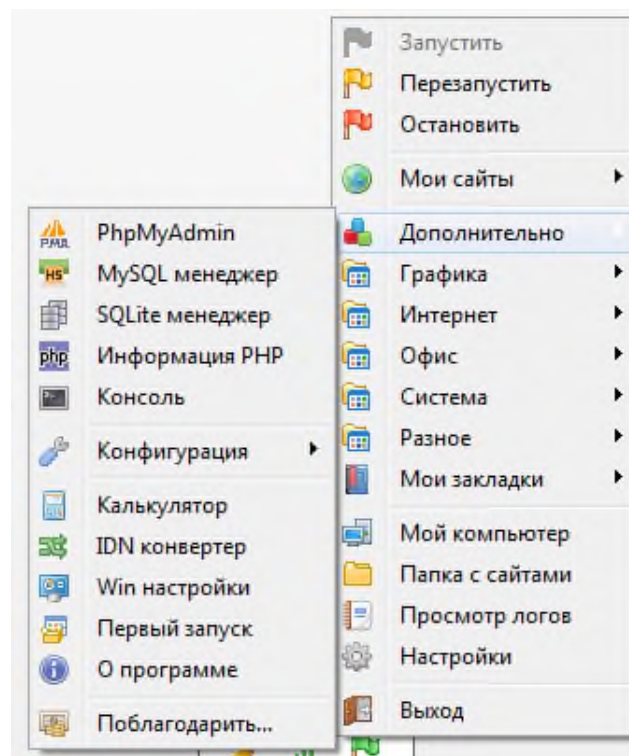
24.4-rasm. Open server o'rnatish.



24.5-rasm. Open server ni o'rnatilish jaroyoni.

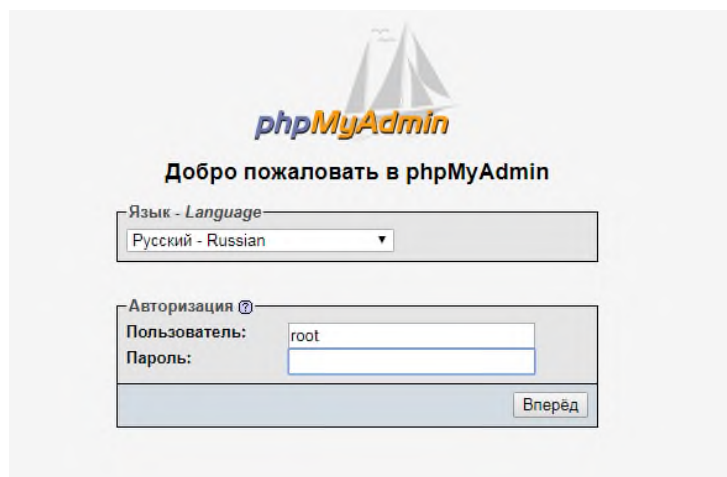


24.6-rasm. Open server ni ishga tushirish jaroyoni.



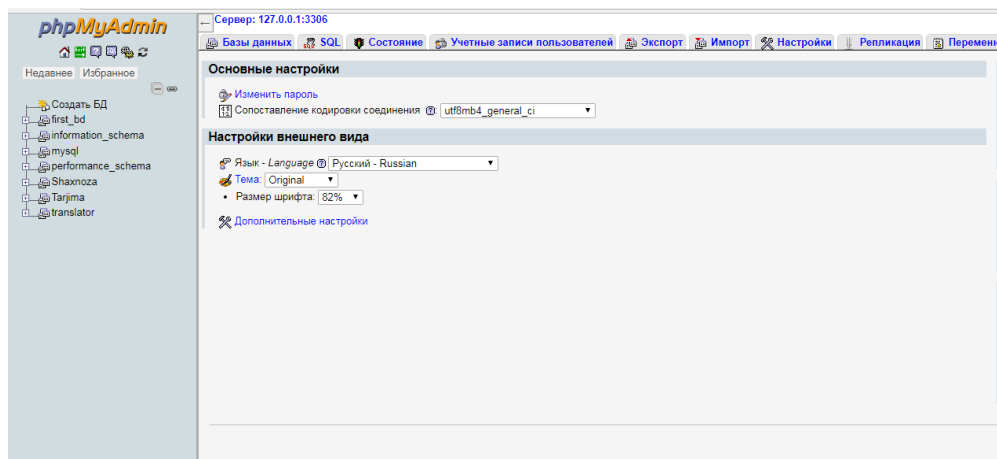
24.7-rasm. Open server da PhpMyAdmin ishga tushirish jaroyoni. PhpMyAdmin ishga tushirish jarayonida login va parol so'raydi.

Login: root
parol:



24.7-rasm. PhpMyAdmin ishga tushirish jarayoni.

Login va Parolni yozilgandan so'ng PhpMyAdmin ning ishchi osnasi ishga tushadi.



24.8-rasm. PhpMyAdmin ning ishchi osnasi.

O'rnatib bo'lganingizdan keyin C diskining ichida OpenServer papkasi paydo boladi. uning ichida:

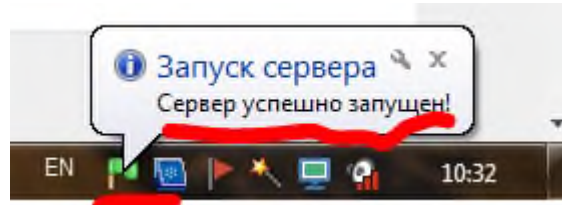
- domains
- modules
- progs
- userdata

papkaları mavud. domains papkasida bizning saytlarimiz joylashadi, modulesda esa local serverimizning modullari yani Apache, PHP, MySQL, MongoDB, MariaDB, cron, sendmail va boshqa saytda ishlatish mumkin bo'lgan modullar bo'r.

Endi C:/OpenServer papkasida Open Server x64 va Open Server x86 dasturlari bor. O'zingizda qanaqa sistema bo'lsa o'sha sistemaga mos dasturga kirasiz. Agar x64 lik operatsion tizim bolsa Open Server x64.

Agar x32 lik operation tizim bo'lsa Open Server x86 dasturiga kirasiz. Keyin masalalar paneli (панель задач)da tillar to'radigan joyda qizil ikonka paydo bo'ladi.

Bu server o'chiq degani. Shu yerga kirib "Запустить"ga kiramiz. Shunda local serverimiz ishga tushadi va qizil bo'lib turgan bayroqcha yashil boladi.

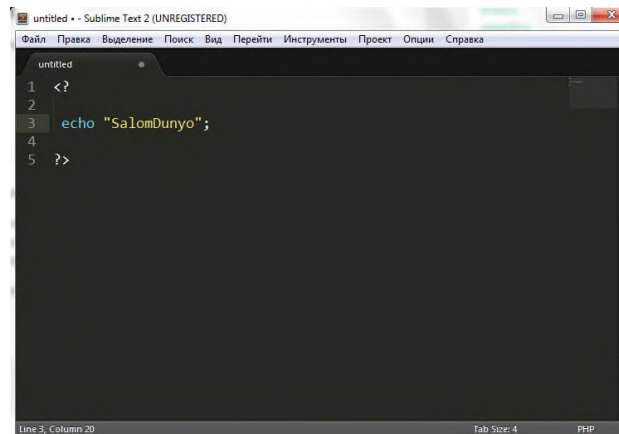


24.9-rasm. Server ishga tushganini bildiruvchi oyna.

Shunda bizga serverimiz ishga tushgani haqida ma'lumot beradi.

Endi esa sayt qo'shish uchun C:/OpenServer/domains/ papkasida saytimizning papkasini ochamiz. Masalan php5.uz deb papka ochsak, serverimizda php5.uz sayti paydo boladi. Saytimizning papkasini yaratib bo'lgandan keyin Open Serverga kirib "Перезапустить"ga kiramiz (har safar yangi sayt qo'shganingizda shunday qilasiz).

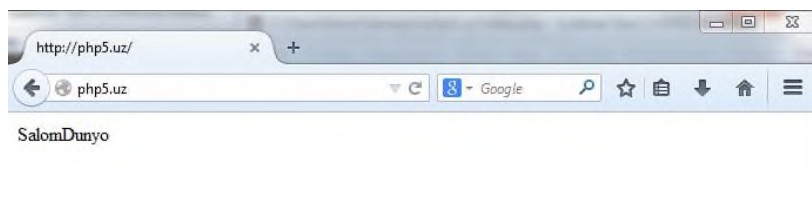
Server qayta ishga tushgandan so'ng browserdan be'malol saytga, yani php5.uz kirsak bo'ladi. Lekin avval bitta ish qilamiz. Open Serverga kiring, unda keyin "Офис" va uning ichida "Редактор Кода"ga kirib quyidagicha kod yozamiz.



24.10-rasm. Kodni tahrirlovchi oyna.

va CTRL+S tugmalarini bosib saqlab olishimiz kerak bo'ladi. Bu faylni php5.uz papkasiga index.php nomi bilan saqlaymiz.

Saqlab bo'lganimizdan keyin esa brauzerdan php5.uz manzilini yozib kiramiz:



24.11-rasm. Brauzerdagi natija.

saytda esa “SalomDunyo” so’zi paydo bo’lsa demak sayt ishlamoqda. Agar xatolik yuzaga kelsa qayta takorlab ko’ring.

NAZORAT SAVOLLARI

1. XAMPP – krossplatformali web – server jamlanmasi va uni o’rnatish?
2. Denwer dasturi bilan ishlash?
3. Open Server va uni o’rnatish?

25-BOB. WEB DASTURLASH USULLARI

25.1. Fayllarni serverga yuklash

Web texnologiyasi olamida fayllar ro'li juda keng. Shuning uchun ham web dasturlar bunga alohida etibor qaratib kelishgan. Shu jumladan Yii ning o'zining qoida va uslublariga ega. Bir kichik misol orqali bu ishlar qanday amalga oshirilishini ko'rsatishga ruxsat bergaysiz. Buning uchun biz misolimizda rasmlarni yuklashga etiborimizni qaratamiz va shu yuklangan rasmni o'qib olish masalasini ham birgalikda ko'rgan bo'lamiz.

Birinchi navbatda biz qanday qilib fayl yuklashni ko'rib chiqamiz buning uchun jadvalimiz hosil qilingan bo'lishi kerak. Chunki ma'lumotlarni qayerdadir saqlashimiz kerak. Demak jadval yaratishdan boshlaymiz.

```
CREATE TABLE 'pictures' ('id' INT(11) NOT NULL
AUTO_INCREMENT, 'pictures' VARCHAR(100) NULL DEFAULT
NULL,
```

```
'description' VARCHAR(100) NULL DEFAULT NULL,
PRIMARY KEY ('id'))
COLLATE='utf8_general_ci'
ENGINE=InnoDB
```

Va shu jadvalga mos model hosil qilamiz.

Hosil bo'lgan modelimizdagi tekshiruv qoidasiga quyidagicha kod qo'shamiz.

```
Class Pictures extends CActiveRecord {
    public function rules(){
        return array(
            //faqatgina rasmlarni yuklashga imkon beradigan fayl uchun
qonunlar o'rnatish!
            Array('pictures', 'file', 'types'=>'jpg, gif, png'),);
        }
} <pre>
```

Keyin controllerimizda quyidagicha kod aniqlaymiz, bu kod bizga faylni yuklab olishimiz va jadvalga ma'lumot qo'shishimiz uchun kerak bo'ladi.

</p>

```
<pre>class SiteController extends Ccontroller {
    public function actionCreate(){
        $model = new Pictures;
```



```

        if(isset($_POST['Pictures']))
        {
            $model->attributes = $_POST['Pictures'];
            $uploadedFile =
CuploadedFile::getInstance($model,'pictures');
            $fileName = $uploadedFile;
            $model->pictures = $fileName;

            if($model->save())
            {
                $model->pictures->saveAs(
Yii::getPathOfAlias('webroot').'/images/uploaded/'.$fileName
                );
            }
        }
        $this->render('create', array('model'=>$model));
    }
}

```

Endi viewimizni hosil qilamiz.

```

<?php echo CHtml::form("", 'post', array('enctype'=>'multipart/form-
data')); ?>
<?php echo CHtml::activeFileField($model, 'pictures'); ?>
<?php echo CHtml::activeTextArea($model, 'description',
array('rows'=>10,'cols'=>25)); ?>
<?php echo CHtml::submitButton('submit'); ?>
<?php echo CHtml::endForm(); ?>

```

Izoh: Fayl yuklayotgan vaqtda faqatgina uning tipini emas balki o'lchamini ham etiborga oladi. Bizning quyidagi misol uchun 1MB dan 313ata bo'lmagan faylni yuklay oladi.

```

Public function rules(){
    return array(
        array('image', 'file', 'types'=>'jpg, gif, png', 'maxSize' =>
1048576),);
    }

```

Validatorlar haqida yanada chuqurroq bilmoqchi bo'lsangiz CfileValidator giper murojaat orqali tanishishingiz mumkin.

Endi faylni yuklab olganimizdan keyin uni ekranga chiqarish masalasini ko'rib chiqamiz. Contrllerimizga quyidagicha kod yozamiz:

```
public function actionView()
{
    // Jadvaldagi barcha ma'lumotlarni olish
    $model = Pictures::model()->findAll();
    $this->render('view', array('model'=>$model));
}
```

Barcha ma'lumotlarni yig'gandan keyin view pagiga jo'natmoqda. View pagimiz quyidagicha ko'rinishga ega bo'ladi:

```
<?php
    foreach ($model as $value)
    {
        echo «<div><img src='/images/uploaded/».$value-
>pictures.»'
        Width='100px' height='100px' style='margin:10px;'/>«;
        echo « «.$value->description.»</div>«;
    }
?>
```

Endi siz View actioniga murojaat qilsangiz barcha yuklangan ma'lumotlariningiz ekranda namoyon bo'ladi.

25.2. PDF hujjatlarni hosil qilish

Biror bir qilayotgan proyektimizda hisobotlarni pdf formatiga o'tkazish kabi ishlarga duch kelamiz. Bunda Yii ning maxsus kengaytmalaridan foydalanish maqsadga muvofiqdir. Shulardan biri tcpdf kengaytmasidir.

Masalaning qo'yilishi:

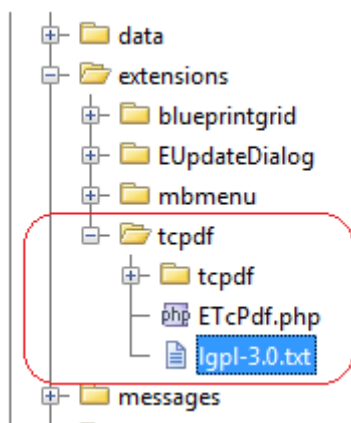
Ma'lumotlar omboridagi employee jadvalini barcha ma'lumotlarini olib uni tartib bo'yicha jadval ko'rinishida pdf formatiga o'tkazish:

Yechimi:

1. <http://www.yiiframework.com/extension/tcpdf> - giper murojaat orqali TCPDF faylini o'zimizga ko'chirib olamiz va /protected/extensions/tcpdf ga joylaymiz

2. <http://www.tcpdf.org/> saytiga kiramiz va ushbu "tcpdf" faylni o'zimizga ko'chiramiz. Keyin/protected/extensions/tcpdf/tcpdf ga joylaymiz.

3. Biz fayllarimizni ushbu ko'rinishda joylayshtirgan bo'lamiz:



4. Endi jadval yaratish kerak bo'ladi. Buning uchun quyidagicha kod orqali ma'lumotlar omboringizda jadval hosil qiling va uni to'ldiring

```
CREATE TABLE 'employee' ('ID' INT(10) UNSIGNED NOT
NULL AUTO_INCREMENT COMMENT 'ID сотрудника',
'firstName' VARCHAR(30) NOT NULL,
'lastName' VARCHAR(40) NOT NULL,
'address' TEXT NULL DEFAULT NULL,
'phoneNumber' TEXT NULL DEFAULT NULL,
PRIMARY KEY ('ID')
)
```

```
COLLATE='utf8_general_ci'
ENGINE=InnoDB;
```

```
INSERT INTO 'employee' ('firstName', 'lastName', 'address',
'phoneNumber') VALUES ('Obidjon', 'Sobirjonov', 'Yakkasaroy
tumani', '123-65-32');
```

```
INSERT INTO 'employee' ('firstName', 'lastName', 'address',
'phoneNumber') VALUES ('Karimbek', 'Sharipov', 'Shayhontohur
tumani', '994-44-31');
```

```
INSERT INTO 'employee' ('firstName', 'lastName', 'address',
'phoneNumber') VALUES ('Foziljon', 'Hakimov', 'Yunusobod tumani',
'333-44-55');
```

Employee jadvali yaratilib ma'lumotlar qo'shilgandan keyin shu jadval uchun Yii ning Gii generatori orqali model yaratamiz.

5. Endi controllerimizda action hosil qilamiz va quyidagicha kod kiritamiz:

```
public function actionPrint()
{
$pdf = Yii::createComponent(
```

```

    'application.extensions.tcpdf.EtcPdf', 'P', 'cm', 'A4', true,
'UTF-8');
$pdf->SetCreator(PDF_CREATOR);
$pdf->SetAuthor(«FirstName»);
$pdf->SetTitle(«Orders»);
$pdf->SetSubject(«Orders»);
$pdf->SetKeywords(«Orders»);
$pdf->setPrintHeader(false);
$pdf->setPrintFooter(false);
$pdf->AddPage();
$pdf->SetFont('freeserif', '', 14);
$tbl = «« . date('d.m.Y', time()) . «»;
$pdf->SetFont('freeserif', '', 10);
$tbl .= '';
$printOrders = null;
$employee = Employee::model()->findAll();
$tbl .= «<table style=\»padding: 5px;\» border=\»1\»>
    <tbody>
        <tr bgcolor=\»#ccc\»>
            <th width=\»30%\»>
                <strong>Employee</strong>
            </th>
            <th width=\»70%\»>
                <strong>Address</strong>
            </th>
        </tr>«;
foreach ($employee as $value)
{
    $tbl .= «<tr><td><strong>FIO:</strong>
        { $value->firstName } – { $value->lastName }
        <strong>Phone Number:</strong>
        { $value->phoneNumber }
    </td>
    <td>
        { $value->address }
    </td></tr>«;
}
$tbl .= «</tbody></table>«;
$pdf->writeHTML($tbl, true, false, false, false, '');

```

```
$pdf->Output('filename.pdf', 'I'); }
```

25.3. Model-View-Controller (Model-Ko'rinish-Nazorat)

MVC tushunchasi (Model-View-Controller: model-ko'rinish-nazoratchi) so'nggi yillarda web-dasturlash dunyosida juda tez-tez esga olinadi.

Web-ilovalarni ishlab chiqish bilan shug'ullanadigan har qanday kishi, bu yoki boshqa usul bilan ushbu qisqartma bilan duch kelgan. Bugun biz MVC kontseptsiyasi nimani anglatishini va nima uchun u ommalashganini aniqlaymiz.

Web ilovalar uchun MVC. MVC dizayn shabloni asosida yotuvchi g'oya juda oddiy: bizning ilovalardagi turli funktsiyalar uchun javobgarlikni aniq ajratish kerak.

Ilova uchta asosiy komponentaga bo'linadi, ularning har biri turli hil vazifalarga javob beradi.

Nazoratchi (Controller). Nazoratchi foydalanuvchi so'rovlarini boshqaradi (Контроллер управляет запросами пользователя (foydalanuvchi turli xil xatti-harakatlarni amalga oshirish uchun interfeys elementlarini bosganda HTTP GET yoki Post so'rovlari sifatida qabul qilinadi). Uning asosiy funktsiyasi - foydalanuvchi tomonidan ko'rsatilgan harakatlarni bajarish uchun zarur bo'lgan resurslar va ob'ektlarning harakatini chaqirish va muvofiqlashtirish. Odatda nazoratchi vazifa uchun mos keladigan modelni chaqiradi va mos bo'lgan ko'rinishni tanlaydi.

Model - bu ma'lumotlar boshqaruvi konsepsiyasini ifodalovchi ma'lumotlar bilan ishlash uchun ishlatiladigan ma'lumotlar va qoidalardir. Har qanday ilovada, butun tuzilma muayyan tarzda qayta ishlangan ma'lumotlar sifatida modellashtiriladi. Ilova uchun foydalanuvchi nima – xabar yoki kitob? Faqat qoidalarga muvofiq qayta ishlanishi kerak bo'lgan ma'lumotlar (sana kelajakni ko'rsatishi mumkin emas, e-mail ma'lum bir formatda bo'lishi kerak, ism X belgilardan uzun bo'lmasligi kerak).

Model, foydalanuvchi so'ragan ma'lumotlarni tekshiruvchiga taqdim etadi (xabar, kitob sahifasi, fotoalbom va shu kabilar). Ma'lumot modeli, biz ularni foydalanuvchi uchun qanday qilib taqdim etishimizga bog'liq bo'lmagan holda, bir xil bo'ladi. Shuning uchun biz ma'lumotlarni ko'rsatish uchun mavjud ko'rinishni tanlaymiz.

Ushbu model bizning murojaatimizning mantig'ining eng muhim qismini, biz bajaradigan muammoni hal qiladigan mantiqni o'z ichiga

oladi (forum, magazin, bank, va shunga o'xshashlar). Nazorat qiluvchi asosan ilova uchun tashkiliy mantiqni o'z ichiga oladi.

Ko'rinish (View). Ko'rinish modeldan olingan ma'lumotlarni taqdim etishning turli usullarini taqdim etadi. U ma'lumotlar bilan to'ldiriladigan shablon bo'lishi mumkin. Turli xil ko'rinishlar bo'lishi mumkin va nazoratchi hozirgi holatga mos bo'lgan yahshisini tanlashga qaror qiladi.

Web ilovalar odatda nazoratchi, model va korinish to'plamlaridan tashkil topgan. Nazoratchi asosiy sifatida o'rnatilishi mumkin, bu barcha so'rovlarni oladi va vaziyatga qarab boshqa harakatlarni amalga oshirish uchun boshqa nazorat vositalarini chaqiradi.

Afzalliklari. MVC kontseptsiyasidan foydalanishdan olingan eng katta afzallik, taqdimot mantig'ini (foydalanuvchi interfeysi) va dastur mantig'ini aniq ajratishdir.

Turli xil turdagi qurilmalardan foydalanadigan turli foydalanuvchilarni qo'llab-quvvatlash bugungu kunda keng tarqalgan muammodir. Taqdim qilinayotgan interfeys so'rov shaxsiy kompyuterdan yoki mobil telefondan kelayotganini farqlay olishi kerak. Model bir xil ma'lumotlarni qaytaradi, yagona farqli tomoni shundaki, ma'lumotlarni chiqarish uchun nazoratchi turli ko'rinishlarni tanlaydi.

Ilova mantig'idan fikrlarni ajratishdan tashqari, MVC kontseptsiyasi ham katta ilovalarning murakkabligini sezilarli darajada kamaytiradi. Kod juda ham tuzilmaviy bo'lib chiqadi va shuning uchun qo'llab-quvvatlash, sinov va echimlarni qayta ishlatish osonlashadi.

Nima uchun ishchi muhitini ishlatish kerak? Qachonki siz ishchi muhitdan foydalanganingizda, MVC bazaviy tarkibi tayyor bo'ladi va sizga MVC shabloniga mos kelishi uchun mos direktoriyalarga fayllaringizni joylagan holda faqatgina tuzilmani kengaytirishingiz qoladi. Bundan tashqari, sizda yahshi testlangan va deyarli yozib bo'lingan funksiyalar to'plami mavjud bo'ladi.

MVC ishchi muhitiga misol sifatida cakePHP ni ko'rib chiqamiz. O'rnatib bo'lingandan so'ng sizda uchta asosiy direktoriya paydo bo'ladi:

- app/
- cake/
- vendors/

app papka fayllaringiz joylashgan joy hisoblanadi. Bu joy ilovalaringizni amalga oshirish uchun.

cake papkada cakePHP fayllari joylashadi (ishchi muhit funksionalligi).

vendors ishlab chiqaruvchilar tomonidan PHP kutubxonalarni saqlash uchun xizmat qiladi.

Sizning ish joyingiz (ilova katalogi) quyidagi tuzilishga ega:

- app/
- config/
- controllers/
- locale/
- models/
- plugins/
- tests/
- tmp/
- vendors/
- views/
- webroot/

Siz controllers direktoriyasiga nazoratlaringizni, models direktoriyasiga modellarni va views direktoriyasiga ko'rinishlarni joylashingiz kerak.

Siz ishchi muhitdan foydalanishni boshlashingiz bilan, yaratish yoki o'zgartirish kerak bo'lgan ilovangizning ixtiyoriy qismi qayerda joylashishi darhol ma'lum bo'ladi. Mazkur tashkilot o'z-o'zidan dasturni ishlab chiqish va qo'llab-quvvatlash jarayonini sezilarli darajada osonlashtiradi.

NAZORAT SAVOLLARI

1. Fayllarni serverga yuklash?
2. PDF hujjatlarni hosil qilish?
3. Model-View-Controller va u bilan ishlash?

26-BOB. SAYTNI RASKRUTKA QILISH. COAP, GOOGLE ANALITIKSDAN FOYDALANISH

26.1. DNS xizmat

DHPC (Dynamic Host Configuration Protocol) tarmoq protokoli bo'lib, u klientlarni dinamik IP adreslar bilan ta'minlab berish uchun qo'llaniladi.

DHCP ni o'rnatish deyarli oson. Buning uchun dhcp-server paketini o'rnatish kerak. Dhcp-serverni o'rnatishning 2 ta usuli mavjud.

1. Grafik muhitida.
2. Konsol muhitida.

DHCP serverini rostdash linux fayl katalogi /etc/dhcpd.conf faylini o'zgartirishdan iborat.

Mandriva Linux Corporate Server 4.0 da tarmoq xizmatlarini o'rnatish uchun quyidagi oynadan "Networking Services" ni tanlaymiz.

"Advanced Options" ga kirib o'zimizga kerak tarmoq xizmatlarini tanlaymiz. Bu yerda DHCP server, DNS server, NTP server va PXE serverini tanlashimiz mumkin. Jimlik holatida ularning hech tanlanmaydi. Tanlab bo'lingandan keyin "Proceed to installation" tanlanadi.



26.1-rasm. Tarmoq xizmati DHCP ni tanlash.

Tarmoq xizmatlari o'rnatilgandan keyin ularni sozlash uchun Mandriva Linux boshqarish markazi (MCC)ga kiriladi. Undan Network & Internet bo'limiga kiramiz.

Bu yerdan yangi tarmoq interfeysini o'rnatish bo'limiga kiramiz. O'zimizga qulay bog'lanish turini tanlaymiz. Ro'yhatda quyidagilar

keltirilgan:

- Lan connection: mahalliy tarmoq bog'lanishi
- Wireless connection: simsiz tarmoq bog'lanishi
- ADSL connection: ADSL bog'lanishi
- Cable connection: kabelli bog'lanish
- ISDN connection: ISDN bog'lanish
- Modem connection: modemli bog'lanish
- DVB connection: DVB li bog'lanish

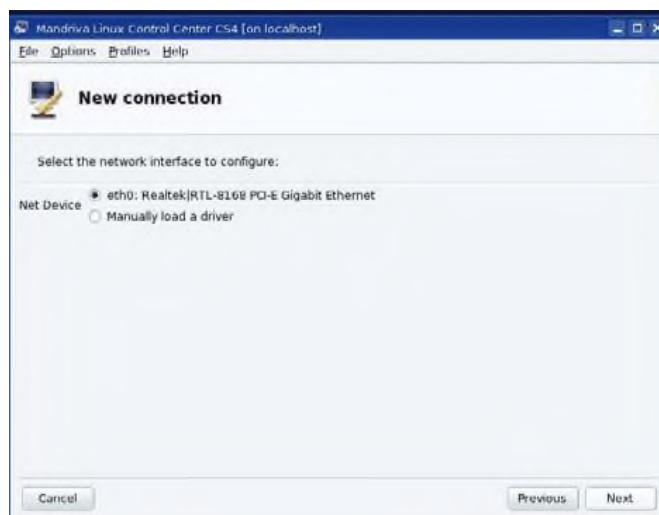
Keyingi bosqichda konfiguratsiya qilinishi kerak bo'lgan tarmoq interfeysi (platasi) tanlanadi va boshlang'ich sozlashlar amalga oshiriladi.



26.2-rasm. Network & Internet bo'limi.



26.3-rasm. Bog'lanish turlari.



26.4-rasm. Tarmoq interfeysi drayverini tanlash.



26.5-rasm. IP protokolni tanlash.

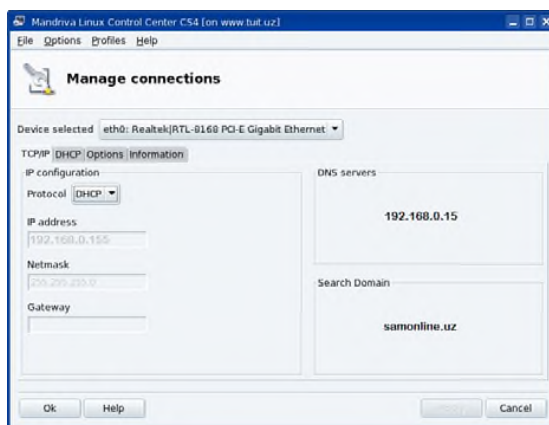
Quyidagi protokollar lokal tarmoqni sozlash uchun ishlatiladi. O'zimiz hohlagan bittasini tanlaymiz. Agar "Automatic IP(BOOTP/DHCP)" tanlansa IP adres server tomonidan dinamik holda beriladi. "Manual configuration" da IP adresni o'zlarimiz beramiz.

Keyingi bosqichda DHCP host nomi beriladi. Agar "Start at boot" ga belgi qo'yilsa kompyuter yuklanganda serverdan dinamik adresni oladi.

Tarmoq va Internet konfiguratsiysi muaffaqiytli tugatilganligi haqida xabar ekranga chiqadi.

Tarmoq rostlangandan keyin Internetga bog'lanish to'g'irlanadi. U 2 qismdan iborat:

- Internet access** da bog'lanish turi va statusi aks etadi.
- Parameters** da host kompyuter nomi, birinchi, ikkinchi, uchunchi DNS server nomi yoki IP adresi kiritiladi.



26.6-rasm. Internetga ulanishni sozlash.



26.7-rasm. Ulanishlarni boshqarish.

Quyida bog'lanishni boshqarish keltirilgan. Bu erda tarmoq haqidagi ma'lumotlar keltirilgan. Ularni o'zgartirishimiz mumkin.

Quyida tarmoqdagi bog'lanishlarni kuzatish oynasi keltirilgan. Bu erda yuborilgan va qabul qilingan paketlar miqdori, ulanish vaqti, tarmoqdagi ma'lumot almashish tezligi keltirilgan.

DNS serverini sozlash **DNS** serveri IP adresni kompyuter nomiga bog'lashga va uni teskarisini amalga oshirishga ruhsat beradi. Masalan: www.mandriva.com nomi 112.55.145.228 adresiga birlashtirilgan.

Mandriva Linux Corporate Server 5 da DNS zonasi va DHCP konfiguratsiysi OpenLDAP katalogida joylashgan.



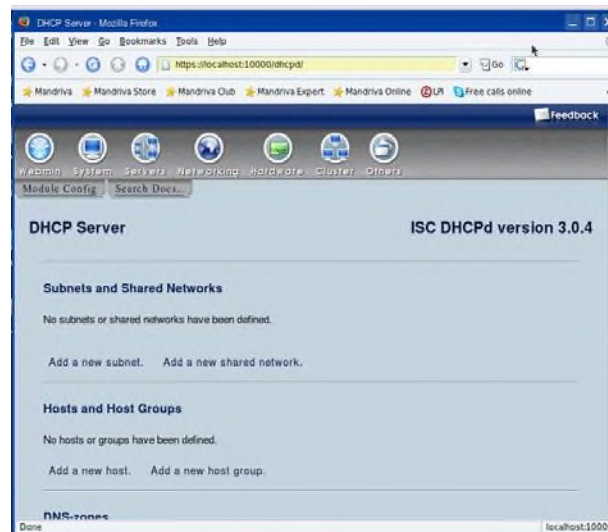
26.8-rasm. DNS serverini boshqarish.

Yangi DNS zonasini qoʻshish uchun “Add a DNS zone” tugmasini tanlab, Maydonlarni toʻldirib chiqing.

Webmin dasturi Webmin dasturidan ham kerakli serverning sozlashlarini amalga oshirish mumkin. Quyidagi muloqot oynada **DHCP** serverini konfiguratsiyasini oʻzgartirish koʻrsatilgan, buning uchun **DHCP** server bandi tanlanadi.



26.9-rasm. Webmin dasturi.



26.10-rasm. DHCP ni sozlash.

DHCP server bandi tanlangandan soʻng bu erda yangi Subnet, host va yangi guruhlar qoʻshish imkoniyti hosil qilingan. Bu muloqot oynada tarmoq adresi, server nomi, server yuklanuvchisi kabi maydonlar toʻldirilgandan soʻng create tugmasi bosiladi va yangi Subnet hosil qilinadi.

26.2. Hosting xizmat xillari

Hosting (*ingl.* »web-hosting») - Internet hosting xizmatining bir turi bo'lib, jismoniy va yuridik shaxslarga o'zlarining web - sahifalarini internetga uzluksiz ulangan web-serverlarga joylashtirish xizmatini nazarda tutadi.

Saytni Internetga joylashtirish uchun birinchi navbatda unga joy kerak. Web-sahifalarni Internet provayderi serverida joylashtirish va qo'llab-quvvatlash hosting deb ataladi. Joy beruvchi kompaniya esa hosting deyiladi. Albatta «tekinga mushuk oftobga chiqmasligi» aniq, lekin bepul xizmat ko'rsatuvchi hostinglar ham ko'plab topiladi (ular chet davlat saytlari). Ammo ular pullik hostinglarchalik xizmat ko'rsata olmaydi.

Bepul hosting. Bepul hostingni yaxshi tomonini shundoq ham bilib olgan bo'lsangiz kerak. Uning yagona qulayligi - bepulligi xolos! Bepul hostingni yomon tomonlari:

1) O'z mijozni saytiga joy berishda o'z reklamasini ham tiqishtiradi. «Sizdan ugina, bizdan bugina» degandek, siz o'z saytingizni bepul hostingga joylashtirasiz va ularning reklamasini o'z saytingizda «namoyish» etasiz. Internetda (chatda emas!) kamida 6 oylik tajribaga ega foydalanuvchi bepul hostinglarni «o'ta yoqimli» reklamalaridan osongina ajrata oladi. www.nm.ru, www.by.ru, www.boom.ru, www.narod.ru (o'zbek saytlaridan eslay olmadim) saytlari sizga hosting xizmatini bilan birgalikda, o'z reklamasini ham «taqdim etadi». Reklama chiqarmaydigan bepul hostinglar ham mavjud. Ularga www.50webs.com, www.xlphp.net, www.awardspace.com larni misol qilsa bo'ladi. Yuqorida sanab o'tilgan hostinglar xizmatidan foydalanadigan bo'lsangiz, sizning saytingiz ikkinchi darajali domenda bo'ladi. (Bu haqda batafsil pastroqda tanishamiz).

2) Dasturlash tillarini qo'llab quvvatlamaydi. Ko'pgina bepul hostinglar dasturlash tillari - PHP, CGI, ASP, MySQL - xullas birortasiniyam qo'llab-quvvatlamaydi. Siz o'z saytingizni HTML va Javascriptda yaratgan bo'lsangiz, bu band sizga taalluqli emas (Ammo dasturlash tillarisiz qanday qilib doimiy ishlaydigan sayt yaratmoqchisiz?).

3) Saytingiz ishlash tezligi past bo'ladi. Bepul xizmat ko'rsatuvchi hostinglar server-kompyuterining ishlash tezligi uncha yuqori bo'lmaydi. Buning ustiga sayt AQShda yoki Yevropaning qaysi bir burchagida turgan server kompyuterda joylashgan bo'lsa ochilishi ham sekinlashadi.

Pullik hosting. Pullik hostingni tanlayotganda quyidagilarga e'tibor qaratish lozim bo'ladi:

- 1) disk makoni;
- 2) Internet kanalining o'tkazish qobiliyati (kengligi);
- 3) fayllarni boshqarish usullari: web-forma yoki FTP bayonnomasi orqali erkin foydalanish;
- 4) standart skriptlar to'plami;
- 5) server tomonida dasturlashtirish mumkinligi (SSI, PHP, ASPlarni qo'llab-quvvatlash, cgi-bin katalogi);
- 6) serverda ma'lumotlar bazalaridan erkin foydalanish – o'z ma'lumotlar bazalarini yaratish va ishlatish mumkinligi;
- 7) shell erkin foydalanish;
- 8) htaccess fayli orqali serverni konfiguratsiya qilish mumkinligi;
- 9) log fayllardan erkin foydalanish;
- 10) uchinchi darajali domenlarni taqdim etish (name.you_name.ru, name1.you_name.ru va boshqa turdagi manzil);
- 11) bir yoki bir necha pochta qutisini qo'llab-quvvatlash;
- 12) uzluksiz elektr energiyasi bilan ta'minlash.

Pullik hosting kompaniyalar O'zbekistonda asta-sekin ko'payib bormoqda va raqobatga qarab narxlari arzonlashmoqda. Ammo hostinglarning tariflari bir-biridan sezilarli darajada farqlanadi.

Misol uchun, ko'zga ko'ringan hostinglar - Arsenal-D va Sarkor kompaniyalarining host-tariflarini solishtirib ko'ramiz:

Arsenal-D: 50 Mb uchun oyiga 3000 so'm sarflanadi. 100 Mbgacha 6000 va 200 Mbgacha 12000 so'm.

Sarkor esa ikki baravar qimmat: 50 Mb uchun oyiga 7000 so'm, 100 Mbgacha 12500 va 500 Mbgacha 41000 so'mni sarflashni talab qiladi. Bir narsani tan olish kerak, ko'p miqdorda joy ajratuvchi hostinglar kam. Joy beruvchilar asosan 20 Mb dan 500 Mb gacha bo'lgan oraliqni ma'qul ko'rishmoqda. Shu sababli ham biror portal-sayt ochmoqchi bo'lsangiz, o'z kompyuteringizni Internetga to'g'ridan-to'g'ri ulab server sifatida ishlatganingiz maqsadga muvofiq.

Pullik hostinglar barchasi hech bo'lmaganda PHP va CGI ni qo'llab-quvvatlaydi.

Va albatta ishlash tezligining yuqoriligi. Agar o'zbek hostinglari xizmatidan foydalanadigan bo'lsangiz, sayt ochilishi ham nisbatan tezroq bo'ladi.

26.3. Web analitika. Google Analytics dan foydalanish

Google Analytics sayt yoki blogingizni oson va tushunarli usulda tahlil qilib beradi.

Masalan: tashrif buyuruvchilar qayerdan kirayapdi. Sayt/blogda nimalar bilan mashg'ul.

Buni sayt yoki blog egasiga qanday foydasi bor?

Foydasi shundaki, siz saytingiz yoki blogingizdagi qatnovni (odamlarning kelishi va ketishi) kuzatib turasiz. Qatnovni kuzatdingizmi, nimalarga ko'proq qiziqishyapti, nimalarda ko'proq ushlanib qolishyapti bilasiz. Nimalarga qiziqish borligini bildingizmi, uni kuchaytirishni, rivojlantirishni boshlaysiz.

Xullas, Google Analytics saytni rivojlantirishda ishlatiladigan servis va sayt yoki blogning rivojlanib borishini ko'rish uchun bundan foydalanish juda muhim.

Qisqacha aytsam, bu orqali saytingiz statistikalarini ko'rib borasiz.

Google Analyticsni o'rnatish butunlay oson ish va buni qadamma-qadam ko'rib chiqamiz.

Bu ish ikkita alohida qismlarga bo'linadi.

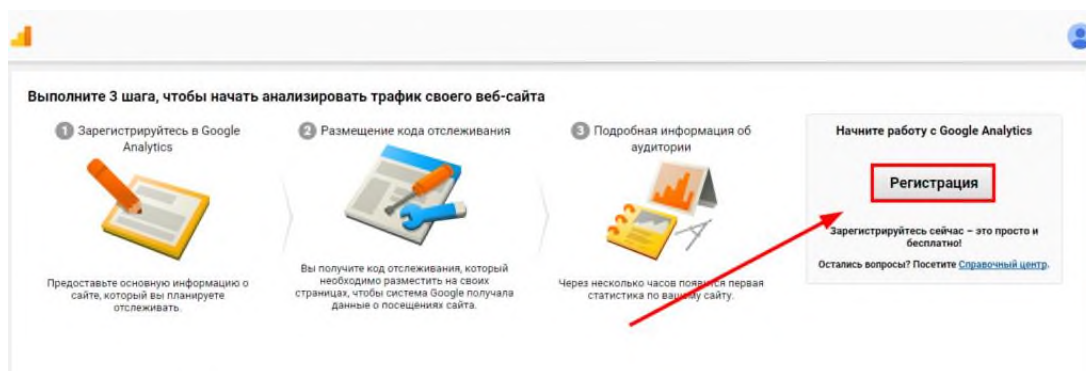
1-Qism. Google Analytics akkauntingizni yaratasiz va saytingizni shu Google Analyticsga ulaysiz. Bularni barchasi Google Analyticsning ichida sodir bo'ladi. Sayt bilan aloqaga o'tishga hojat yo'q.

2-Qism. Siz Google Analytics beradigan kuzatuv kodini saytingizga o'rnatib qo'yishingiz kerak.

1-qism. Google Analytics akkauntini ochish. Bu uchun, oddiy Google akkauntingiz bo'lishi kerak. agar yo'q bo'lsa, unda mana sizga ssilkayangi akkaunt ochib olishingiz mumkin.

Agar siz haligacha Google akkauntingizga kirmagan bo'lsangiz, Google sizdan parol loqiningizni yozib akkauntingizga kirishni so'raydi.

Google login va parolingizni yozganingizdan so'ng, quyidagi rasmda ko'rsatilgan sahifaga tushasiz. Bu yerdan "Регистрация" tugmasiga bosasiz.



26.11-rasm. Google Analytics ga kirish.

Sayingiz haqidagi malumotlarni kiritishingiz kerak bo'ladi ya'ni akkauntingizni sozlash uchun kiritadigan narsalaringiz. Bular:

Akkaunt nomi – bu shunchaki akkauntingizning ichki nomi. Xohlagan narsangizni kiritishingiz mumkin. Masalalan men “UZINFOBIZ” qo'ydim.

Sayt nomi – bu ham ichki boshqaruv nom. Oddiy so'zlar bilan sayingizning nomini kiriting. Masalan: “Uzinfobiz”.

Web-sayt URL – Bu yerda sayingizning url adresini kiritasiz yani manabunday: `http://sizning sayt.uz/` Masalan, meniki: `http://uzinfobiz.ru/`

Sanoat – Bu ixtiyoriy. Bu yerni bo'sh qoldirishingiz mumkin. Bu muhim emas.

Hisobotning vaqt oralig'i – Ko'pincha siz o'zingizning zonangizga qarab vaqtni tanlashingiz mumkin. Lekin, men sizga foydalanuvchilaringiz ko'proq kirayotgan hudud vaqtini tanlashingizni maslahat beraman. Siz to'g'ri vaqtni tanlash huquqiga egasiz.

Настройка аккаунта

Название аккаунта обязательно
В аккаунтах может быть более одного идентификатора отслеживания.

▶ Название нового аккаунта **1**

Настройка ресурса

Название сайта обязательно

▶ Мой новый веб-сайт **2**

URL веб-сайта обязательно

▶ `http://` ▶ Пример: `http://www.mywebsite.com` **3**

Отрасль ?

▶ Выберите один вариант **4**

Часовой пояс отчетов

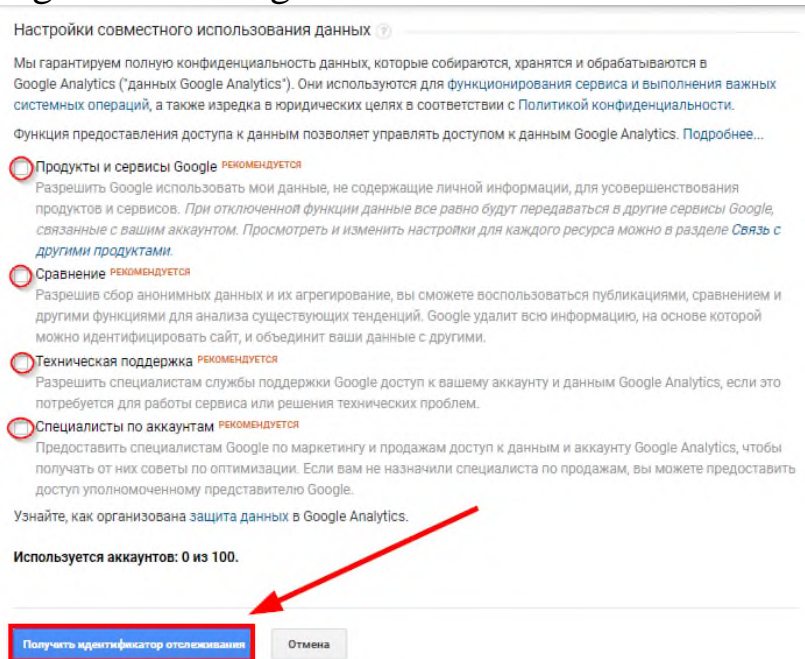
▶ Соединенные Штаты ▶ (GMT-08:00) Тихоокеанское время **5**

Настройки совместного использования данных ?

26.12-rasm. Akkauntni sozlash.

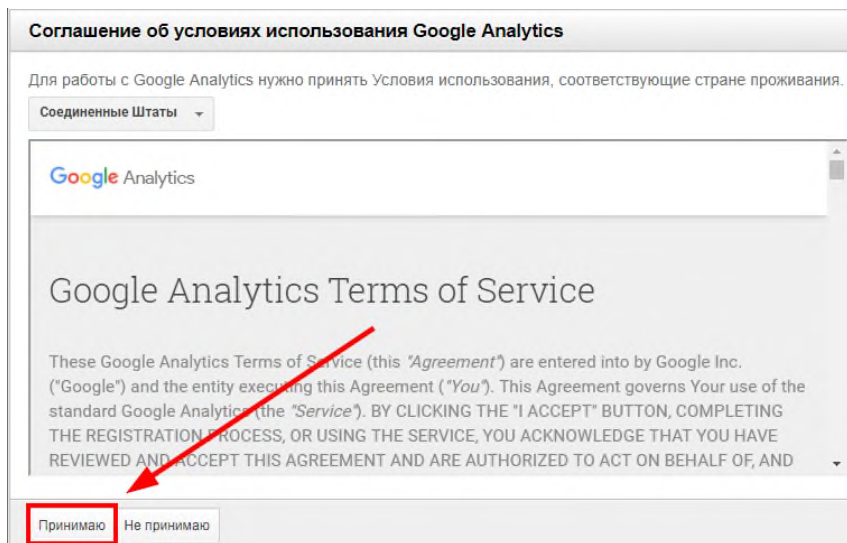
Bularni to'ldirgandan keyin, "Настройки совместного использования данных" ga tushib o'zingizga kerak bo'lgan tanlovlarni (xizmatlarni tanlang)! Ular qizil ramka bilan ajratib ko'rsatilgan. Men ularni bo'sh qo'yib ketdim.

Strelka bilan ko'rsatilgan "Получить идентификатор отслеживания" tugmasini bosing!



26.13-rasm. Xizmatlarni tanlash.

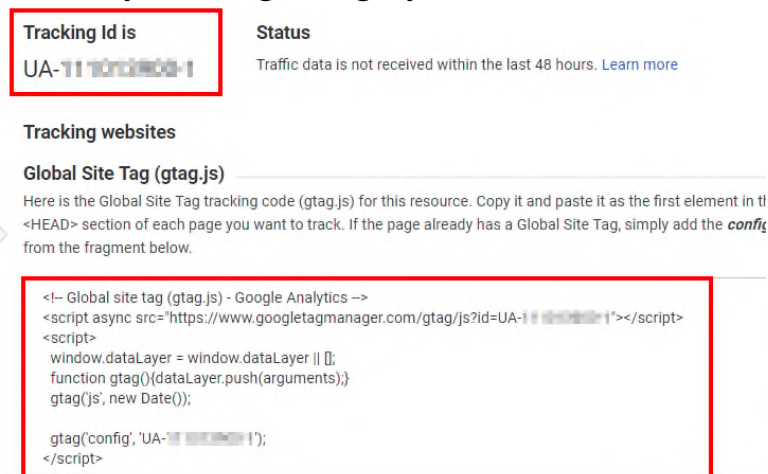
Xizmat ko'rsatish shartlarini qabul qilish uchun "Принимаю" ni bosing!



26.14-rasm. Google Analyticsdan foydalanish shartlarini qabul qilish.

Xizmat ko'rsatish shartlarini qabul qilganingizdan keyin, Google

sizni to'g'ri Google Analyticsning ichiga yuboradi.



The screenshot shows the Google Analytics interface. At the top, there is a table with two columns: 'Tracking Id is' and 'Status'. The 'Tracking Id is' column contains the value 'UA-111111111-1', which is highlighted with a red box. The 'Status' column contains the text 'Traffic data is not received within the last 48 hours. Learn more'. Below this table, there is a section titled 'Tracking websites'. Underneath, there is a section for 'Global Site Tag (gtag.js)'. It contains the text: 'Here is the Global Site Tag tracking code (gtag.js) for this resource. Copy it and paste it as the first element in the <HEAD> section of each page you want to track. If the page already has a Global Site Tag, simply add the *config* from the fragment below.' Below this text, there is a code block containing the following JavaScript code, which is also highlighted with a red box:

```
<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=UA-111111111-1"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());

  gtag('config', 'UA-111111111-1');
</script>
```

26.15-rasm. Google Analyticsning ichiga kirish.

Bu yerda kuzatuv ID va kuzatuv kodini topasiz.

Bu ma'lumotlarni hozircha saqlab qo'ying, chunki bu ma'lumotlar 330eying qismda kerak bo'ladi.

2-Qism. WordPressga kuzatuv kodi qanday o'rnatiladi?

Google Analytics kuzatuv kodini WordPress saytga o'rnatish uchun ikkita yo'lingiz bor, har ikkisinin ham ijobiy va salbiy tomonlari bor:

PLAGINLARSIZ VA PLAGINLAR ORQALI.

Plaginsiz amalga oshirishni istasangiz, bu yerdan ko'rsatmani o'qib oling!

Plaginlar orqali amalga oshirishi xohlasangiz, o'qishda davom eting! Buni ikki xil shaklda qilsangiz bo'ladi.

Siz Insert Header and Footer plaginini o'rnatishingiz bo'ladi. Bu tez usul. Bu plagin Google Analyticsga cheklanmagan. Agar siz qachondir header va footer fayllariga kod qo'shmoqchi bo'lsangiz ham, bu plagindan foydalansangiz bo'ladi.

Yoki siz Google Analytics Dashboard for WP (GADWP) plaginini tanlasangiz ham bo'ladi. Bu plagin orqali kuzatuv kodini o'rnatishingiz ham bo'ladi va WordPress boshqaruv paneli ichida Google Analytics hisoblarini ko'rsangiz ham bo'ladi.

Shuni biling: GADWP bu ishni qiladigan yolg'iz plagin emas bu vazifani yana MonsterInsight plagini ham amalga oshiradi.

Eng yaxshi tanlov siz hisobotlarni WordPress boshqaruv panelida ko'rishni xohlash, xohlamasligingizga qarab yuzaga keladi.

Ba'zilar WordPress boshqaruv paneli ichida saytning statusini ko'rishni foydali deb topishadi. Boshqalar esa boshqaruv paneli ichida hisoblar cheklangan deb topishadi.

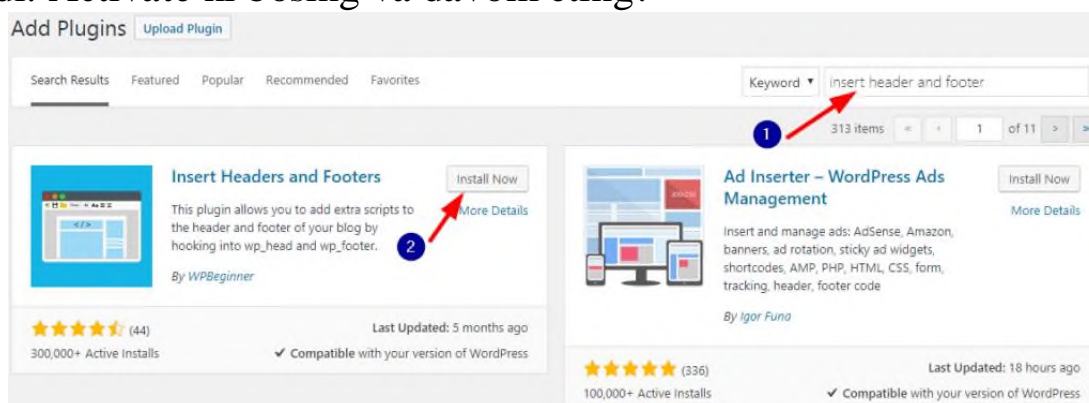
Agar tanlashda ikkilansangiz, boshqaruv paneli ichida hisob beradigan plaginni oʻrnating. Keyinchalik bu yaxshi tuyulmasa, har doim uni oʻchirib yuborishingiz mumkin.

Insert Header and Footer bilan kuzatuv kodi qanday qoʻshiladi.

Demak, Boshqaruv paneli ichida Analyticni koʻrishni xohlamaysiz. Bu degani endi Insert Header and Footer plaginini oʻrnatasiz.

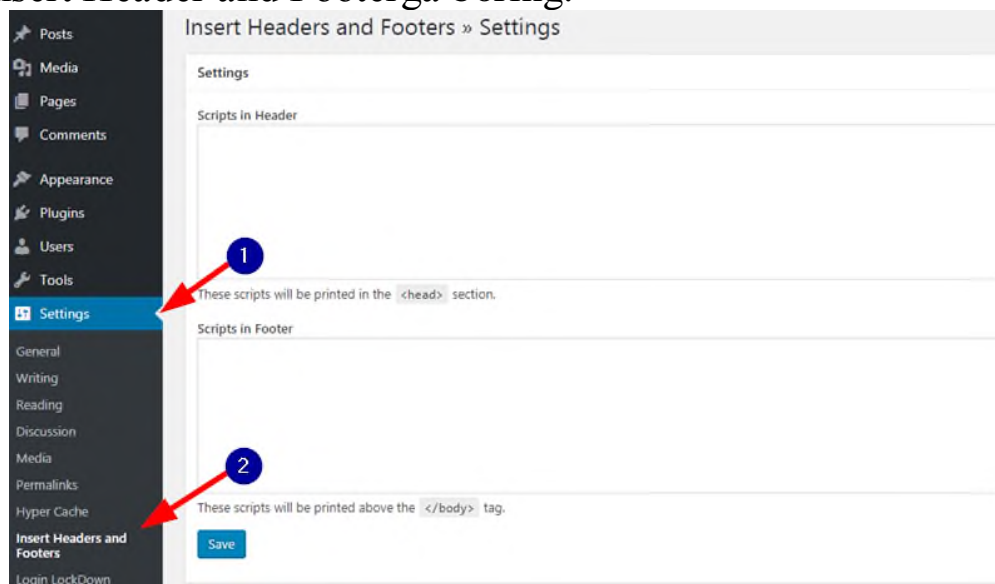
WordPress Dashboard (boshqaruv paneli) ga kiring. Oldin Plugins ni keyin Add New ni bosing! Keyin Insert Header and Footer ni qidiring.

Plaginni topasiz, Install Now tugmasini bosasiz, u esa Activate ga oʻzgaradi. Activate ni bosing va davom eting!



26.16-rasm. Plaginnlarni qoʻshish.

Plaginni aktivatsiya qilganingizdan soʻng, boshqaruv panelida SettingsàInsert Header and Footerga boring.

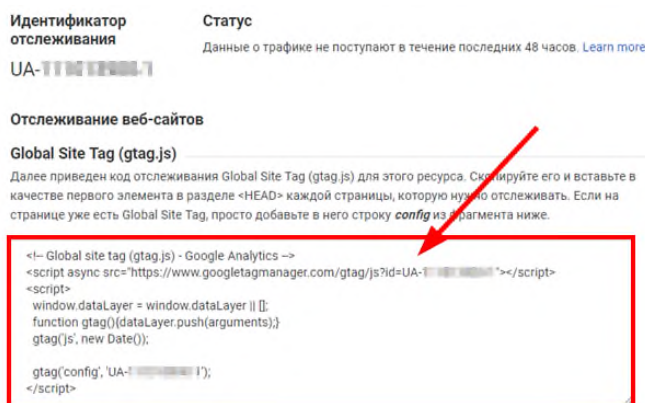


26.17-rasm. Boshqaruv paneli.

Agar esingizda boʻlsa sizga kuzatuv kodini saqlab qoʻyishni aytgandim, foydalanish vaqti keldi.

Google Analyticga qayting va “Отслеживание веб-сайтов” dagi

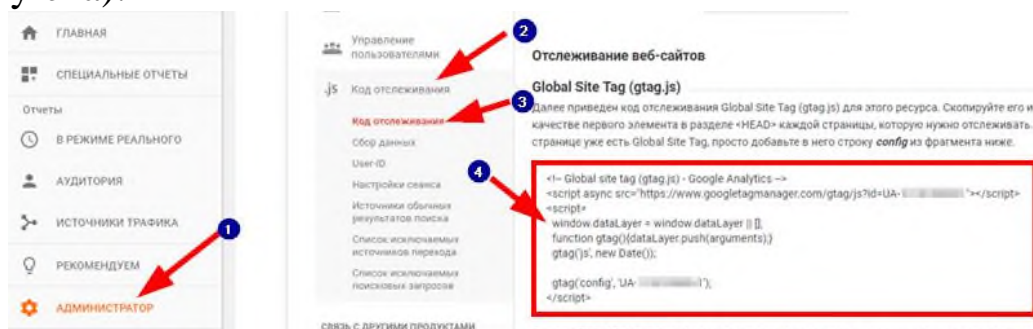
butun scriptni nusxa (kopiya) qilib oling.



26.18-rasm. Google Analyticsga qaytish.

Agar bu tabni behosdan yopib qo'ygan bo'lsangiz ham, qayta topa olasiz.

Bu uchun quyidagi rasmda ko'rsatilgandek ish tuting (sonlar ketma-ketligi bo'yicha).



26.19-rasm. Google Analyticsni qayta ko'rish.

Butun kodni nusxa qilganingizdan keyin WordPress Dashboardga qayting va kodni Scripts in Header ga o'tkazing.



26.20-rasm. WordPress Dashboard.

Save tugmasini bosning. Shu bilan barcha ish bajarildi. Unchalik qiyin

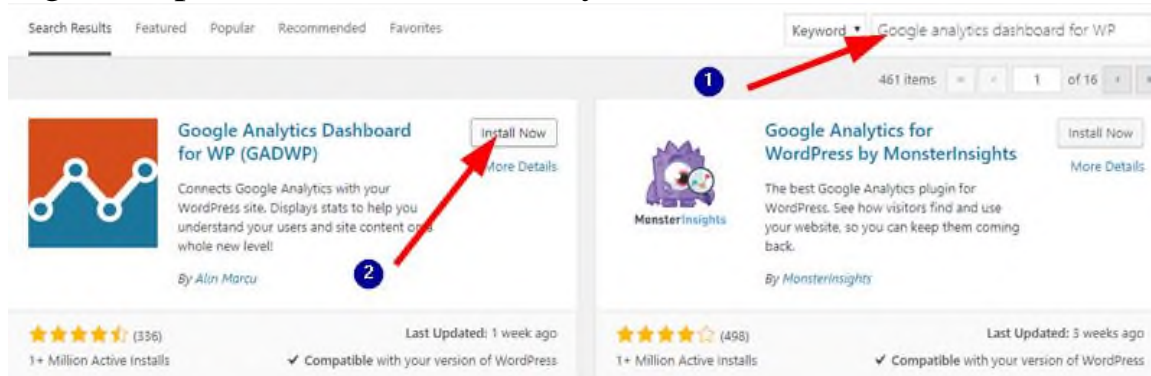
emas, to'g'rimi?

WP boshqaruv paneli uchun Google Analytics kodi qanday qo'shiladi.

Demak, siz WordPress boshqaruv paneli ichida saytingiz statistikalarini ko'rishni xohlaysiz. Bunda Google Analytics Dashboard for WP plaginini o'rnatamiz.

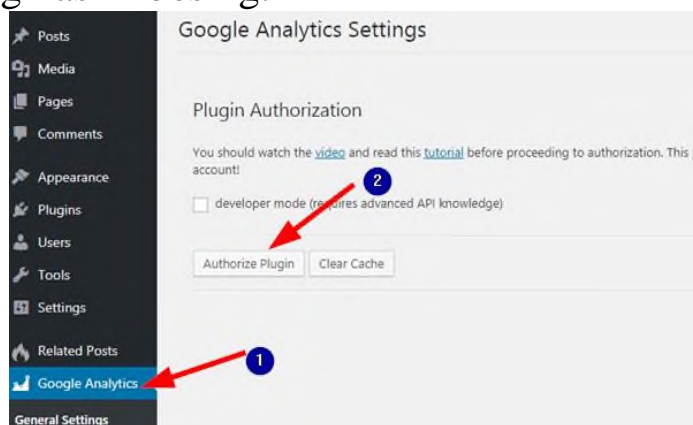
Plaginni o'rnatish juda oson. WordPress Dashboarddan (boshqaruv paneli) Plugins va undan keyin Add New ga kiring va Google Analytics Dashboard for WP ni qidiring.

Plaginni topasiz va oldin Install keyin Activate ni bosasiz.



26.21-rasm. Plaginni o'rnatish.

Endi, WordPress Dashboarddagi Google Analytics va undan keyin Authorize Plugin tugmasini bosing:

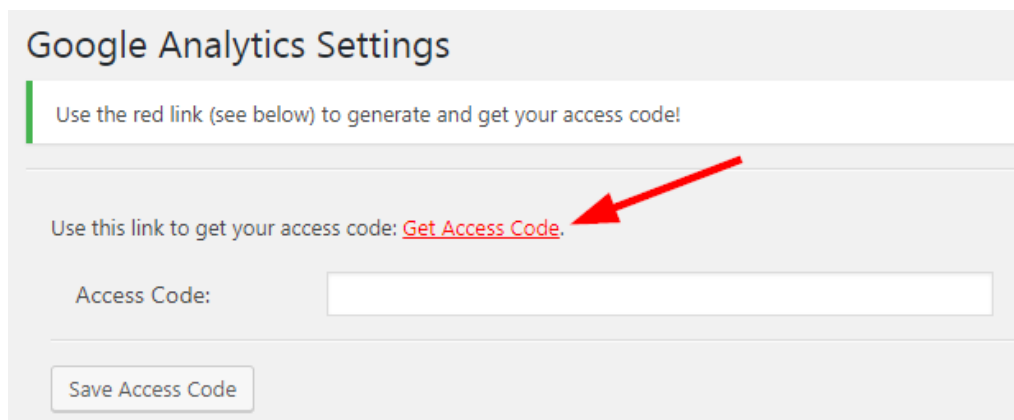


26.22-rasm. Plaginni avtorizasiya qilish.

WordPress Dshboardda statistikani ko'rsatishi uchun, siz bu plagin uchun Google Analytics malumotlaringizga kirishiga ruxsat berishingiz kerak.

Authorize Plugin tugmasini bosganingizdan so'ng, plagin sizni Access Code (kirish kodi) ni kiritishga undaydi.

Kirish kodini olish uchun, qizil Get Access Code ssilkasiga kiring!



Google Analytics Settings

Use the red link (see below) to generate and get your access code!

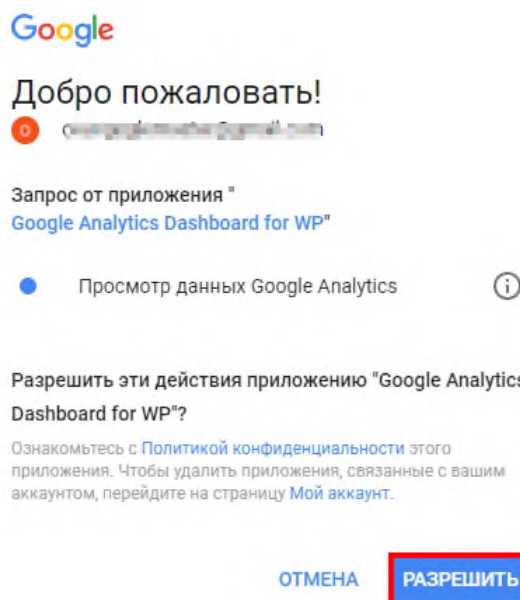
Use this link to get your access code: [Get Access Code](#).

Access Code:

Save Access Code

26.23-rasm. Kirish kodini olish.

Endi Google Analytics akkauntingizga kirish uchun plaginga ruxsat berishingiz kerak. “Разрешить” tugmasini bosing!



Google

Добро пожаловать!

Запрос от приложения "Google Analytics Dashboard for WP"

Просмотр данных Google Analytics

Разрешить эти действия приложению "Google Analytics Dashboard for WP"?

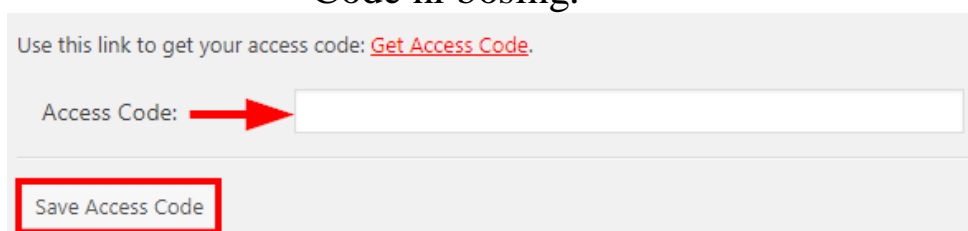
Ознакомьтесь с Политикой конфиденциальности этого приложения. Чтобы удалить приложения, связанные с вашим аккаунтом, перейдите на страницу [Мой аккаунт](#).

ОТМЕНА РАЗРЕШИТЬ

26.24-rasm. Google Analytics akkauntingizga kirish.

“Разрешить” tugmasini bosganingizdan so’ng, Google Analytics kirish kodini beradi:

Bu kodni kopiya (nusxa) qilib oling, WordPress Dashboardga qayting va kodni Access Code formasiga kiriting. Keyin Save Access Code ni bosing.



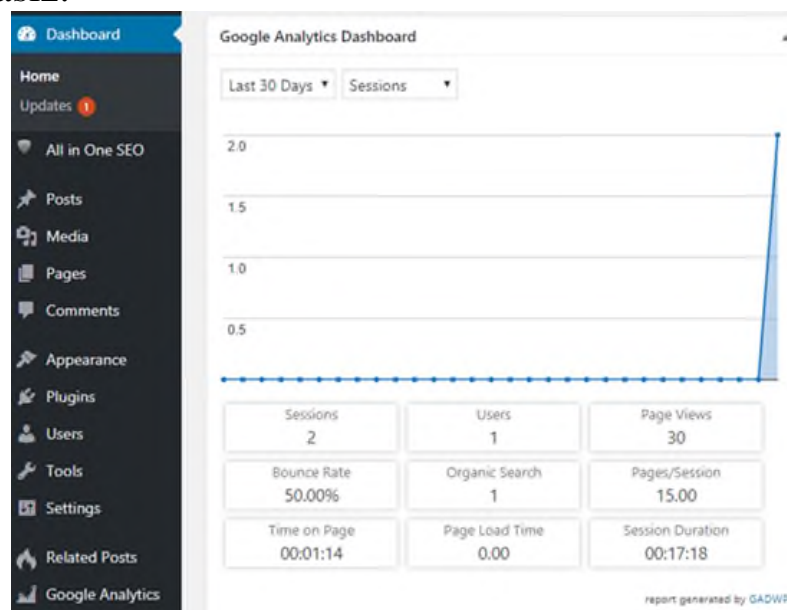
Use this link to get your access code: [Get Access Code](#).

Access Code:

Save Access Code

26.25-rasm. Kodni kiritish.

Siz ishni bajarib bo'ldingiz. Dashboard → Home boshqaruv panelida Analyticsni ko'rasiz.



26.26-rasm. Dashboard → Home boshqaruv paneli.

Google Analyticsni o'zingizga qulay bo'lgan usulda o'rnatib bo'ldingizmi?

Bunda qiyin joyi yo'q edi, agar plagin o'rnatish bo'yicha muammolar bo'lgan bo'lsa plagin o'rnatish haqidagi maqolamdan muammoyingizni yechishingiz mumkin. Boshqa savollar bo'ladigan bo'lsa ham kommentariyada qoldiring.

“Google Analytics» da «Darodar sade referrerini” taniy olishning yo'llari.

Ko'pgina web-sayt egalari uchun havola spam marketing kampaniyasi samaradorligiga ta'sir qiluvchi muhim muammo hisoblanadi. Agar web-sayt egalik qilsangiz, umumiy domenlardan yoki boshqa yangiliklardan kelgan trafikni ko'rishingiz mumkin. Yangi boshlanuvchilar uchun, qonuniy trafik uchun Google'da soxta trafikni xato qilishga moyil bo'lishi mumkin. Shu bilan birga, spam-da'vat Darodarning ruhiy yo'naltiruvchi kabi saytlaridan keladi. Sifat trafigi asosiy maqsadi - o'z saytlariga trafikni oshirish va shirkat savdosini amalga oshirishni maqsad qilgan ayyor web-saytlardan keladi.

Semal dan eng yaxshi mutaxassisi Sohail Saydiq tomonidan ushbu maqola sizni spam-spam-mexanizmlarini va uni Google Analytics-da qanday qilib aniqlashni o'rgatishi mumkin.

Spam spam-ikki xil farqlanishi mumkin:

1. affordable website design services. Trafik spam-web-brauzerlaridan kelgan.

Ko'p turli sabablarga ko'ra web-brauzerlardan foydalanadigan ko'plab web-saytlar mavjud. Qonuniy web-saytlar brauzerlari o'zlarining yagona identifikatorlari bo'lib, butun jarayoni muvaffaqiyatli bajaradi. Web-sayt brauzerlari web-saytingizga kirishadi, shu jumladan maqsadli qurbonga erishish uchun harakat tugmachalariga qo'ng'iroq qilish uchun havolani bosib. Aksariyat hollarda ushbu spam-tavsiyalar saytga dori-darmon kompaniyasiga yoki onlayn-do'konga yo'naltirish sifatida keladi. Boshqa holatlarda, darodar ruhiy yo'naltiruvchi kabi spam-xabarlar porno web-saytlardan kelishi mumkin. Ularning asosiy maqsadi, har bir sotib olish uchun komissiya olishdir. Darodar qonuniy sayt deb da'vo qilmoqda. Ular tegishli SEO web-sayt emas. Foydalanuvchilar Google Analytics sahifasiga soxta trafikni jo'natish uchun o'zlarining tonnalarini yuborish uchun brauzerdan foydalanadilar.

2. Sade yo'naltiruvchi transport. U ruh spam uchun, web-saytida web tashrif yo'q. Buning o'rniga, spamming Google Analytics sahifangizda millionlab sahifa so'rovlarini yuboradi. Xuddi shunday siz GA hisobingizdagi trafikni sezasiz, ammo web-saytingiz boshqaruv panelida aks ettirilmaydi. Ehtimol, bu turdagi tirbandlik Darodarning ruhiy yo'naltiruvchi kabi domenlardan kelib chiqishi mumkin.

Google Analytics da tavsiya etilgan spam tashriflarini qanday aniqlash mumkin. E-tijorat web-saytlari ishlaydigan odamlar uchun soxta trafikni qonuniy bo'lmagan holatdan farqlash muhimdir. Ushbu web tashriflar, sizning saytingiz ma'lumotlarini noto'g'ri qilib, marketing usullarining aniqligini pasaytirishi mumkin. Soxta trafik oldindan taxmin qilinadigan naqshlardan so'ng sodir bo'ladi. Masalan, yo'l-yo'riqlar tirbandlik darajasi 0% yoki 100% ga teng. O'rtacha ko'rish sessiyasi davomiyligi har doim 0 soniya. Xuddi shu web tashriflardan yangi sessiyalar kelmaydi. Uni aniqlashning yana bir muhim usuli - ochilish sahifalaridir . Web-ishlab chiquvchilarning ko'pchiligi ruhiy spamni yaratib, o'zlarining bot-botlarini bosish uchun joyni topishga ishonch hosil qilish uchun katta suzuvchi tugmalardan foydalanadilar.

Boshqa holatlarda Google Analytics'dagi spam-filtrlar jo'natuvchi spamni aniqlay oladi. Bunday web tashriflarni nazorat qilish uchun ilg'or spam-filtrlardan va Darodar hayalet yo'naltiruvchi kabi domenlardan olingan trafiklardan foydalanish juda muhimdir.

Natijada. Yo'naltiruvchi spam sizning raqamli marketingni harakatlaringizni tanqidiy qilib, web-saytingizdagi ma'lumotlarni o'chirib

qo'yishi mumkin. Yo'naltirilgan spamni saqlab qolish uchun xavfsiz elektron pochta platformasidan foydalanish juda muhimdir. Misol uchun, «Darodar» ruhiy yo'naltiruvchi web-saytingizdagi web-saytingizni aks ettira olmaydigan trafik ma'lumotlarini ishlab chiqishi mumkin.

Yandex. Metrika tashrif buyuruvchilar maxfiylikni saqlash haqida qay-g'uradi va yig'ilgan ma'lumotni noxaxsiylikni saqlab qoladi. Shuning uchun ba'zi ma'lumotlar, misol uchun, ijtimoiy-demografik (jins, yosh va boshqalar) ma'lumotlarni chegaralangan tartibda taqdim etadilar. Bu ma'lumotlar tashrif buyuruvchilar soni kamida 10 nafarni tashkil etganda taqdim etiladi.

Misol uchun, kunlik tashrif buyuruvchilar ichida erkaklar necha kishini tashkil etishini aniqlash kerak. Hisobot tu-zilgan vaqtda saytga 5 kishi tashrif buyur-gan (10dan kam). Bunday sharoitda sayt hisoboti kunlik tashrif buyurganlarning umumiy soni haqida ma'lumotni mujassamlashtiradi, ammo tashrif buyurganlar jinsi haqida ma'lumot taqdim etilmaydi.

Hisobotlar ko'rinishi - hisobotlar guruhi. Tashriflar. Saytga tashriflarning umu-miy xaritasi va dinamikadagi konversiya.

Manbalar. Tashrif buyuruvchilarni saytga keltiruvchi manbalarga asosan tashriflarni taqsimoti: reklamali e'lon, qidiruv so'rovi, ijtimoiy tarmoqlar va boshqalar.

Tashrif buyuruvchilar. Tashrif buyuruvchilar tavsifining tahlili: mintaqa, jinsi va yoshi, qiziqishlari, saytga tashrif buyurish-larning soni va davomiyligi va boshqalar.

Mohiyat. Tashrif buyuruvchilarning sayt kontenti bilan o'zaro aloqasi:

- Alohida sahifalarga tashriflar;
- Tashriflar boshlanadigan va tugaydigan sahifalar;
- Sahifalar URLsining parametrlari;
- Tashqi havolalardan o'tishlar;
- Fayllar yuklovi;
- Tashrif buyuruvchi tomonidan amalga oshirilgan buyurtmalar va ularning bahosi, shuningdek, tashriflarning ixtiyoriy para-metrlari (agar muallif hisoblagichni Met-rikaga ushbu ma'lumotlarni taqdim etadigan qilib modifikatsiya qilgan bo'lsa).

O'zini tutish (xulq). Saytga tashrif buyuruvchi ortidan kuzatuv:

- Harakatni video-formatda taqdimi;
- Web-formalar to'ldirilish jarayonining batafsil tahlili;

- Havolalar, kliklar, sahifalar aylanishi va boshqalardan o'tishlarning vizual xaritasi.

Texnologiyalar. Foydalanuvchi tomonidan foydalaniladigan qurilmalar, ope-rativ tizimlar, brauzerlar va boshqa dasturiy ta'minot haqida ma'lumot.

Monitoring. Saytning hammabopligi va saytga yuklama monitoringining natijalari.

Maqsadli qo'ng'iroq. Saytga o'tib kelgan yoki turli manbalardan raqamni olgan potensialga ega mijozning qo'ng'iroqlar statistikasi. Maqsadli qo'ng'iroq qurilmasi foydalanuvchilarga ochiqdir.

Elektron kommersiya. Foydalanuvchi-ning saytining tovar va buyurtmalarning umumiy va detallashtirilgan axboroti (internet-do'kon), misol uchun:

- Savatga qo'shilgan va sotib olingan tovarlar miqdori;
- Barcha buyurtmalardan foyda, bitta bu-yurtmadan foyda;
- Yanada foydali buyurtmalar kelib tushadigan manbalar.

“Tashriflar” hisoboti. Hisobot vaqt oqimiga qarab tashriflarning umumiy tasvirini taqdim etadi.

Belgilangan vaqt oralig'idagi ko'rsatkichlar gistogramma ko'rinishida namoyon bo'ladi.

“Сводка” (ma'lumot) hisoboti. Saytga bo'lgan har bitta tashrif o'zining manbaiga ega. Metrika barcha manbalarni bir qancha guruhlariga to'playdi:

- to'g'ri o'tishlarni saytning manzilini qo'lda tergan yoki brauzer ilovalarida o'tib keluvchi tashif buyuruvchilar amalga oshiradilar;

- saytdagi havolalar orqali o'tishlar tashrif buyuruvchilarni muallifning saytiga keltiradi va sayt manzilini havola bilan referrer sifatidan qoldiradilar;

- qidiruv tizimlaridan o'tishlar qidiruv amalga oshirilganda taqdim etilgan havolalar yordamida amalga oshiriladi. Metrika ko'plab Rossiya va xorijiy qidiruv tizimlarini aniqlaydi. Qidiruv tizimining nomi bilan birgalikda, qoidaga ko'ra, qidiruv jumlasini ham qayd qilindi;

- ijtimoiy tarmoqlardan o'tishlar ijtimoiy tarmoq orqali saytga havolalar yordamida o'tiladi. Ko'plab ijtimoiy tarmoqlar havolalarni joylashtirgan foydalanuvchilar yoki guruhlarni aniqlash imkonini beradi;

- reklama orqali o'tishlar Yandex (Direkt, Market), mediya maydonlar, shuningdek, UTM, Openstat, Google AdWords kabi bir qator «belgilar» kabi reklamali servislar uchun ro'yxatdan o'tadi. Bu kabi

axborotni «Rek-lamali tizimlar» hisobotida olish mumkin;

- saqlangan sahifalardan o'tishlar havola bo'yicha o'tishlar kompyuterda lokal saqlangan havola orqali o'tishni anglatadi (odatda quyidagi ko'rinishdagi referrerga ega bo'ladi C:\\Documents\User\page.html);

- ichki o'tishlar ro'yxatga olinadi (agarda foydalanuvchining saytdagi taym-autda ko'rsatilgan faolligi katta vaqt oralig'iga uzilsayu, ammo foydalanuvchi brauzerni yopmasa). Keyin esa faollik yangilanadi foydalanuvchining yangi «tashrif»i yaratiladi, manba sifatida ushbu tur ko'rsatiladi;

- elektron pochta qutisi tarqatmalari or-qali o'tishlar bu foydalanuvchining o'tish-lari elektron pochta qutisidagi xat orqali amalga oshirilganini bildiradi. Bunday o'tishlarning yanada aniq hisobini olish uchun pochta qutisiga keluvchi shu kabi havolalar UTM metkalariga (utm_source= email ili utm_source=e-mail mavjud) yoki Openstat (o'zida birinchi yoki oxirgi e-mail yoki e-mail). Ba'zi hollarda manba aniqlanmay qolishi mumkin;

- ichki o'tishlar foydalanuvchining saytda tashrifi vaqti tashriflarda belgilangan taym-autidan oshib ketsa, ro'yxatga oladi, ammo foydalanuvchi brauzer oynasini yopmaydi. So'ngra qayta yuklash orqali yangi «tashrif» ro'yxatga olinadi, ushbu tashrif uchun ushbu tur manba sifatida belgilanadi;

- pochta qutisidagi tarqatmalardan o'tishlar foydalanuvchi o'tishi elektron pochta qutisi orqali amalga oshirilganini bildiradi. Bunday o'tishlarning aniq qaydi uchun pochta qutisidagi havolalarga UTM belgisini qo'yish kerak bo'ladi (bularda utm_source= email yoki utm_source=e-mail) yoki Openstat (ularda ilk yoki so'nggi email yoki e-mail) yoki from=email (yoki e-mail);

Ba'zi hollarda manba aniqlanmagan bo'lishi ham mumkin.

HTTPS protokoliga ochiq bo'lgan sayt-lardan o'tishlarni servis To'g'ridan-to'g'ri kirishlar guruhiga kiritadi. Metrika saytga o'tishning bunday manbaini to'g'ri aniq-lashi uchun utm_referrer belgi(metka)si-dan foydalaning.

Ma'lumot(svodka)da ushbu guruhlar bo'yicha to'plangan ma'lumotlar keltirilgan.

“Qidiruv tizimlari” hisoboti. Muallifning saytiga yetaklab kelgan qidiruv tizimlari reytingini o'zida ifoda etadi. Har bir qidiruv tizimi uchun, tashrif buyuruvchilar qidiruvlar natijasida Muallifning saytiga havolani topgan qidiruv iborasi ochiqdir.

“Ijtimoiy tarmoqlar” hisoboti. Hisobot ijtimoiy tarmoqlardan o'tishlar haqidagi ma'lumotni o'z ichiga oladi, bunday tashrif buyuruvchilarning o'zlarini tu-tishlari kuzatish hamda konversiya ko'rsatkichlarini o'lchash imkonini beradi.

Diagramma eng yirik ijtimoiy tarmoqlar - manbalar bo'yicha ma'lumotlarni aks ettiradi. Hisobotda muallifning saytiga havolani joylashtirgan, ijtimoiy tarmoq-ning alohida guruhlari va foydalanuvchilari bo'yicha bo'lingan batafsil statistika taqdim etilgan (bunday bo'linganlikni qo'llab-quvvatlaydigan (ushlab turadigan) ijtimoiy tarmoqlar uchun).

Tashrif buyuruvchining o'zini tu-ti-shining tahlili:

- Web chaqiruv;
- Chaqiriqlar xatirasi;
- Havolalar xaritasi;
- Skrolling xaritasi;
- Shakllar tahlili.

Webvazor. Webvazor texnologiyasi Yandeks. Metrikaning foydalanuvchilariga o'z saytlarida tashrif buyuruvchilarning o'zlarini tutishlarini sifat jihatidan yangi darajada tahlil qilish imkonini beradi. Sayt egasi tashrif buyuruvchining harakatlarini videoformatida tasvirlashi mumkin, ular buni na-vigatsiyani amalga oshirishlari kabi har bir sahifada qo'llaydilar, sichqoncha kursorni siljitadilar, havolalar bo'ylab chaqiradilar. Tashrif buyuruvchilar o'zlarini tutishlarining ba-tafsil tahlili navigatsiya, mantiq (logika) va yuzabilitidagi muammolarni aniqlashda, natijada esa - saytning konversiyasini oshi-rishda yordam beradi. Ushbu ma'lumotlar ochiq bo'lishi uchun Webvizorni ulash va sozlash lozim.

Saytni qanday qilib tahlil qilish ke-rak. Webvazor tizimi tomonidan to'plangan ma'lumotlarni hattoki oddiygina ko'rib chiqilishi ko'p narsalarni, ya'ni tashrif buyu-ruvchilar saytdan qanday foydalanishlari, ular qanday murakkabliklarga duch kelayotganliklari, nima uchun u yoki bu harakatlarni bajarayotganliklarini bilib olish imkonini beradi. Tashrif buyurishlarning ko'plab miqdori yig'ilganda, ularning hammasini o'rganib chiqishdan ma'no yo'q. Foydala-nuvchilarning fokus-guruhlarini shakllantirish, ular bo'yicha tanlanmalar qilish va har bir guruh ichidagi axloqni tahlil qilish lo-zim. Buning uchun zarur bo'lgan quyidagi barcha vositalar mavjud:

- har bir tashrifni eng batafsil tafsilotlari: trafik manbai, reklama kampaniyasi, qidiruv so'rovi, avvalgi tashrif vaqti va boshqalar;
- filtrlarning qudratli tizimi tashrif bu-yurishlarni har qanday parametrlar va ularning birikmalari bo'yicha segmentlashtirish imkonini

beradi;

- maqsadlarni belgilash mexanizmi boshqalaridan konversiya sodir bo'lgan tashriflarni ajratish imkonini beradi.

Aynan nimani tahlil qilmoq kerak. Birinchi navbatda, saytning kalitli sahifa-la-rida tashrif buyuruvchilarning o'zlarini tutishlariga e'tibor bering. Bular tashrif buyuruvchi kalitli harakatlarni amalga oshi-radigan yoki muhim qaror qabul qiladigan sahifalardir. Bunday sahifalarga:

- Kirish sahifalari. Reklama e'lonlari olib boradigan sahifalarga alohida e'tibor qaratish lozim;

- Qidiruv shaklli sahifa (tovarlar, para-metrlar bo'yicha tanlash);
- Tovar katalogi sahifasi;
- Tovar kartochkasi;
- Savat (korzina) sahifasi;
- Buyurtmani rasmiylashtirish sahifasi;
- To'lov sahifalari kiradi;

Shuningdek, quyidagi jarayonlarga alohida e'tibor qaratmoq lozim: buyurtmani ro'yxatga olish yoki rasmiylashtirish jarayoniga. Webvazor yordamida misol uchun, ro'yxatga olishning qaysi qadamida sayt tashrif buyuruvchilarni yo'qotayotganligini va buyurtma rasmiylashtirilishida navigatsiya shaffof yoki shaffof emasligini bilib olish mumkin. Tashrif buyuruvchilarning o'zlarini tu-tishlari tahlilini klik va havolalar xarita (karta)lari bilan to'ldirgan holda, tashrif buyuruvchilar saytda o'zlarini qanday tuta-yotganliklari haqida to'liq tasavvurga ega bo'linadi.

Kliklar xaritasi bu - Muallifning saytidagi kliklar bo'yicha statistikani o'lchash va aks ettirish vositasidir. Xarita kliklarni sahifaning barcha elementlari bo'yicha aks ettiradi (shu jumladan, havola bo'lmaganlari bo'yicha ham). Kliklar xaritasi sayt yuzabilitisini baholash, chaqiriqboproq elementlarni aniqlash, dizaynning qaysi elementlari saytlar tashrif buyuruvchilariga, lekin bunday bo'lmasada, havola bo'lib tuyulishini aniqlashga yordam beradi. Kliklar xaritada, ularning chastotalaridan kelib chiqqan holda, turli ranglar bilan yoritiladi. Sahifaning ostidagi panelda sahifadagi kliklarning umumiy soni hamda muallifga ko'rinadigan joriy maydonga tushgan kliklarning ulushi ko'rsatiladi. Foydalanuvchi xaritani aks ettirishning bir necha rejimlaridan foydalanishi mumkin:

- "Issiqlik xaritasi" - iliq ranglar tez-tez bo'ladigan chaqiriqlarga, sovuqroqlari esa - kamdan-kamlariga to'g'ri keladi;

- "Monoxrom xarita" - rang zichligi mazkur nuqtadagi kliklar

chastotasiga to'g'ri keladi;

- “Havola va tugmachalar bo'yicha kliklar” - havola yoki tugmachalar bo'l-magan elementlar bo'yicha kliklar xaritada aks ettirilmaydi;

- “Shaffoflik xaritasi” - kliklar xaritasi “tumanli niqob” kabi aks ettiriladi: chaqiriqboproq elementlar “tuman” ichra aniqroq ko'rinadi;

- “Elementlar xaritasi” - xaritada sayt sahifasining barcha elementlari aks ettiriladi.

| Brauzer | Sahifaning maksimal balandligi/
kengligi (piksellarda) | Piksellarning maksimal miqdori |
|-------------------|---|--------------------------------|
| Mozilla Firefox | 32 767 | 472 907 776 (22 528 x 20 992) |
| Chrome | 32 767 | 268 435 456 (16 384 x 16 384) |
| Internet Explorer | 8 192 | 67 108 864 (8 192 x 8 192) |

26.27-rasm. Brauzerlarda xaritani aks ettirish uchun qiymatlar.

Ma'lumotlarni `iframe`'da ko'rsatilishi. Agarda muallifning sayti `iframe` da ko'rsatishdan himoyalangan bo'lsa (ya'ni serverni sozlashlarda `X-Frame-Options` sarlavhasidan foydalanilsa), to'plangan ma'lumotlarni ko'rib bo'lmaydi. Saytga tashriflar haqidagi ma'lumotlarni ko'rib chiqish uchun, server sozlashlariga `webvisor.com` domeni va uning domenostilari uchun, shuningdek, muallifning sayti domeni uchun muntazam ifoda yordamida istisnolarni qo'shib qo'yish kerak bo'ladi.

```
https?:://([^\.]+\.)?yourdomain\.com|  
([^\.]+\.)?webvisor\.com/
```

Agarda foydalanuvchi `nginx` dan foydalansa, sozlamaga quyidagi istisnolarni qo'shib qo'ying:

```
location / {  
    set $frame_options '';  
    if ($http_referer !~ '^https?://  
    ([^\.]+\.)?yourdomain\.com|  
    ([^\.]+\.)?webvisor\.com/') {  
        set $frame_options 'SAMEORIGIN';  
    }  
    add_header X-Frame-Options $frame_  
    options;  
    ...  
}
```

bunda `yourdomain\.com` - muallif saytining domeni nomi. Saytning domen zonasidan foydalanish lozim (`com` dan farq qilishi mumkin).

Sahifalar o'lchamining cheklanishi. Xaritada ma'lumotlarni aks ettirish uchun sayt sahifalari o'lchamlarining ayrim cheklovlariga rioya qilish lozim. Quyida turli brauzerlarda xaritani aks ettirish uchun qiymatlar 1-jadvalda ko'rsatilgan:

Havolalar xaritasi - muallifning saytidagi havolalar bo'yicha o'tishlar statistikasini o'lchash vositasi. Xaritadaagi havolalar, ularning ommabopligidan kelib chiqqan holda, turli ranglar bilan yoritiladi.

Havolaga chaqiriq qilinganda quyidagi ma'lumotlar aks ettiriladi:

- havolalar bo'yicha o'tishlar soni;
- sahifadagi boshqa havolalarga nisbatan o'tishlar ulushi.

Xarita faqatgina Yandex. Metrika hisoblagichi o'rnatilgan sayt sahifalarida, shuningdek, agarda saytda UTF-8 kodlashtirilishidan foydalanilgan bo'lsagina, aks ettiriladi. Hisobotda redirektga olib boruvchi havola(ссылка)lar yoritilmaydi. Bunday havolalar yoritilishi uchun yo'riqnomaga amal qiling. Xarita URL dan qurilganda www, reklama belgilari va # dan keyin havolaning bir qismi (shu jumladan, qadar) kabi elementlar istisno qilinadi, biroq URL ning oddiy parametrlari istisno qilinmaydi. Shuning uchun kliklar xaritasi <http://site.com?id=1> va <http://site.com?id=2> sahifalari uchun har xil bo'ladi.

NAZORAT SAVOLLARI

1. DNS xizmat tushunchasi?
2. Hosting xizmat turlari?
3. Web analitika va Google Analytics dan foydalanish?

GLOSSARIY

1. API (Application Programming Interface – Ilovalarni dasturlash interfeysi) - Kutubhona yoki modullar funksional dasturlariga kirish uchun foydalaniladigan usul va funksiyalar to'plami
2. CMS (Content Management System – Kontentlarni boshqarish tizimi) - Ma'lumotlarni o'chirish, qo'shish va o'zgartirishning qulay vositalarini taqdim etuvchi, web-sayt interaktiv modullari va ma'lumotlarni to'ldirishni boshqaruvchi tizim
3. CSS (Cascading Style Sheets – Kaskadli jadval stillari) - HTML, XHTML va XML-hujjatlarning dizayni uchun mo'ljallangan, stil xususiyatlarining standart tili
4. CSS-design (CSS-dizayn, CSS-joylashtirish) - Hujjatni joylashtirish usuli, tuzilish va taqdimotning tushunchasini aks ettiradi
5. DB (Database – Ma'lumotlar bazasi) - Bitta yoki bir nechta mahsus fayl shakllarida mavjud bo'lgan, tuzilmaviy va tashkiliy ma'lumotlar to'plami
6. DOM (Document Object Model – Hujjatning ob'jekt modeli) - Dasturlar va skriptlarga HTML va XML-hujjatlar tarkibini boshqarish, hamda ularning tuzilishi va dizaynini o'zgartirishga imkon beruvchi, platformadan mustaqil

- dasturiy interfeys (API)
7. HTML (HyperText Markup Language – Gipermatnli belgili til) - SGML ilovasi hisoblanuvchi va gipermatnli hujjatlar tuzilmasini yozish uchun mo'ljallangan, standart belgili tillardan biri
 8. HTTP (HyperText Transfer Protocol – Gipermantlarni uzatuvchi protokol) - Web-sahifa va fayllarni uzatish uchun mo'ljallangan, amaliy bosqichdagi tarmoq protokoli
 9. PHP (Hypertext Preprocessor – Gipermatn prosessori) - Ma'lumotlar bazasi bilan ishlashda va web-server tomonidan web-sahifalarni generasialash uchun mo'ljallangan, dasturlashning mashhur script tillaridan biri
 10. W3C (World Wide Web Consortium – Butun dunyo o'rgimchak to'ri konsorsiumi) - Internet tarmog'i uchun standart texnologiyalarni qo'llaydigan va ishlab chiqaradigan tashkilot
 11. Webpage (Web-sahifa) - Standart belgili tillardan biri yordamida tashkil etilgan, gipermatnli hujjat
 12. Web-server - Web-sayt funksionalligini ta'minlovchi, mahsus dasturiy ta'minot. Web-serverning asosiy vazifasi – kontentning dinamik generasiyasi uchun skriptlarni ishga tushirish va brauzer orqali

- foydalanuvchiga web-sahifani uzatish
13. Web-sayt - Web-saytning qat'iy belgilangan atamasi mavjud emas. Web-saytni Internet tarmog'ida o'zining unikal manziliga ega bo'lgan joy sifatida tavsiflash mumkin
 14. XHTML (Extensible HyperText Markup Language – Gipermatnli kengaytirilgan belgili til) - XML ilovasi hisoblanuvchi, standart belgili tillardan biri va gipermantli hujjatlar tuzilmasini yozish uchun mo'ljallangan
 15. XML (Extensible Markup Language – Kengaytirilgan belgili til) - SGML standart belgili tilining soddalashtirilgan qismi bo'lgan standart formatlash tillaridan biri va tuzilmaviy ma'lumotlarni saqlash, dasturlar o'rtasida axborot almashinuvi va belgili hujjatlarda boshqa fiksirlangan tizimlarni yaratish uchun mo'ljallangan
 16. XML Declaration (XML e'lon qilish) - XML-hujjatni qayta ishlash uchun maxsus qo'llanma
 17. XML Schema (XML-hujjatlar tuzilmasini yozuvchi til) - XML-hujjat sintaksisi va tuzilmasini aniqlovchi standart
 18. Giper murojaat - Web-sahifa giper murojaat deb nomlanuvchi, matnli satrlarning boshqa sahifalariga yo'naltiriladi

19. JavaScript
- Netscape tomonidan ishlab chiqilgan Internetning eng mashhur senariyli tillaridan biri
20. JPEG (Joint Photographic Expert Group)
- Ushbu format JPEG va JPG formatidagi siqilgan tasvir formatlarini saqlash va grafikalarini rivojlantirishga qaratilgan
21. MySQL
- Ochiq kodli ma'lumotlar bazasining dasturiy ta'minoti

FOYDALANILGAN ADABIYOTLAR RO'YHATI

Asosiy adabiyotlar

1. Макфарланд Д. JavaScript и jQuery: исчерпывающее руководство. [пер.с англ. М.А. Райтман]. – 3-е издание. – М.: Эксмо, 2015. -880 с.
2. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. – СПб.: Питер, 2011. –496 с.
3. Макфарланд Д. Новая большая книга CSS – СПб.: Питер, 2016. –720 с.
4. Гольцман В. MySQL 5.0. Библиотека программиста. Питер; Санкт-Петербург; 2010.

Qo'shimcha adabiyotlar

1. Мирзиёев Ш.М. Буюк келажегимизни мард ва олижаноб халқимиз билан бирга кураимиз. 2017.
2. Мирзиёев Ш.М. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. 2017.
3. Мирзиёев Ш.М. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. 2017.
4. Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. – СПб.: Питер, 2011. -272с.
5. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. 2-е изд. – СПб.: Питер, 2014. –320с.
6. Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство. [пер.с англ. М.А. Райтман]. – 4-е издание. – М.: Эксмо, 2014. -528 с.
7. Стефанов С. Шаблоны. – Пер.с англ. – СПб.: Символ-Плюс, 2011. – 272 с.
8. Мэтт Зандстра. PHP объекты, шаблоны и методики программирования. 2-е издание. Москва. 2010.
9. Шварц Б., Зайцев П., Ткаченко В., Заводны Дж., Ленц А., Бэллинг Д. MySQL оптимизация производительности, 2-издание. – Пер.с англ. –СПб.: Символ-Плюс, 2010. 832 с.
10. Горнаков С.Г. Осваиваем популярные системы управления сайтом (CMS). Москва 2009.
11. Рамел Д. Самоучитель Joomla!: Пер.с англ. – СПб.: БХВ-Петербург. 2008. – 448 с.

12. Котеров Д.В., Костарев А.Ф. PHP5. – СПб.: БХВ-Петербург. 2005. – 1120с.
13. Зольников Д.С. PHP5. Как самостоятельно создать сайт любой сложности – 2-е изд., стер. – М.: НТ Пресс, 2007. – 272 с.
14. Когзолл Д. PHP5. Полное руководство. : Пер. с англ. – М.: Издательский дом «Вильямс». 2006. – 752 с.
15. Шлоссейгл Д. профессиональное программирование на PHP. : Пер.с англ. – М.: Издательский дом «Вильямс». 2006. – 624 с.
16. Plug-in PHP 100 power solutions. By Robin Nixon. 2010 year. 383 pages. <http://mhprofessional.com/>; Create dynamic sites with PHP & MySQL. www.id.uzh.ch/cl/zinfo/pdf/php-mysql.pdf

Internet saytlari

1. <http://www.tutorialspoint.com/>
2. <http://www.intuit.ru/>
3. <http://www.php.net>
4. <http://www.phpframework.com>
5. <http://www.aspnethelp.com/>
6. <http://www.natahaus.ru/>
7. www.w3schools.com/
8. www.tutorialspoint.com/php/
9. <https://www.codecademy.com/tracks/php>
10. www.freewebmasterhelp.com › Tutorials › PHP
11. <http://www.htmlbooks.ru/>