

O'ZBEKISTON RESPUBLIKASI

OLIY VA O'RTA MAXSUS TA'LIM VAZIRLIGI

Namangan muhandislik – qurilish instituti

WEB DASTURLASH

Namangan 2020 yil

UO‘K:

KBK:

I –

Jakbarov O., G‘oyipov U.

**Web dasturlash: O‘quv qo‘llanma – N.: “Namangan” nashriyoti,
2020 – 200 b.**

Taqrizchilar: NamMTI, Texnologik jarayonlarni avtomatlashtirish va boshqarish va informasion texnologiyalar kafedrası dosenti **K.D.Ismanova**

Namangan davlat universiteti “Amaliy matematika” kafedrasining dosenti, pedagogika fanlari nomzodi, dotsent **N.Ataxanov**

Namangan muhandislik-qurilish instituti “Oliy matematika” kafedrası professori, fizika-matematika fanlari doktori **V.Xojiboev**

Mazkur o‘quv qo‘llanma oliy ta‘lim tizimining 5111000-Kasb ta‘limi (5330200-Informatika va axborot texnologiyalari) bakalavr ta‘lim yo‘nalishi talabalari uchun mo‘ljallangan bo‘lib, talabalarga Internet tarmog‘i uchun Web ilovalarni yaratish bo‘yicha dasturlash tillari, amaliy ko‘nikmalarga ega bo‘lishlari uchun har bir bo‘limlar va mavzular kesimida misollar bilan tushuntirish keltirib o‘tiladi. SHuningdek, o‘quv qo‘llanmada HTML gipermatn belgilash tili, CSS hamda JavaScript Web ilovalarni yaratish tillari bo‘yicha asosiy tushunchalar amaliy masalalar bilan bog‘lab mustaqil o‘rganishga yo‘naltiriladi. SHuningdek, ushbu o‘quv qo‘llanma orqali talabalarni mantiqiy fikr yuritish qobiliyatini oshirish, ilmiy adabiyotlar bilan mustaqil ishlashga o‘rgatish, va yangi kompyuter texnologiyalari yordamida internet tarmog‘i uchun ilovalar ishlab chiqishni o‘rgatishga mo‘ljallangandir. Shuningdek, o‘quv qo‘llanmadan mazkur fanni o‘rganmoqchi bo‘lgan barcha bilim oluvchilar uchun keng foydalanishlari tavsiya etiladi.

ISBN 978-9943-328-58-2

Jakbarov O., G‘oyipov U.

Kirish

Bugungi kunda Internetning ommaviyligi haqida gapirish o'rinsiz. Internet hayotimizning bir bo'lagiga aylandi, biz uning xizmatlaridan har kuni foydalanishga odatlandik. Hozirda ixtiyoriy inson webtexnologiyalarning inson hayotining ta'lim, kommersiya, siyosat, ko'ngil ochar, ... bo'laklariga kirib borganligini tasavvur eta oladi va uning guvohi va foydalanuvchisiga aylanmoqda.

Xususan, Axborot texnologiyalari va kommunikasiyalari sohasida boshqaruv tizimini yanada takomillashtirish, elektron davlat xizmatlari va telekommunikasiya xizmatlari spektrini kengaytirish, telekommunikasiya infratuzilmasini rivojlantirish maqsadida, O'zbekiston Respublikasi Prezidenti SH.Mirziyoev tomonidan 2018 yil 19 fevral PF-5349-sonli "Axborot texnologiyalari va kommunikasiyalari sohasini yanada takomillashtirish chora-tadbirlari to'g'risida"gi Farmonida ham bugungi kundagi AKT sohasida qator kamchiliklar va ularni bartaraf etish bo'yicha alohida topshiriqlar berib o'tiladi.

Dastlabki web-sahifalar juda sodda tuzilishga ega bo'lib, ular matnni formatlash va giperko'rsatkichlardan tarkib topgan edi. Web texnologiyalar rivojlanishi natijasida Web sahifalar tarkibida Plug-in dasturlar joylashtirila boshlandi, natijada Web sahifalarga inter faol xususiyati berildi. Web texnologiyalarning rivojlanishining oxirgi natijalaridan biri bu skript tillaridir (Script Languages).Ularni ishlatishdan maqsad Web serverining ishini yengillashtirish, xar-xil ishlar uchun Web serverini bezovta qilmasdan, bunday masalalarni foydalanuvchi kompyuterining o'zida yaratishdir. Web texnologiyasining oxirgi erishgan yutuqlaridan biri dinamik Web sahifalardir. Dinamik Web sahifalar CGI dasturlar bilan bevosita bog'liq bo'lib, CGI dasturlar serverda joylashgan va server imkoniyatlarini ishlatuvchi dasturlardir.

Mazkur qo'llanma web-hujjatlarni yaratish, ularni Internetda chop etish, web-hujjatni ko'rkamlashtirish, qiziqarli va o'ziga tortuvchi qilib yaratish, vaqti kelsa ma'lumotlarni yangilash kabi vazifalarni o'rgatishga mo'ljallangan.

Shu bilan birga, axborot texnologiyalari va kommunikasiyalarini boshqarish va joriy etish sohasidagi muammo va kamchiliklarni bartaraf etishga xissa qo'shish maqsadida ushbu sohaning mutaxassislarini yetishtirishda manba sifatida ushbu o'quv qo'llanma ishlab chiqildi.

1-BOB. "WEB DASTURLASH" FANIGA KIRISH.

1.1-§. Web dizaynning asosiy tushunchalari. Web texnologiya: HTML; Web brauzerlar; Web-saxifa, Web-sayt. Web-server; razmetkali va senariyli dasturlash tillari; "klient-server" texnologiyasi. Server strukturasi. Web xosting.



O`quv modullari

web-sahifa, web-sayt, web-server, HTML, XML, XHTML, WML, klient-server texnologiyasi.

Web sahifa Internet tarmoqlarida joylashgan fayllar to'plami bo'lib, ularni soni soat sayin ko'payib bormoqda. Bu fayllarda ma'lumotlarni turli xillarini: matn, grafik, tasvir, video, audio ma'lumotlarni uchratish mumkin. Bugungi kunda Web Internet resurslari ichida eng ommaviysi hisoblanadi. Chunki, avvaldan tayyorlangan Web sahifa orqali tegishli ma'lumotlarni to'ldirish foydalanuvchining qanchadan-qancha vaqtini tejash imkonini beradi.

Web-texnologiya klassifikatsiyasi. Web-texnologiyani (Internet-texnologiya) o'rganishni Web-dizaynning quyidagi uchta tushunchasini o'rganishdan boshlaymiz:

1. Web-sahifa;
2. Web-sayt;
3. Web-server.



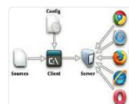
Texnologiya grek tilidan (techne) tarjima qilganda san'at, maxorat, bilish ma'nolarini anglatadi, bular esao'z navbatida jarayonlardir. Jarayonlar - bu qo'yilgan maqsadga erishish uchun ma'lum xarakatlar majmuasidir.



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



Web-sayt – bir qancha web-sahifalarning mantiqiy birlashmasi.



Web- server – tarmoqqa ulangan kompyuter yoki undagi dastur hisoblanib, umumiy resurslarni klientga taqdim etish yoki ularni boshqarish vazifalarini bajaradi. Internet tarmog'ini foydalanuvchilarga tarmoq resurslaridan erkin foydalanish imkoniyatini beradigan web- serverlarsiz tasavvur etib bo'lmaydi. Bunday serverlarda Internetda taqdim etilgan axborotning katta qismi jamlangan.

Web-serverlar ma'lumotlar bazalari va multimediyali ma'lumotlarni bir biriga moslashtiradi. Webserverda Web-sahifa va Web-saytlar saqlanadi. Web-

serverda mijoz kompyuteri tizimini tashkil qilishning umumiy tamoyillari nuqtai nazaridan mijozserver texnologiyalari ishlatiladi.

Hozirgi kunda oddiy web-serverni yaratish texnologiyasini ancha oddiy vazifa deb hisoblash bo'ladi. Asosiy qiyinchilik server sahifasini badiiy bezashdan

Razmetkali va senariyli dasturlash tillari

Maxsus til mavjud bo'lib, bu til yordamida matnlar, grafik ma'lumotlar Web-sahifa hujjatga joylashtiriladi va bu hujjatni barcha kompyuterda ko'rish imkoniyati mavjuddir. Bunday maxsus tillar razmetkali tillar deb ataladi. Ularning asosiy vazifasi – Web-sahifaga “ma'lumotlarni joylashtirish” va ular orasidagi aloqani (gipersaloqalar) ta'minlashdan iborat. Razmetkali tillar quyidagilarni o'z ichiga oladi:

HTML (HyperText Markup Language)

Dastlab World Wide Web tizimi matnli ma'lumotlarni va HTML hujjatlarni ko'rishga mo'ljallangan, matnni taxrirllovchi tilga o'xshash tizim bo'lgan. Ayni damda HTML tili WWW daga eng ommabop tillardan biri hisoblanadi. HTML tilida yozilgan ma'lumotlar o'z ichiga matn 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 yer oynasida ma'lumotlar turlicha ko'rinadi. HTML hujjatga ma'lumotlarni joylashtirish va tahrirlash uchun yuzlab teglar mavjud. Masalan, <p> va </p> teglari abzatsni tashkil etadi, <i> va </i> juft teglari esa, matnni yozma (kursiv) holda ko'rsatish uchun qo'llaniladi. Shu bilan birga gipermatnli ssilkalar teglari ham mavjud. Ushbu elementlar foydalanuvchiga gipermatn ustiga sichqoncha kursori bosilganda boshqa hujjatga bog'lanish imkonini beradi. Butunjaxon o'rgimchak to'rining asosiy va HTML ning tarkibiy qismini gipermatnlar va giper murojaatlar tashkil etadi. Maxsus komandalar yordamida matnning ma'lum qismi shunday ajratiladiki, natijada o'sha matn ustiga sichqon tugmasi bosilsa boshqa matn yoki sahifa ochiladi. Bundan tashqari multimediya vositalarining ishlab ketishi yoki bo'lmasa, ma'lumotni diskda saqlash taklifi ham berilishi mumkin.

Quyida biz HTML tilida tuzilgan dastur kodi bilan tanishib chiqamiz:



1.1.1-misol. Web sahifaga fon berish

<HTML>

<HEAD>

<TITLE> Sahifa fonini berish misoli </TITLE> </HEAD>

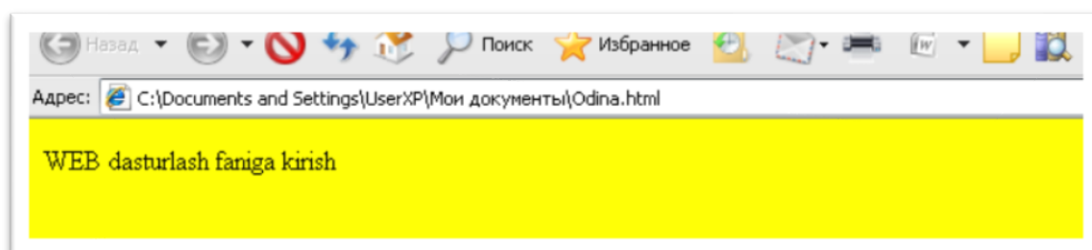
<BODY bgcolor q “yellow” text q “black” link q “red” vlink q “purple” alink q “green”>

WEB dasturlash faniga kirish

</BODY>

</HTML>

Bu dasturni ishga tushirish natijasida quyidagi oyna ochiladi:



1.1.1-rasm

Gipermatn yoki gipermurojaat biror bir tasvirga ham qo'yilishi mumkinki uning ustiga bosilganda ham yuqorida aytilgan holatlar ro'y berishi mumkin. Har bir web-saxifa o'zida bir nechta gipermatn yoki gipermurojaatlarnimujassam etishi mumkin. Gipermurojaatlar web-saytlar bo'ylab xarakatning asosi xisoblanadi. Murojaatni tanlaganda foydalanuvchi brauzer oynasiga yuklanuvchi yoki yordamchiprogrammani ishga tushuruvchi URL bilan bog'langan adresga «tushib» qoladi. Ba'zan gipermurojaat natijasi e-mail yoki FTP serverga yo'llanma beruvchi yangi web-saxifani ochilishiga olib keladi. Foydalanuvchi murojaatni tanlab olishi uchun web-dizayner uni yaratishi kerak. Gipermurojaat yaratish uchun <a> (anchor, yakor) elementidan foydalaniladi. U o'zida yo'llanma beruvchi URL adresni ko'rsatuvchi href atributi bilan to'ldiriladi. Shuning uchun gipermurojaatni yaratish uchun URL adresni aniqlab olish kerak. Gipermurojaat yaratish uchun (anchor, yakor) elementidan foydalaniladi. U o'zida yo'llanma beruvchi URL adresni ko'rsatuvchi href atributi bilan to'ldiriladi. Shuning uchun gipermurojaatni yaratish uchun URL adresni aniqlab olish kerak. Gipermurojaatni yaratishda agar Internetdagi xizmat yoki adresdan foydalanmoqchi bo'lsak albatta uning to'liq adresini ko'rsatish shart. Agar o'zimizda bor bo'lgan web-saxifalardan gipermurojaatlar yaratmoqchi bo'lsak ba'zi bir ishni osonlashtiruvchi xolatlar mavjud:

XML (eXtensible Markup Language).

XML tili ham HTML tiliga o'xshash til xisoblanadi. 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. HTML teglari hujjatni ekranda ko'rinishini ifodalaydi. XML teglari hujjatdagi ma'lumotlarni tavsiflash uchun ishlatiladi. Undan tashqari XML yordamida yangi teglarni yaratish mumkin. XMLda ma'lumotlar tuzilmaviy holda saqlanadi. XML

asosan ma'lumotlar almashinuvida ko'p ishlatiladi, chunki XML platformadan mustaqil bo'lib, HTTP orqali ishlashi juda qulay. Quyida XMLda tuzilgan dastur bilan tanishamiz:



1.1.2-misol. XMLda hujjat yaratish

```
<?xml version="1.0"?>
```

```
<kitob>
```

```
<nomi> Oddiy XML </nomi>
```

```
<sana> 16 aprel, 2011 yil </sana>
```

```
<muallif>
```

```
<familiyasi>Imomova</familiyasi>
```

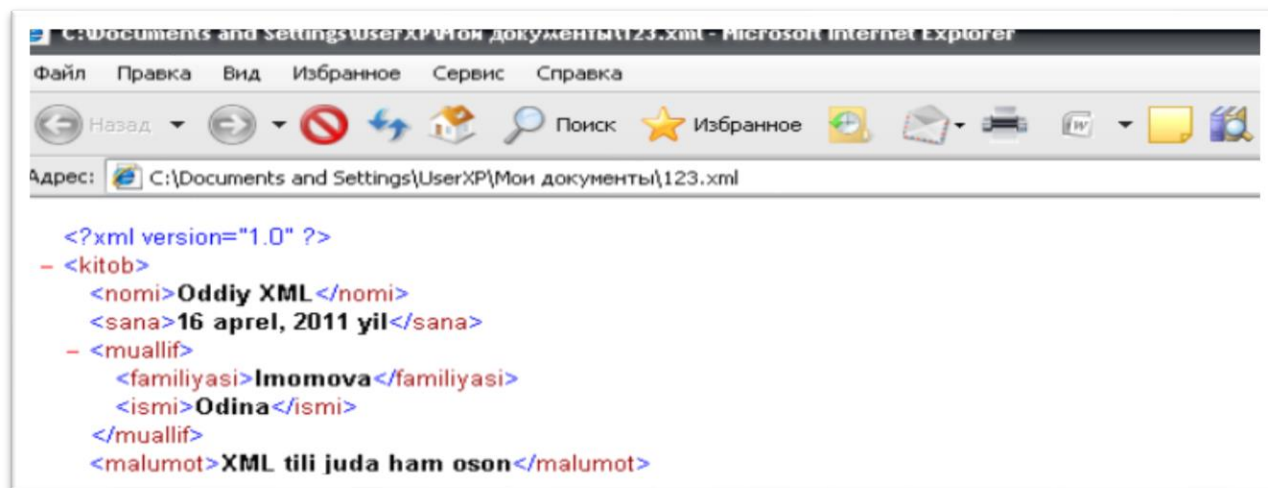
```
<ismi> Odina </ismi>
```

```
</muallif>
```

```
<malumot> XML tili juda ham oson </malumot>
```

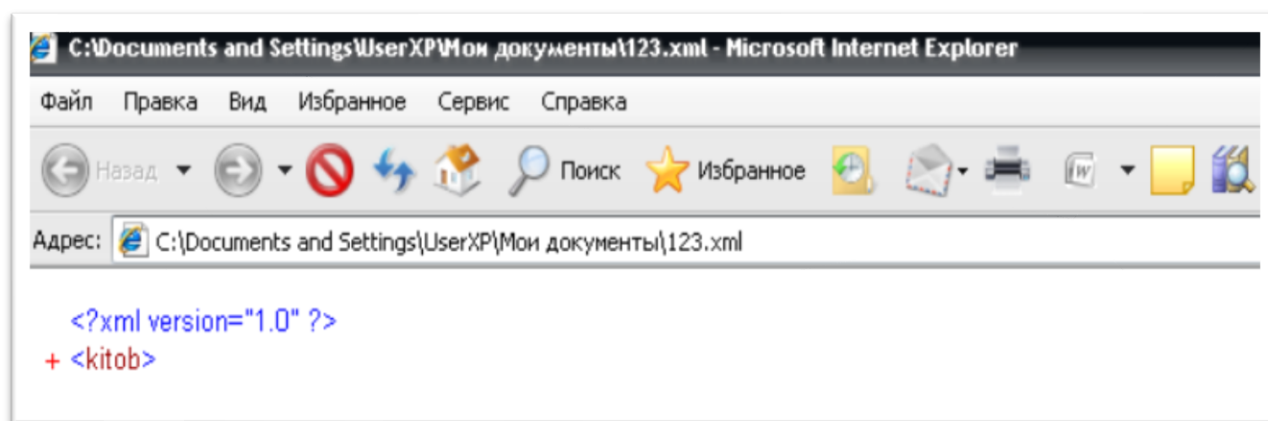
```
</kitob>
```

Bu dastur bloknotga yoziladi va <dastur_nomi>.xml ko'rinishida saqlanadi. Bu dasturni ishga tushirish natijasida quyidagi oynaga ega bo'lamiz



1.1.2-rasm

Buni quyidagicha qisqartiriladi:



1.1.3-rasm

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 (ChannelDefinition Format) - Microsoft ishlab chiqqan brauzerlarda push-kanal hosil qilishda qo'llaniladi.

Ssenariyli tillar. "Klient-server" texnologiyasi

Hozirda Web-sahifaning rivojlanishi yanada interaktiv pog'onasiga 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: klient tomonidagi dasturlarlash (client-side) va server tomonidagi (server-side). Ushbu texnologiyalarni tushunish uchun avvalo bevosita "klient-server" texnologiyasini tushunish kerak

Web-sahifaning interaktiv dasturi ssenariy deb ataladi. Bunday atama dasturning natijasiga bog'liq holda vujudga kelgan. Uning asosiy vazifasi Web-sahifasida foydalanuvchi holatiga, harakatiga «reaksiya» berishdir. Shu tariqa ssenariylar klient tomonida bajariluvchi va server tomonida bajariluvchi ssenariylarga bo'linadi. Klient tomonida bajariluvchi ssenariylar brauzer yordamida bajariladi. Server tomonida bajariluvchi ssenariylar esa Web-server yordamida bajariladi.

Klient tomonidagi ssenariylar.

Klient 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

JavaScript – bu til Netscape va Sun Microsystems tomonidan yaratilgan bo'lib, Web-sahifaning funksional imkoniyatlarini o'ttirish 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 tipini aniqlash mumkin. JavaScript-ssenariylar foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni to'g'riligini tekshirishda ham qulay hisoblanadi.

VBScript

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) bilan. Shunga qaramay VBScript klient tomonida ishlovchi ssenariy hisoblanadi, ASP esa server tomonida ishlaydi.

Server tomonidagi senariylar.

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

Perl

Perl tili Web-illovalar yaratishda eng ommabop tillardan biri hisoblanadi. Matnlarni qidirish va taxrirlash, fayllar bilan qulay ishlay olish qoidalari bilan Perl tili Internet ning 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 Perl ning asosiy yutuqlaridan, 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 bajarishda juda 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 hisoblanib, imkoniyatlari va samaradorligi yuqori bo'lganligi bois ko'pgina kompaniyalar o'z vositalariga ASP ni qo'llab quvvatlash imkoniyatlarini kiritmoqdalar. ASP-vositalarini ishlab chiqish bo'yicha yirik kompaniya ChillsoftLider sredi nezavisimo`x proivzoditeley ASP-sredstv – kompaniyaChillsoft UNIX ning bir qancha turi va turli Web-serverlarda ASP ni qo'llash imkoniyatini kiritgan. Ko'pgina HTML-muxarrirlar, 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

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

JSP (JavaServerPage) 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

SSI (Server Side Include) vositasi dastlab HTML-faylni dastlab serverda qayta ishlaydi va undan so'ng uni klientga 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.

Server strukturasi.

Server - tarmoqqa ulangan va undan foydalanuvchilarga ma'lum xizmatlar ko'rsatuvchi kompyuter. Serverlar ma'lumotlarni saqlashi, ma'lumotlar bazasini boshqarishi, masalalarni masofadan qayta ishlashi, masalalarni bosib chiqarishi va boshqa bir qator vazifalarni bajarishi mumkin.

Ishchi stansiya - tarmoqqa ulangan shaxsiy kompyuter, foydalanuvchi shu orqali axborot resurslariga kirib boradi.

Tarmoqning ishchi stansiyasi ham tarmoq, ham lokal rejimida ishlaydi. U shaxsiy operatsion tizim (MS-DOS, Windows va hokazo) bilan ta'minlangan amaliy vazifalarni hal etish uchun foydalanuvchini barcha zarur vositalar bilan ta'minlaydi. Server turlaridan biri - fayl-serverga (File Server) alohida e'tibor berish kerak. Fayl-server tarmoqdan foydalanuvchilarning ma'lumotlarini saqlaydi va ularning ushbu ma'lumotlarga kirishini ta'minlaydi. Bu kompyuter katta hajmdagi tezkor xotiraga, katta hajmdagi qattiq diskka ega. U maxsus operatsion tizim boshqaruvi ostida ishlaydi. Fayl-server quyidagi vazifalarni bajaradi: ma'lumotlarni saqlash, ma'lumotlarni arxivlash, ma'lumotlar o'zgarishini sinxronlash, ma'lumotlarni uzatish. Ko'pgina vazifalarni bajarishda bitta fayl-serverdan foydalanish kamlik qiladi. Bu paytda tarmoqqa bir qancha fayl-serverlar qo'shilishi mumkin. Tarmoqdagi qurilmalarning o'zaro aloqasini boshqarish. Axborot tizimlari kompyuter tarmoqlari bazasida yaratilgan, u quyidagi vazifalar yechimini hal etadi: ma'lumotlarni saqlash, qayta ishlash, foydalanuvchining ularga kirishini ta'minlash va ma'lumotlarni qayta ishlash natijasini uzatish. Markazlashgan qayta ishlash tizimida bu vazifalarni markaziy EHM (Mainframe, Host) bajaradi. Kompyuter tarmoqlari ma'lumotlarni qayta taqsimlab, qayta ishlashni amalga oshiradi. Bu holatda ma'lumotlarni qayta ishlash ikki ob'ekt: mijoz va server o'rtasida taqsimlanadi. Mijoz (klient) - vazifa, ishchi stansiya yoki kompyuter tarmog'idan foydalanuvchi. Mijoz ma'lumotlarni qayta ishlash jarayonida qiyin ishlarni bajarishi, faylni o'qishi, ma'lumotlar bazasida axborot izlash uchun serverga so'rov jo'natishi mumkin. Ilgaritdan belgilangan server mijozdan tushgan so'rovni bajaradi. Server hamma foydalanadigan ma'lumotlarni saqlaydi ushbu ma'lumotlarga kirishni tashkil etadi va mijozga ma'lumotlarni beradi. Mijoz olingan ma'lumotlarni qayta ishlaydi va qayta ishlangan natijalarni foydalanuvchiga qulay ko'rinishda taqdim etadi. Bunday tizimlar uchun mijoz-server yoki mijoz-server arxitekturasini atamasi qabul qilingan.

Mijoz-server arxitekturasidan bir darajali lokal hisoblash tarmog'ida, shuningdek ajratilgan serverli tarmoqda foydalanish mumkin

Web server — foydalanuvchidan kelayotgan so'rovlarni qabul qilib, tegishli javobni jo'natadigan dasturdir. Foydalanuvchi sifatida veb brouzer(http, https protokoli orqali), mobil telefon(wap protokoli orqali), ba'zi dasturlar(antiviruslar) bo'lishi mumkin. Veb server orqali butun sayt boshqariladi, ya'ni sayt kataloglariga ruxsatlarni beradi, xavfsizlik jihatidan IP manzillarni blokka oladi, foydalanuvchilarni ro'yxatdan o'tkazish va boshqa ko'plab ishlarni amalga oshiradi.

Web serverni siz o'z uyingizda yaratishingiz mumkin. Uning uchun bitta kompyuter ajratib, unga biror veb dastur o'rnatasiz va bu dasturni yaxshilab sozlashingiz kerak bo'ladi. So'ng o'z saytingizni shu serverga joylashtirasiz va biror domen sotib olasiz. Bundan kelib chiqadiki, saytingiz ishlashi uchun siz ajratgan kompyuter doim yoqilgan holatda bo'lishi kerak.

Web serverni tushuntirish uchun bitta lirik chekinish, o'xshatish qilaman(qo'polroq bo'lishi mumkin. Tushuntirish uchun nimalar qilishmaydi-a).Biror saytni bitta somsa deb tasavvur qilamiz. Somsa ichidagi masalig'lar, saytning ichidagi ma'lumotlar(matn, rasm, video,..). Bu somsani biror bir inson pishirgan, demak pishirgan odam, veb dasturchidir(kimlardir somsa pishishiga yordam bergan bo'lishi mumkin, bular veb dizaynerlar, verstkachilar). Bu somsani sotish kerak, uning uchun joy topish kerak va ruxsat olish kerak. Bu o'xshatish domen va hostingga mos tushadi. Sotuvchi bo'lsa, veb serverdir, ya'ni sotishni boshqaradi(puli borlarga sotadi, hohlasa sotmasligi ham mumkin), hullas sotishni boshqaradi.

Web serverga o'rnatiladigan veb dastur turlari ko'pdir. Eng zamonaviy va eng ommabop dastur bu – albatta "Apache"dir. Apache Software Foundation korxonasi tomonidan chiqarilgan. Quyida ba'zi bir veb serverlarni nomini keltirdim.

- Apache – Apache Software Foundation.
- IIS – Microsoft mahsuloti hisoblanadi. Server operatsion tizimlariga o'rnatilgan(Windows Server 2003, Windows Server 2008,..).
- Nginx – ochiq veb server. Katta hajmdagi saytlar uchun.
- Lighttpd – ochiq veb server.
- Google Web Server – Google tomonidan ishlab chiqilgan veb server.
- Resin
- Cherokee – veb interfeys orqali boshqariladi.
- Rootage – Java tilida yozilgan veb server.
- THTTPD – oddiy, kichik hajmdagi, tez va xavfsiz veb server.

Web xosting.

«**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.»

Keling yanada oddiylashtirib, hostingni ko'p qavatli uy deb olaylik. Demak, sayt bu ko'p qavatli uyda joylashgan bir kvartira.Bundan kelib chiqsak domen o'sha kvartiraning manzili, hosting esa shu manzilda joylashgan kvartira. Kvartirada siz uchun zarur barcha narsalar mavjud bo'lsa, hostingda saytingiz uchun zarur barcha fayllari joylashadi.

Demak, hosting tanlashda nimalarga e'tibor berish lozim:

1. **Narxi.** Veb-dasturlashga endi kirib kelgan inson uchun bu muhim jihat desak yanglishmagan bo'laman. Ko'p hollar bu asosiy jihat desak yanglish bo'lmaydi. Bu masalada endi o'z hamyoningiz bilan kelishgan holda qaror qabul qilasiz.
2. **Do'stlar maslahati.** Deylik, biron do'stingiz yoki tanishingiz biron hostingdan foydalanmaslikni tavsiya etsa unga quloq solgan maqbul!
3. **Sifatli texnik qo'llab-quvvatlash.** Albatta, texnik yordam qanchalik tez va sifatli ko'rsatilishi ko'p narsani anglatadi. Shu sabab ushbu jihatga alohida e'tibor bering.

4. **Qulay admin-panel.** Hostingda barcha dasturiy taminot imkon qadar tushunarli va qulay bo'lishi lozim. Lekin, bu borada bir narsa deyish qiyin, chunki, menga yoqqa boshqaruv paneli kimgadir yoqmasligi mumkin. Qaysi boshqaruv panelini tanlash o'zingizga havola. Shaxsan menga cPanel yoqadi.
5. **Serverga tushadigan yuklanish chegarasi.** Saytingizga qancha ko'p inson kirsam serverga shuncha ko'p yuklanish tushadi. Ishni endi boshlayotganda bu haqida chuqur o'ylanish shart emas. Chunki, boshida saytingizga o'zingiz va bir-ikki tanishingizdan boshqa hech kim kirmaydi
6. **Trafik bo'yicha chegara.** Bu ham juda muhim hisoblangan jihat bo'lib, imkon qadar trafik bo'yicha chegara qo'yilmagan hositng tanlang.
7. **Turli chegirma va aksiyalar.** Balki, bundan narsalar juda unaqa ahamiyatga ega emasdir. Lekin, e'tibor bergan yaxshi. Misol uchun bazi hostinglarda hostingni bir yilga sotib olsangiz domen sovg'a.



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. *Internet texnologiya deganda nimani tushunasiz?*
2. *Qanday razmetkali tillarni bilasiz?*
3. *Klient-server texnologiyani qanday tushunasiz?*
4. *Internet tarmog'idagi Web-sahifalarni ko'rishimiz uchun qo'llaniladigan texnologiyalar?*
5. *Web-texnologiya klassifikatsiyasi?*
6. *Qanday teglar juft teglar deb ataladi?*
7. *Ssenariy nima?*
8. *Klient tomonida bajariluvchi ssenariylar nima yordamida bajariladi?*
9. *Server tomonida bajariluvchi ssenariylar nima yordamida bajariladi?*
10. *Gipersaloqalar nima?*

1.2-§. Web dizaynning paydo bo'lishi va rivojlanish bosqichlari. O'zbekistonda dizaynning rivojlanish tendensiyalari.



O`quv modullari

Web dizayn, Ozbekistonda web dizaynning web dizaynni rivojlanish tendensiyalari

Butunjahon tarmog'ida elektron sahifalar va saytlarning turli shakllari asosidagi web-sahifalar kundan kunga ko'payib bormoqda. Bu holat mamlakatimizda ham jadallik bilan rivojlanib bormoqda. Kimdir o'z shaxsiy sahifasini yaratsa, kimdir kompaniya va firmalar korporativ saytlarini, kimlardir esa ta'limga oid saytlarni hamda elektron o'quv materiallarini yaratmoqda. Yana kimdir o'z kundaliklarini, ya'ni bloglarini yaratish ustida ish olib bormoqda.

Shu bilan birga, yirik portallarning faoliyati ham rivojlanmoqda. Bularning hammasi Web saytlarni yaratish bilan bog'liq turli dasturlash tillari va dasturiy vositalari asosida yaratilmoqda. Web-sahifalarni yaratish bilan shug'ullanuvchi mutaxassislarning ba'zilar HTML tilida, ba'zilar PHP tilida, yana ba'zilar Web saytlarni yaratish va tahrirlash dasturiy vositalari yordamida, ishning mohiyatidan kelib chiqib ba'zi shu sohani mutaxassislari takomillashib borayotgan CMS (Content Management System) deb ataluvchi tizimlar yordamida web-saytlar, sahifalarni yaratmoqda.

Web saytlar yaratish tarixiga bir nazar

1989 yilda Tim Berners-Li tomonidan World Wide Web, ya'ni Bununjahon o'rgimchak to'ri ixtiro qilinganidan boshlab har bir foydalanuvchi oddiy bo'lsa-da, o'z Web-sahifasini 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 bosqishlarini (HTML-kodni yozishdan bu kodni serverga yuklashgacha) boshqarishni o'z zimmasiga oladi. Shu bilan birga, ko'pgina hollarda foydalanuvchi bu kodni bilishi shart bo'lmagan. Sababi bu dasturlarda Web-sahifalarni 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.

1994 yili PHP tilinig yaratuvchisi Rasmus Lerdorf o'zinig saytiga mehmonlar kirishini hisoblash uchun Perl dasturlash tilida maxsus qobiq yozib amalda qo'llagan. Ko'p o'tmay qobiqni ishlash unumdorligi juda past va sekinligi aniqlanganidan so'ng, dasturlarni yangidan "C" tilida yozib chiqishga to'g'ri keladi. Keyin, dastlabki dastur kodlari muallif tarafidan barchaga ko'rish uchun serverga nashr qilingan. Server foydalanuvchilari kodlar bilan qiziqib, uni ishlatish muxlislari ham paydo bo'lgan. Hademay, bu dasturlar alohida loyihaga aylanib, 1995 yilning iyun oyida dasturiy mahsulot PHP (Personal Home Page) nomi bilan birinchi nashri chiqarildi. Imkoniyatlari oddiy va sodda bo'lib: bir necha buyruqlarni tushunadigan kod analizatori (tekshiruvchisi), mehmonlar hisoblagichini, kitobini, chatini yaratish uchun foydali dasturlar to'plamidan iborat bo'lgan.

1996 yil aprel oyida dasturlar jiddiy qayta ishlanganidan so'ng, PHP/FI (Personal Home Page / Forms Interpreter) nomi bilan mahsulotning ikkinchi nashri paydo bo'ldi. Bu mahsulot HTML-kod ichiga yozilib, HTML-formalarni qayta ishlab, hozirgi PHP dasturlash tilining tayanch imkoniytlarini ichiga olgan. PHP/FI kod yozilishi Perl tiliga juda o'xshagan, lekin soddaroq bo'lgan.

1997 yili PHP/FI 2.0 nashri chiqdi. O'sha paytda bu mahsulot bilan dunyo bo'yicha bir necha ming odam foydalanib, taxminan 50 ming domen bo'lib, internetning 1%-ni tashkil qildi.

1997 yilda Endi Gutmans va Ziv Suraski PHP/FI kodini boshqatdan yozib chiqishdi, chunki eski kod ular ishlatayotgan elektron tijorat tizimlari uchun yaroqsiz edi. Eski kodning mualliflaridan yordam olish uchun ular birlashishni taklif etib, PHP3 nomli loyihani PHP/FI -ni rasmiy vorisi deb e'lon qilishdi. Yangi loyiha uyushgandan keyin PHP/FI loyihasi ishlab chiqarilishi to'xtatilgan.

PHP 3.0 ning eng kuchli taraflaridan biri uning kengaytirala olinadigan yadrosi (tizimning bosharuv qismi) bo'lib, bundan tashqari, ma'lumot jamg'armalar bilan, turli protokollar va interfeyslar bilan birgalikda ishlash keng imkoniyatlari yaratildi. Muvaffaqiyatga erishishga ancha ahamiyatli fakt bu yangi tilni boyligi va ob'ektlarga mo'ljallangan dasturlashni qo'llay olishi. Yangi loyiha bilan birga nafahat tilni tashqi, ichki tuzulishi o'zgardi, balki o'zini nomi ham. Endi PHP qisqartmasi "PHP:Hypertext Preprocessor" ma'nosini anglatishi bildirildi.

1998 yilning oxirida PHP foydalanuvchilarning soni o'n minglardan oshdi. Yuz mingdan oshiq web-saytlar bu tilni qo'llashini e'lon qilishdi. Taxminan Internetning 10% serverlarida PHP 3.0 o'rnatilgan edi.

1998 yilning iyun oyda PHP3 to'qqiz oy ommaviy tekshiruvdan keyin rasman e'lon qilindi. Shu yilning qishida Endi Gutmans va Ziv Suraski PHP yadrosini qaytadan ishlab chiqarishni boshlashgan. Ularning asosiy vazifasi PHP tiziminig unumdorligini ko'tarish va kodning modullarini yaxshilash edi. 1999 yilning o'rtalarida birinchi marta taqdim qilingan yangi yadro "Zend Engine" deb nomlangan ("Zend": mualliflar "Ziv" va "Andi" ismlardan tashkil topgan). Uning asosida tuzilgan yangi til PHP4 2000 yilning may oyida rasman chiqarilgan. Unumdorlik yaxshilangandan tashqari, PHP 4.0 muhim yangiliklarga ega bo'lib, sessiyalarni qo'llash, buferli chiqarish, kiritilgan ma'lumotlarni havfsiz qayta ishlash va yana bir necha yangi til tuzuvchilarini paydo bo'lishidan iborat.

Hozirgi kunlarda "Zend Engine" qayta yaxshilanib PHP5 tili ishlab chiqarildi. Asosiy o'zgarishlar ob'ektlarga mo'ljallangan dasturlash modelida bo'lib, tilning imkoniyatlari yanada kengaytirdi.

Hozirda PHP tilini yuz minglab dasturchilar ishatib, bir necha million saytlar yaratilib, Internetning 80% domenlarini tashkil etmoqda.

PHP tilining rivojlanish bosqichlari

Nashr	Sana	Izoh
PHP 1.0	1995 yil, 8 iyun	Rasmiy nomi "Personal Home Page Tools (PHP Tools)". Bu "PHP" qisqartmani birinchi marta ishlatilinishi.
PHP 2.0	1996 yil, 16 aprel	Web sahifalarni "Tez va oson ishlaydigan qurol" deb e'lon qilingan mahsulot.

PHP 3.0	1998 yil, 6 iyun	Bir kishi ishlab chiqaruvchi loyihadan ko`p ishlab chiqaruvchilar loyihasiga aylandi. Endi Gutmans va Ziv Suraski kodni boshqatdan yozib chiqishdi.
PHP 4.0.0	2000 yil, 22 may	Yangi yadro “Zend Engine” asosida tuzilgan
PHP 4.1.0	2001 yil, 10 dekabr	Til konstruktsiyasiga superglobal imkoniyatlar qoshilgan (\$_GET, \$_POST, \$_SESSION, etc)
PHP 4.2.0	2002 yil, 22 aprel	Xavfsizlikni oshirish maqsadida tizimning sozlash “Register_globals” parametri ochirilgan
PHP 4.3.0	2002 yil, 27 dekabr	“CGI” (Common Gateway Interface)ga qo`shimcha “CLI” (command line interpreter) texnologiyasi kiritilgan.
PHP 4.4.0	2005 yil, 11 iyul	phpize va php-config skriptlari uchun qo`llanma varaqlari qo`shilgan.
PHP 5.0.0	2004 yil, 13 iyul	Zend Engine II – yangi ob`ekt modeli kiritilgan.
PHP 5.1.0	2005 yil, 24 noyabr	Zend Engine II ishlash unumdorligi yaxshilangan.
PHP 5.2.0	2006 yil, 2 noyabr	Kengaytiradigan filtr modullarining PHP tizimida ishlatish imkoniyatlari ochiq holati o`rnatilgan (default).

PHP turli xil sistemalarda ishlay oladi. U sistema Windows, Unix ning ko`plab versiyalari, shuningdek Linux va xatto Macintosh bo`lishi mumkin. PHP ko`plab tarmoq serverlarida, xususan Apache, Microsoft Internet Information Server, Web Site Pro, Iplanet Web Server va Microsoft Personal Web Serverlarda ishlashi mumkin. Agar o`zimiz tuzgan dasturlarimizni Windows tizimida tekshirishni xoxlasak oxirgi sanab o`tilgan serverdan foydalanishimiz mumkin, xatto Apache serveri Windows sistemasi boshqaruvida ishlasa ham.

PHP interpretatori yordamida dasturni aloxida mustaqil ko`rinishda kompilyasiya qilish mumkin. U xolda dasturni mustaqil ishga tushirish mumkin. PHP tilini yaratishda ma`lumotlar bazasi bilan bog`lanish talablarini aloxida e`tiborga olingan. Ko`plab ma`lumotlar bazalarini PHP da o`qish mumkin. Masalan, bularga Adabas D, InternetBase, Golid, dBase, mSQL, Sybase, Empress,

MySQL, Velosic, FilePro, Oracle, Unixdbm, Informix va xokazolarni keltirishimiz mumkin. SHuningdek, PHP ODBC standartini ham o`qiy oladi.

Ajax texnologiyasi terminini Jesse James Garrent 2005-yil 18-fevraldagi “Web sahifalarga yangi yondashuv” nomli maqolasida birinchi bo`lib ishlatgan. Keyinchalik uni amalda qo`llagan. AJAX - Asynchronous Javascript and XML ya`ni asinxron JavaScript va XML.

Web texnologiyalari 2 turga bo`linadi.

1. Server

2. Klient

Klient qismdagi ayrim qulayliklarning ta`minlanishi ajax texnologiyasi bilan bog`liq ya`ni internet tezlik uncha yuqori bo`lmasa ham ajax texnologiyasi yordamida serverdan ma`lumotlarni olish mumkin. Agar php yordamida serverga murojaat qiladigan bo`lsak sahifalar qayta yuklanishi kerak bo`ladi. Bunda internet tezligi yuqori bo`lishi kerak va har bir harakat uchun sahifalar yangilanishini kutish kerak. Ajax texnologiyasi serverga qismga yordamchi qism hisoblanadi, ya`ni foydalanuvchining imkoniyatlarini oshiradi.

Ajax texnologiyasining eng muhim jihatlaridan biri bu sahifalarni qayta yuklamasdan serverdan ma`lumotlarni yuklab oladi.

Ajax ning 2 ta xususiyati bor:

Sahifalar qayta yuklanmasidan serverga borib amal bajarib qaytib kela oladi. Bunda ma`lumotlar JSON yoki XML formatida jo`natiladi va qabul qilinadi.

DHTML ya`ni o`zgaruvchan sahifalar yaratish, ya`ni javascript bilan ishlash.

MySQL Shvetsiyadagi MySQL AB firmasi tomonida yaratilgan bo`lib, 2008 yilning fevralida Sun Microsystems tomonida sotib olingan. Endlikda dastur kodini yaratishgda ushbu firma javobgar. MySQL Server ochiq, ya`ni tekin dasturiy ta`minot bo`lib, General Public License himoyasidadir ya`ni turli himoyalar qo`yilmagan (kalitlar, litsenziya). MySQL AB/Sun firmasi dastur kodiga mualliflik huquqiga egaligi sababli, ushbu dasturiy ta`minotning kommerzial turi ham bor. MySQL nomining kelib chiqishi aniq emas. 1996 yildan beri ko`plab biblioteka va tool (vosita)lar nomi oldida My prefiksi qo`yila boshlangan.

Bu dastur o`zi bilan quyidagi amallarni bajaradi, yani kompyuterlar ichki tizimi bilan aloqa o`rnatish va ma`lumotlarni takrorlash va ma`lumotlarning saqlanish joyini aniqlash.

Sayt xaritasini yaratishdek oddiy funktsiyani qo`shish uchun ham Web-yaratuvshilar tomonidan ancha-muncha mehnat talab etilar edi. Axir buning uchun barcha ilovalarni ko`rib chiqish, sahifalar 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.

Sayt yangilanishi muammosini, ya`ni yangilanishning avtomatlashtirishini hal qilish maqsadida yirik axborot kompaniyalari, jumladan, Newsweek va Time yangi dasturiy ta`minot yaratish ishlarini boshlashga majbur bo`ladi. Bu dasturiy ta`minotlar kontentni (ma`lumotlarni) boshqarish tizimlari (CMS – Content Management System) deb ataldi.

CMS—bu Content Management System dan olingan kengaytma, u soʻzma-soʻz tarjima qilinganda “Sayt kontentini Boshqarish tizimi” yoki “Saytni Boshqarish Tizimi” larni anglatadi. Baʼzida CMS ni saytning “Harakatlantiruvchisi” deb atashadi.

Oʻzbekistonda dizaynning rivojlanish tendensiyalari.

Hozirgi kunda Oʻzbekiston Respublikasida kompyuter va axborot texnologiyalarining rivojlanishi telekommunikatsiya tarmoqlarining modernizatsiyasi, maʼlumot uzatish va internet xizmatlarining keng qoʻllanilishiga imkoniyat yaratmoqda.

Respublikada talantli yoshlar ishtirok etadigan internet – festivallar, internet – forumlar, shaharlarda va qishloqlarda umumiy tarmoqqa ulanishlar miqdori oshib bormoqda, bundan kelib chiqadiki, maʼlumot uzatish xizmatlari kengaymoqda va ularning servis xizmatlari yanada intellektualroq boʻlib bormoqda.

Axborot texnologiyalarining hozirgi zamon taraqqiyoti xamda yutuqlari fan va inson faoliyatining barcha soxalarini axborotlashtirish zarurligini koʻrsatmoqda.

Chunki aynan mana shu narsa butun jamiyatning axborotlashtirilishi uchun asos va muxim zamin boʻladi.

Iqtisodiyotni globalashtirish sharoitida informasion madad istagan darajadagi tashkiliy xoʻjalik faoliyatining unumli rivojlanishiga, aholining axborotlashtirilganligi va farovonligini oʻsishiga va odamlar oʻrtasidagi oʻzaro munosabatlar yaxshilanishiga koʻmaklashuvchi muhim elementlardan biriga aylanib bormoqda. Bunday yutuqlar keng koʻlamli WWW tarmogʻi va unga joylashgan Web–saytlarning doimo ishlab turishi evaziga erishiladi. SHuning uchun hozirgi vaqtda har bir tashkilot yoki har bir shaxs, qaerda ishlashidan qatʼiy nazar, internetda oʻz joyini yaratish zarurligini tushunmoqda. CHunki bu usul natijasida butun dunyoga oʻzi toʻgʻrisida, oʻzining ish faoliyati toʻgʻrisida gapirish va internetning bepoyon resurslaridan oʻz maqsadi yoʻlida foydalanish imkoniyatlari paydo boʻladi. Axborotni Internetda joylash uchun, yuqorida aytilganidek, uni ekranda aks ettirish uchun qulay boʻlgan koʻrinishda, yaʼni Web–sahifa koʻrinishida taqdim etish zarur, chunki bu sahifada joylashtirishni xohlagan foydalanuvchi axboroti har tomonlama boʻlishi va u koʻp hollarda Web–saytda joylashishi mumkin.

Aql bilan yaratilgan sayt, qoida boʻyicha, informasion yagona bus—butun hisoblanadi va joylanajak axborot mohiyatiga qarab alohida kriteriya va standartlarga ega boʻladi, va koʻplab tashkilot va muassasalar, xususiyy foydalanuvchilar uchun oʻzlarining tashkilot va shaxsiy ish faoliyatlarida Web – texnologiyani va uning elementlarini tadbiq etish boʻyicha koʻplab har xil muammolarga duch kelinadi. Bu Web–saytlar internet serverlarida joylashguncha va nashr qilinguncha Web–texnologiya va dizayn asoslaridan yuqori va chuqur bilim talab qiluvchi, koʻpqirrali va professional–intellektual ishdir. Va bu muammolar zamonaviy Web–texnologiyalar boʻyicha mutaxassislar yetarlicha boʻlgandagina yechiladi. Shuning uchun Web–sahifa yaratish va uni oʻrganish jarayoni, shu berilgan yoʻnalish boʻyicha mutaxassislarni tayyorlash hozirgi kun

uchun juda muhim va faol hisoblanadi. Bu mutaxassislik kuchli va kelajagi porloq professiya, u hozir ham, kelajakda ham hamisha, o'z o'rniga ega bo'ladi

Boshqa tomondan Web–saytlarni yaratish va o'rganish muammosining faolligi va muhimligini quyidagi sabablar bilan asoslash mumkin:

- ✓ taqdim qilinayotgan internet servis xizmatlari muhitini ko'p miqdorda kengayishi va ulardan foydalanishning osonligi bilan;
- ✓ tarmoq Web–texnologiya, ularning resurslari va imkoniyatlarini hamma joyda tarqatish bilan;
- ✓ istagan darajali axborotlarga bo'layotgan talablarning o'sib borishi bilan;
- ✓ tashkilot va xususiy shaxslarning o'zlari va o'z ish faoliyatlari to'g'risidagi axborotlarni internetda joylash uchun intilishlarning o'sishi bilan;
- ✓ tarmoq ma'lumotlar bazasidagi global masshtabdagi har xil maqsaddagi axborotlarga e'tibor qaratish bilan;
- ✓ har mamlakat, millat va fan odamlarining o'zaro muloqot va tushunishlarga bo'lgan intilishi bilan;
- ✓ Web–texnologiya barcha yo'nalishlari bo'yicha mutaxassislarning yetishmasligi bilan.

Shunday qilib, internet xizmatlaridan ikkita xizmat spektriga alohida e'tibor berish kerak, ya'ni tarmoq abonentlararo axborot almashuvi va axborot qidirish va tarmoq berilganlar bazasidan foydalanish. Xizmat taraqqiyotining samarasi to'g'risida gap ketganda barchamiz har xil Internet xizmatlaridan potensial foydalanuvchiga aylanib qolishimizni ta'kidlash zarur.



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. *Internetning paydo bo'lish tarixi haqida nimalarni bilasiz?*
2. *www nima?*
3. *Internet asoschisi haqida nimalarni bilasiz?*
4. *O'zbekistonda internetni rivojlanishi tarixini bilasizmi?*
5. *Web texnologiya nima?*

1.3-§. Web dizayn texnologiyalari: saytning mantiqiy va fizik strukturasi

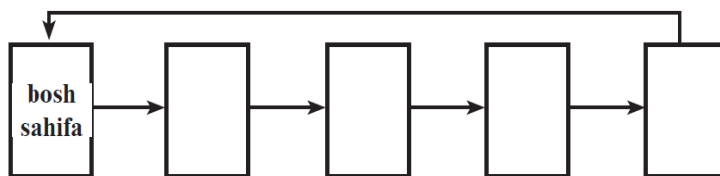


O'quv modullari

katalog tuzilishi, navigatsiya tuzilishi, navigatsiyaning matnli sistemasi, tugmalar (knopki), navigatsiya xaritalari.

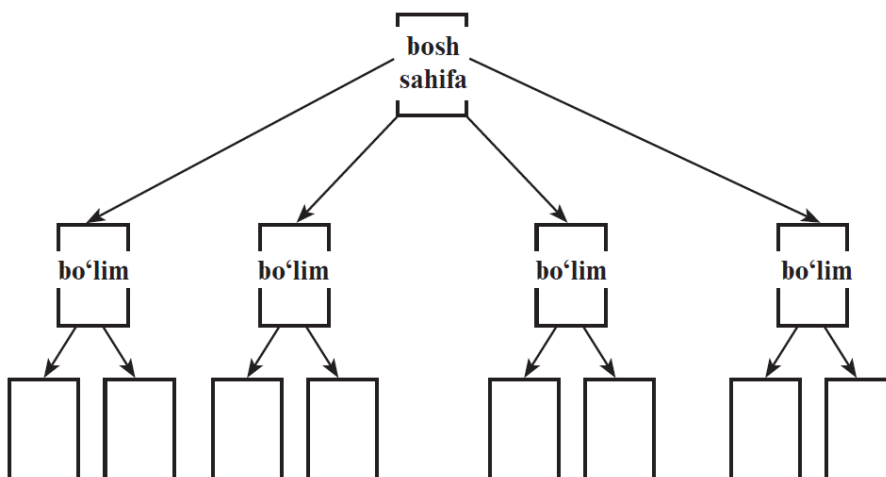
Saytning umumiy tuzilishi. Saytning tuzilishini rejalashtirganda bir nechta asosiy narsalarni albatta o'ylab ko'rish kerak: katalog tuzilishi, navigatsiya tuzilishi, sayt zastavkasining zarurligini.

Katalog tuzilishi. Eng avvalo sizning loyihangiz ichidagilarni tartibga solish kerak. Hamma rasmlarni bitta papkaga ish fayllarini boshqasiga joylashtiring. Sahifalarning o'zini ham bo'lsa bo'ladi (masalan, bo'limlarga). Shunday qilsangiz, o'zingizga oson bo'ladi. Buning uchun kelajakdagi saytingizning tuzilishini aniqlashtirib olishingiz kerak. Bir necha asosiy tuzilishlar mavjud:

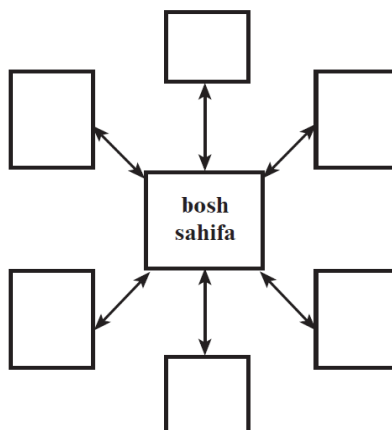


- *chiziqli* – betlar berilgan tartibda joylashtirilgan. Bir betdan ikkinchi betga o'tish qat'iy teglangan. Bunday tuzilish oson va tartibi bilan o'qishga asoslangan. Betlarni berilgan tartibda joylashtirishda, amin bo'ling-ki, foydalanuvchi kerakli ma'lumotlarning birortasini chetda qoldirmaydi:

- *ierarxik*– betlar toifaga va toifa bo'lakchalariga bo'lingan. Bunday tuzilish juda ham qulay:



- *erkin*– betlar bemaolol tartibda joylashgan. Bunday tuzilish faqat uchta katta bo'lmagan saytlar uchun:



Fayl nomlari haqida bir necha soʻzlar: ular harflardan, siz bergan ramzlardan iborat boʻlib, bosh harflarga ega boʻlishi shart emas.

Navigatsiya tuzilishi. Bu resursni rejalashtirishda eng asosiy qism boʻlib, saytdagi axborotga kirishi oddiy va tushunarli boʻlishi kerak. Foydalanuvchi xohlagan vaqtda kerakli savollarga javob berishi kerak:

1. Men qayerda turibman?
2. Qayerga borishim kerak?
3. Qanday qilib u yerga borsa boʻladi?
4. Orqaga qanday qaytsa boʻladi?

Bunday masalalarga javob topish uchun har xil variantlardan foydalaniladi. Masalan, qanday qilib «menyu» sahifasidan foydalanuvchini topish mumkin va menyu oʻzini qanday qilib hamma sayt sahifalarida koʻrinarli qilishi mumkin.

Yana shu bilan birga navigatsiya kalitlarini ham qilish mumkin, yaʼni har bir sahifada sxema boʻyicha shu sahifaga toʻliq yoʻlni koʻrsatish: bosh bet – boʻlim nomi – sahifa nomi.

Umuman olganda, 3 klik qoidaga rioya qilish kerak, chunki foydalanuvchi saytning xohlagan sahifasiga oʻtishda kamida 3 klikda yetib borishi kerak.

Navigatsiyaning asosiy sistemalari:

- *Navigatsiyaning matnli sistemasi* – eng koʻp tarqalgan turlaridan biri. Aytish kerak-ki, boshqa navigatsiya sistemalaridan foydalangan taqdiringizda ham navigatsiya matni qatnashib turishi kerak. Umuman aytganda, bunday navigatsiya matnli joʻnatmani har xil usullar bilan bezatishni aks ettiradi.

Gorizontal va vertikal menyularga boʻlinadi. Vertikal har bir varaqning oʻng yoki chap tomonida joylashadi. Gorizontal menyu – sayt shapkasining tagida joylashgan. Qulaylikni yaratishda har bir sahifaning tagida gorizontal menyuni nusxa qilishni maslahat beriladi.

- *Tugmalar (knopki)* – chastota boʻyicha ishlatish ikkinchi navigatsiya elementlari boʻlib keladi. Tugmalar har xil oʻlcham, shakl, rang va uslubda boʻlishi mumkin. Ularni ishlatishda har bir voqeani tahlil qiluvchi (masalan, JavaScript)ga ulash kerak boʻladi va shunda har bir voqeani tugmani bosishda yoki boʻlmasa sichqonchani boshqarishda tavsiflab beradi.

Eng optimal qarorlardan biri: tugma turini qoldirish (fonda), matnni esa undagi joʻnatma turida bezatish kerak. Xulosa shuki, CSS vositalari bilan bunga erishish qiyin emas.

- *Navigatsiya xaritalari.* Bunday kirishishda bitta tasvir olinadi va ularning har xil sohalariga joʻnatma berkitiladi. Eng keng tarqalgan misollardan biri bu jahon turistik xaritasi, tugmani bosa

siz, qaysi mamlakat boʻlishidan qatʼi nazar va shu mamlakatga bagʻishlangan sahifaga tushasiz. Sahifaning tagida uncha koʻzga koʻrinmaydigan gorizontal menyuda xaritani albatta nusxa qilish kerak. Boʻlmasa, arzimagan sabablar bilan foydalanuvchi hech qayerga kirolmasligi mumkin.

Masalan, hammaning munosabati har xildir: birlari buni qoʻllab-quvvatlaydilar, boshqalarga bu yoqmaydi.

Shuning uchun saytning zastavkasini qiladigan bo'lsangiz, yuklamani o'tkazib yuborishni yoki roliklarni ko'rish xohishi bo'lmaganlarga bu funksiyani tadbqiq etishni unutmang.

Saytning asosiy sahifasi.Odatda bu foydalanuvchi tushadigan birinchi sahifa. Shuning uchun ba'zi bir shartlarni qoniqtirishi kerak:

- bosh sahifa saytning tematikasini aks ettirishi kerak;
- u qiziqarli bo'lishi kerak, sababi foydalanuvchida saytning keyingi sahifalariga kirish xohishi uyg'onishi kerak;
- iloji bo'lsa, sayt bo'yicha qidiruv funksiyasini joylashtirish kerak;
- unda axborotni yangilab turadigan bo'limlar bo'lsin: masalan, yangiliklar, aksiya, maslahat kuni;
- qayta aloqa usuli mavjud bo'lishi kerak (e-mail, telefon, manzil);
- bosh sahifani boshqa sahifadan sayt umumiy uslubiga javob beradigan qiling.

Saytning boshqa sahifalari.Saytning hamma sahifalar tashqi turi butun bir uslubda bo'lishi kerak. Ularda resursning nomi va bosh sahifaga jo'natmasi ko'rsatilishi kerak. Har bir sahifa o'ziga xos maftunkor, ya'ni kerakli ma'lumotga ega bo'lishi kerak. Bunda bir nechta kerakli daqiqalarni hisobga olganda:

1. Sahifaning uzunligini kuzatish kerak. Agarda u 2,5 ekrandan katta bo'lsa, uni ikki sahifaga bo'lish kerak;
2. Matnni xat boshilarga bo'lish kerak, ortga va maydon. Matnni sarlavha va sarlavhachalarga ajrating. Shunday qiling-ki u tez va oson qabul qilinishi kerak;
3. Matnni rasm va grafika bilan ta'minlang, faqat bachkana bo'lib ketmasin va hammasi dalili bilan bo'lishi kerak.



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. Saytning mantiqiy va fizik strukturasi deganda nimani tushunasiz?
2. Navigatsiya nima va u nima uchun zarur?
3. Sayt dizayninin loyihalashda nimalarga e'tibor berish kerak?
4. Saytning asosiy sahifasi qanday bo'lishi kerak?

1.4-§.Saytning statik va dinamik komponovkasi

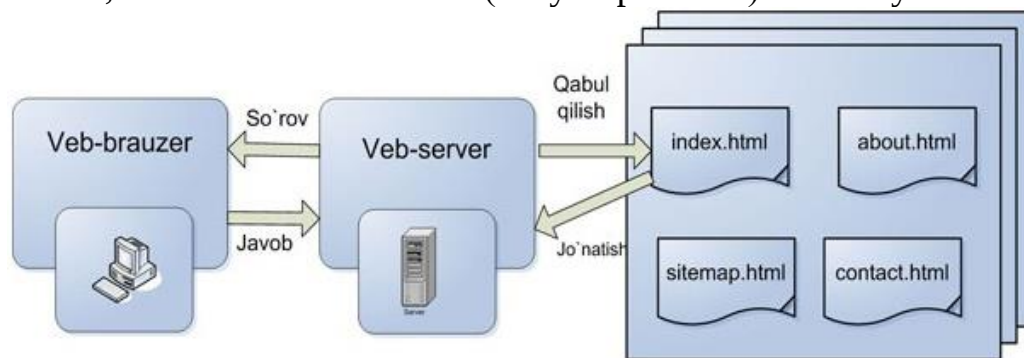


O`quv modullari

web dizayn, O'zbekistonda web dizaynning web dizaynni rivojlanish tendensiyalari

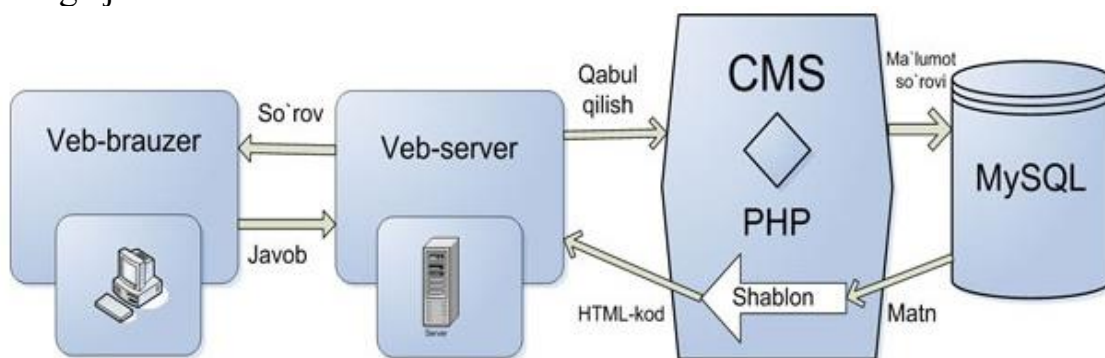
Oddiy sayt qanday prinsip asosida ishlashini ko'rib chiqamiz. Foydalanuvchi biror-bir, masalan, page.html sahifasiga murojaat qildi deylik. Veb-server o'zidagi

Veb-sahifalar katalogini ko`rib shiqadi va agar page.html bo`lsa, foydalanuvchiga uni ko`rsatadi, aks holda esa – 404-chi (“Fayl topilmadi”)xabarni yetkazadi.



1.4.1-rasm. Web-serverning statik sayt xolida ishlashi

Dinamik sayt esa biroz boshqacha ishlaydi. Foydalanuvchi biror sahifaga 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) qaratilgan bo`ladi. Dvijok foydalanuvchiga kerakli ma`lumotni biror manbadan (masalan, ma`lumotlar bazasi yoki diskdagi fayldan) oladi, uni HTML-sahifaga aylantiradi va Veb-serverga uzatadi, u esa o`z navbatida, foydalanuvchi brauzeriga jo`natad.



1.4.2-rasm Web-serverning dinamik sayt bilan ishlashi

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 yuz sahifalik oddiy sayt yaratdingiz. Ertami-kech siz bu sayt dizaynini, ko`rinishini o`zgartirmoqchi bo`lasiz. U holda esa siz shu yuz sahifani qo`lda o`zgartirib chiqishingizga to`g`ri keladi. Ustiga-ustak, siz avvaliga bu yuz sahifani kompyuteringizdagi lokal serverga yuklab olishingiz, undan so`ng tahrirlab, kerakli o`zgartirishlarni kiritishingiz va oxirida yana serverga yuklashingiz kerak bo`ladi.

Ana endi dinamik sayt qanday ishlashini ko`rib shiqaylik. Asosiy voqea – “HTML-sahifani 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 shabloni fayllarini o`zgartirishingiz kerak.

CMS tizimi sahifalarni dinamik ko`rinishga keltiradi: ular foydalanuvchi sahifani ko`rmoqchi bo`lgandagina amalda yaratiladi. Bu degani, sahifalar foydalanuvchi tomonidan bevosita ko`rilayotganda yangilanishi yoki o`zgartirilishi mumkin.

Bu tizimlarning asosiy jihati shundaki, dinamik web-sahifalarni oson yarata olish va ularni turli xil axborot bilan to`ldirish imkoniyatini berishdir. CMS tizimlari odatda murakkab bo`lib, ular o`z ishiga xabarlarni tarqatish xizmatlari (RSS), forumlar va internet-do`konlarni oladi. SHu bilan birga, ularni o`zgartirish ham oson.



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. *Statik va dinamik saytlar qanday farqlanadi?*
2. *Statik saytning afzalliklari va kamchiliklari qanday?*
3. *Dinamik sayt afzalliklari va kamchiliklari qanday?*
4. *Dinamik sayt yaratish qanday bosqichda amalga oshiriladi?*

1.5-§. Web-saxifa elementlari



O`quv modullari

sarlavha, logotip, qidiruv, reklama banner, kontent, navigatsiya elementlari, tashriflar hisobi

Har qanday web-sahifa har bir Internet-resursning zarur komponentlari bo`lgan standart elementlarning maxsus majmuini o'z ichiga oladi. Ular saytning turi va maqsadidan kelib chiqib sahifaga joylashtirilishi mumkin. Web sahifalarda eng kamida quyidagi elementlar mavjud bo`lishi mumkin:

1. *Sarlavha/Logotip*
2. *Qidiruv*
3. *Reklama banneri*
4. *Kontent/asosiy matn*
5. *Navigatsiya elementlari*
6. *Saytni ishlab chiquvchi haqida ma'lumot*
7. *Tashriflar hisobi*

Albatta, bunday ob'ektlarning ko`lami va soni saytning tematik markaziga, sizning hoxishingizga qarab farq qilishi mumkin! Ushbu elementlar yaratuvchi tomonidan belgilanadigan maqsadlar bo'yicha aniqlashtiriladi. Bunday elementlarning yig'ilishi, ularning o'zaro kelishuvi dizayni web dizaynerning asosiy vazifalaridan biri hisoblanadi.

Biz ko'rib chiqmoqchi bo`lgan veb-sahifaning birinchi elementi uning nomi. Har ikkala holatda ham hujjatning yuqori qismida joylashgan bo'lishi kerak, ammo matn va grafik shaklda bajarilishi mumkin.

Saytning logotipi yoki nomi tahskilotning kirish qismidagi belgiga o'xshash vazifani bajaradi. Do'konga borganimizda, darvoza oldida belgilarni ko'rish kifoya. Internetda har bir sahifada ushbu yozuvni ko'rish kerak.

Biz saytning logotipi sahifaning yuqori qismida, odatda chap yoki o'ng tomonda bo'lishini kutamiz. Buning sababi, logotip butun saytni bir butun sifatida aks ettiradi va shuning uchun saytning mantiqiy ierarxiyasidagi eng yuqori pozitsiyani egallaydi. Har bir sahifaning ierarxiyasida siz ushbu dominant pozitsiyani ikki shaklda tasvirlashingiz mumkin: logoni sahifadagi eng ko'zga ko'ringan ob'ekt sifatida yaratib qo'ying yoki sahifaning mazmuni uchun ramka vazifasini bajaradigan qilib qo'ying.

Har bir sayt samaradorligini oshiruvchi asosiy element bu qidiruv tizimidir. Qidiruv tizimining mavjudligi va mukammaligi sayt reytingini oshiradi. Ma'lumotlarni oson topish imkonini beradi.

Sayingizda zaruriy bo'lmagan element reklama bannerlaridan foydalanishingiz mumkin. Bu saytni moliyaviy qo'llab quvvatlanishini ta'minlaydi. Saytning asosiy mazmuni content qismida aks etadi. Unda barcha zaruriy ma'lumotlar talab bo'yicha joylashishi mumkin.

Web-sahifaning keyingi majburiy qismi- ushbu hujjatni saytning boshqa bo'limlari bilan bog'laydigan hiperishoratlar - navigatsiya elementlaridir. Navigatsiya elementlari matn satrlari, grafik ob'ektlar, ya'ni tugmalar yoki Java ilovalari kabi faol komponentlar shaklida bo'lishi mumkin.

Eng yaxshi tashkil etilgan yondashuv navigatsiya elementlarini sahifaning chap chegarasida yoki yuqori qismida joylashtirishdir.

Hujjatning pastki qismida sayt egalari va e-pochta manzillari haqidagi ma'lumotlarni nashr etilishi odatiy bo'lib, u yerga tashrif buyuruvchilar sayt egalari javoblarini, taklif va takliflarini yuborishi mumkin.

Agar web-sahifa boshlang'ich hujjat bo'lsa, uning pastki qismida mehmon hisoblagichi joylashtiriladi, bu hisoblagich ko'rsatkichini o'zgartirib, foydalanuvchi brauzerida hujjatning har bir ochilishini qamrab oladi. Buning yordamida web dasturchi o'zining sahifasini tashrif buyurgan mehmonlar sonini osongina aniqlashi mumkin. Hisoblagich taymeri faqat saytga kirish paytida chaqirilgan birinchi sahifada o'rnatiladi. Xuddi shu sahifada bir nechta taymerni joylashtirish tavsiya etilmaydi.



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. *Web sahifa qanday elementlardan iborat?*
2. *Sarlavha nima?*
3. *Saytda qidiruv tizimi ahamiyati qanday?*
4. *Sayning foydalanuvchilarni ro'yxatga olish tizimining ahamiyati qanday?*

2-BOB. HTML. ASOSIY TUSHUNCHALAR.

2.1-§. Hujjat tuzilishi. HTMLning asosiy teglari.



O`quv modullari

hujjat tuzilishi, HTML, teg, HTMLning asosiy teglari, SublimeText

HTML haqida asosiy tushunchalar. HTML (*HyperText Markup Language*) – kompyuter ekranida ma'lumotlarni ma'lum bir shaklda aks ettirishga mo'ljallangan hujjatlarni belgilash tili.

Hujjatni yaratishda matninng qaysidir qismini qalin, kursiv, o'lchami kata yoki rangli qilib ajratishga, matnni markazdan to'g'rilashga to'g'ri keladi. Buning uchun matn muharririda yetarlicha uskuna va imkoniyatlar mavjud hamda uni formatga tadbiq etish mumkin. Maslan, matnni kursiv qilish uchun uni belgilash va **Kursiv** tugmasini bosish kerak bo'ladi. HTML tilida bunga quyidagi kodni yozish bilan erishish mumkin: `<i>Matn</i>`

`<i>` belgisi matn belgilanishini boshlangan joyni bildiradi `</i>` belgisi esa belgilangan fragment tugaganini bildiradi.

`<i>` va `</i>` belgilarini *teglar* deb atash qabul qilingan.

Teg – HTML element, brauzerga ko'rsatgan holda rasm, animatsiya va matnning web sahifasining qayerida joylashishini, matnni qanday – katta, qalin, kichkina, kursiv bilan yozilishini, qayerdan yangi satrni boshlab, qayerdan tugashini ko'rsatib beradi.

HTML tili teglari ramzlar bilan ajraladi: chegaralovchilar (`<b>`), bunda tegning identifikatori (nomi) parametri yoziladi (bizning misolimizda bu b). Bu qoidaning yagona inkori bu qiyinroq chegaralanuvchi (`<!--va-->`) tegi izohlaridir. Teglar nomini va ularning parametrini xohlagan registrda yozib olish mumkin. HTML teglarining ko'pi juft bo'lib qo'llaniladi, ya'ni ochuvchi teg va yopuvchi teglar HTML qoidasiga binoan yopiq teglar singari yozib olinadi, teg nomidan oldin faqat ramzi bilan juftli teglarning yagona prinsipial farqi shundaki, yopiq teglar parametrlarni qo'llamaydilar.

Asosan, hamma teglar juftli, ya'ni biri ochuvchi `<`, boshqasi yopuvchi `>`. Bunday juftlik teglar konteyner deyiladi. Uningo'rtasidagi joylashgan boshqa hamma narsalar konteyner ma'lumotidir. Teglar harakati faqat ularning ichidagi narsalarga qo'llaniladi. Ba'zida tugallangan tegni tushirish mumkin.

Teglar yordamida hujjatning barcha strukturasini tavsiflanadi. Teglar burchakli qavslar `"<"` va `">"` ichiga yozilgan so'zlar bilan belgilanadi. Juda ko'p teglar juft teg hisoblanib, ochiluvchi (`<i>`) va unga mos yopiluvchi (`</i>`) teglardan iborat bo'ladi. Yopiluvchi tegi ochiluvchi teg oldidan (`"/"`) belgisini qo'yish bilan farqlanadi. Shuningdek, yopish tegi mavjud bo'lmagan teglar ham mavjud, masalan, yangi satrga o'tish tegi `<br>`.

Ba'zi teglar parametrlarga ega bo'lishi mumkin (ba'zan ular atributlar deb ataladi). Parametr q "qiymat" formatidagi bo'sh joydan keyin teg nomidan keyin ko'rsatiladi.

Agar bir nechta parametrlar mavjud bo'lsa, ular bo'shliq bilan ajratiladi.

Masalan:

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

Ushbu misolda <meta> tegining http-equiv parametri Content-Type ga o'rnatiladi va content parametri text/html ga o'rnatiladi; charset=utf-8.

Teglar bir-birining ichiga ya'ni ichma-ich joylashishi mumkin.

Masalan:

```
<p><i> Matn </i></p>
```

Teglar joylashtirilganda ularni yopish tartibiga rioya qilinishi kerak.

Misol uchun, ushbu kodni xato hisoblanadi:

```
<p><i> Matn </p></i>
```



HTMLda teglar va parametrlarning nomlari katta va kichik harflarda yozilishi mumkin. XHTML tilida faqat kichik harflar bilan yozilishi kerak.

HTMLda tayyorlangan hujjatlarni web-brauzer deb nomlangan maxsus dasturlar yordamida ko'rish mumkin. Web-brauzerlar hujjatning strukturasi tavsiflaydigan manba kodiga asoslanib hujjatlarni ko'rsatib beradi. Web-brauzer oynasida ko'rsatilgan HTML hujjatining talqini natijasiga web-sahifa deb ataladi. HTML hujjatidan farqli o'laroq, web-sahifa faqat matnni emas, balki grafiklarni, videolarni, ovozlarni ham o'z ichiga olishi mumkin. Shuningdek, web-sahifada birdaniga bir necha HTML-hujjatlar natijasini ko'rsatishi ham mumkin. HTML formatidagi hujjatlar kengaytmasi **html** yoki **htm** ko'rinishida bo'ladi.

Hujjat tuzilishi. Birinchi HTML-hujjatimizni yaratib ko'ramiz. Uni yaratish uchun ixtiyoriy matn muharriridan foydalanish mumkin. Eng keng tarqalgan muharrir oddiy Bloknot hisoblanadi va unda 2.1.1-misol matnini kiritamiz.



2.1.1-misol. Birinchi HTML-hujjat

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title>Sahifa sarlavhasi</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
</head>
```

```
<body>
```

```
<p>
```

```
<strong>
```

Bu matn qalin shiriftda ko'rinadi

```
</strong>
```

```
</p>
```

</body></html>

Biz kiritilgan matnni HTML formatida, masalan, 2.1.1-misol.html nomi bilan saqlaymiz. Buning uchun:

- **Fayl** menyusidan **Soxranit kak** tanlanadi.
- Ochilgan oynada **Imya fayla** satriga "test.html" kiritiladi
- **Tip fayla** ro'yxatidan **Vse faylo`** banditanlanadi.
- **Soxranit** tugmasi bosiladi (Masalan, Rabochiy stol tanlanadi). Bloknot dasturini yopiladi.

Endi esa saqlangan test.html fayni ochish uchun, saqlangan faylni topib uni brauzerda ochilishini tanlash kerak masalan: **GoogleChromeda**



2.1.1-rasm

Har bir ketma-ketlik to'g'ri amalga oshirilsa, yaratilgan html fayl Web brauzer oynasida ko'rsatiladi va sarlavha satri "Sahifa sarlavhasi" ko'rinishida bo'ladi. Web-brauzer oynasidagi teglar ko'rsatilmaydi! Endi web-brauzer oynasida sarlavha satrini o'zgartirishni harakat qilib ko'ring. Buning uchun siz HTML formatida manba matnini ochishingiz kerak. Bu uchta usulda amalga oshirilishi mumkin:

- 1) View menyusida View HTML code ni tanlang;
- 2) Web brauzerining oynasida biron-bir joyda sichqonchaning o'ng tugmasini bosing. Ochilgan kontekst menyusidan View HTML- kodini tanlang;
- 3) **Bloknot** yoki boshqa matn muharriri yordamida sahifa kodini o'z ichiga olgan faylni oching. Bu usul eng universaldir. Natijada hujjat matn tahrirlash uchun ochiladi. Endi ushbu qatorni o'zgartiramiz.

<title> Sahifa sarlavhasi</ title>

<title> Mening birinchi web sahifam </ title> ga o'zgartirib faylni saqlang (**Fayl** menyusi, **Saqlash**).

Endi web-brauzeriga qaytib, web-sahifani yangilang. Buni quyidagi yo'llar bilan yangilashingiz mumkin:

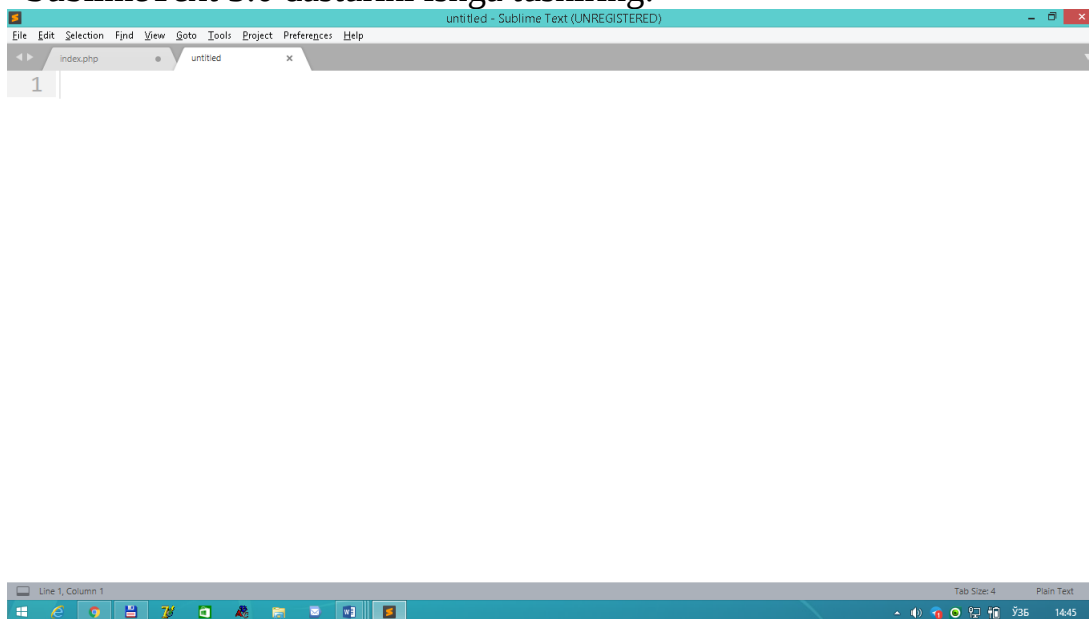
- View menyusida Refresh bo'limini tanlang;
- kontekst menyusida Refresh bo'limini tanlang;
- **Paneli instrumentov** bo'limidan **Obnovit** tugmanisini bosing;
- klaviaturadan<F5> tugmasini bosing.

Shunday qilib, sarlavha satri "Mening birinchi web-sahifam" ga o'zgaradi. Hujjat kodidagi biror narsani o'zgartirib, amalga oshirilgan harakatlar natijalarini qaytadan takrorlab ko'ring. Algoritm quyidagicha: manba kodini oching, sozlashni qiling, saqlang va web-sahifani yangilang.

Hozirgi kunda SublimeText 3.0 dasturi web hujjatlarni taxrirlasahning juda yaxshi dasturlaridan biridir. U HTML, JavaScript, PHP va shu kabi sintaksislarni yoritib boradi. SublimeText 3.0 dasturini uning sahifasi

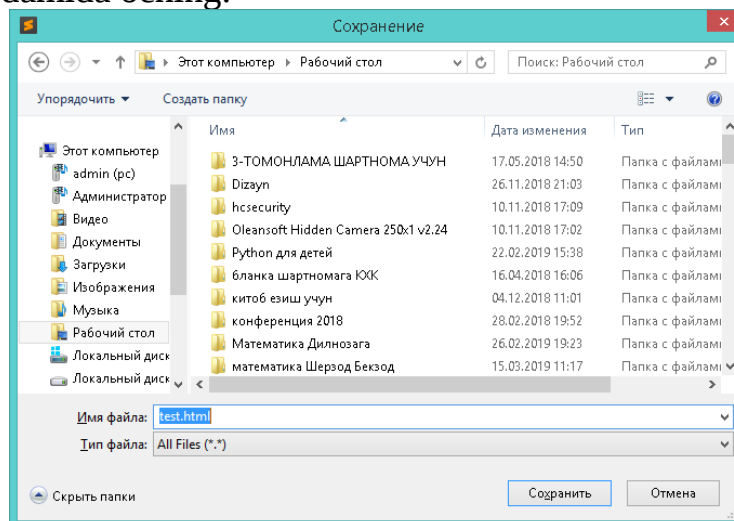
(<https://www.sublimetext.com/>) dan bepul yuklab olishingiz mumkin. Uni o'rnatish juda oddiy va sharhlarga ehtiyoj yo'q.

SublimeText 3.0 dasturini ishga tushiring.



2.1.2-rasm

2.1.1-misolda ko'rsatilgan kodni yozing va fayl menyusidan **Save (CtrlQS)**ni tanlang. Ochilgan oynada **Fayl** nomi satriga "**2.1.1-misol.html**" ni kiriting. Papkani tanlang, masalan, Ish stoli va **Save** ni bosing. Ko'rish uchun faylni web-brauzer yordamida oching.



2.1.3-rasm

Faylni tahrirlash uchun Fayl menyusida Open File ni tanlang. E'tibor bering SublimeText 3.0 o'rniga siz NotePadQQ, PHPStorm, WebStorm, Aptana Studio yoki NetBeans tahrirlovchilaridan foydalanishingiz ham mumkin.

Shunday qilib, biz HTML-hujjatlarni yaratish texnologiyasini o'rganib chiqdik, hujjat kodini saqlash, namoyish qilish va o'zgartirishni o'rgandik. 2.1.2-misolda har qanday HTML hujjatining umumiy strukturasi keltirilgan.



2.1.2-misol.HTML hujjatning tuzilishi

```
<!DOCTYPE html><!--Hujjat formati deklaratsiyasi-->
<html>
<head>
<!-- Hujjat haqida texnik ma'lumotlar-->
</head>
<body>
<!-- Hujjatning asosiy qismi -->
</body>
</html>
```

<!DOCTYPE> tegi web-brauzer uchun fayl formatini belgilab beradi va uning barcha ko'rsatmalarini o'z ichiga oladi.

HTML hujjatining barcha matnlari <html>va</html> teglari orasida joylashgan bo'ladi. HTML hujjat ikki qismdan iborat: bosh (<head>va</head> teglari orasidagi) va tarkib qismlari (<body> va </body> teglari orasidagi).

Nazorat uchun savollar

1. HTML (Hyper Text Markup Language) tili qanday til?
2. HTML ning asosiy qoidalari nimalardan iborat?
3. HTML–fayl deganda nima tushuniladi?
4. HTML– faylning kengayishi qanday?
5. Teglar, deb nimaga aytiladi?
6. HTML– hujjat strukturasi nimadan iborat?
7. Sarlavha va abzaslarni formatlashtirish qanday bajariladi?

2.2-§. HEAD bo'limi. Hujjatning texnik tavsifi



O`quv modullari

*head, **body**, hujjatning texnik tavsifi, teg xususiyatlari*

HEAD bo'limining asosiy ma'lumotlari - nom, uning tavsifi va qidiruv tizimi uchun kalit so'zlar, muallifga tegishli ma'lumotlar va sahifa yaratish vaqti, sahifaning asosiy manzili, kodlash va h.k hisoblanadi.

HEAD bo'limidagi yagona majburiy teg <title> tegi ishlatiladi.<title> va </title> o'rtasidagi matn web-brauzerining yuqori qismida ko'rsatiladi. Sarlavha uzunligi 60 belgidan oshmasligi kerak, aks holda web-brauzerining boshiga to'liq mos kelmaydi:

```
<title> Sarlavha satri</title>
```

Maslahat. Ko'pincha <title> va </title> belgilari orasidagi matn ushbu sahifaga havola matn sifatida qidiruv tizimi natijalarda ishlatiladi. Shu sababli, sarlavha sahifaning mazmunini iloji boricha to'liq tavsiflashi kerak. "Bosh sahifa", "Birinchi sahifa" kabi narsalarni yozmang.

Bitta <meta> tegidan foydalanib, qidiruv tizimi uchun sahifa tarkibi va kalit so'zlarni tavsifi belgilashingiz mumkin. Agar <title> va </title> belgilaridagi matn bu sahifa uchun havola matni sifatida ishlatilsa, <meta> tegidagi tavsif quyidagi havola bilan ko'rsatiladi:

<meta name q "description" content q "Sahifaning mazmuni tavsifi">

<meta name q "keywords " content q "Vergulli ajratilgan kalit so'zlar ro'yxati">

Bundan tashqari, turli tillarda bir nechta tavsiflarni ham ko'rsatishingiz mumkin. Buning uchun **lang** parametrda ishlatiladigan tilni tanlang:

➤ <meta name q "description" lang q "uz" content q "Sahifa tarkibi tavsifi">

➤ <meta name q "description" lang q "en" content q "Description">

➤ <meta name q "keywords" lang q "uz" content q "Kalit so'zlar vergul bilan ajratib yoziladi">

➤ <meta name q " keywords " lang q "eng" content q " keywords ">

Bundan tashqari, <meta> taglari web-sahifani qidirish mashinalari tomonidan qidirishda indeksasiyalashga ruxsat berish yoki taqiqlashga imkon beradi:

<meta name q "robots" content q "<index>, <havola orqali o'tish>">

content parametrlari quyidagi qiymatlarni qabul qilishi mumkin:

➤ index – qidiruv tizimiga ruxsat;

➤ noindex - qidiruv tizimiga taqiq;

➤ follow - bu web-sahifaga havola bo'yicha o'tishga ruxsat;

➤ nofollow - bu web-sahifaga havola bo'yicha o'tishga taqiq;

➤ all – index va follow birgalikda yoqilgan. Odatiy holda shunday bo'ladi;

➤ none- noindex va nofollow birgalikda.

Bir qator misollar keltiramiz.

Indeksasiyalanish va havola orqali o'tishga ruxsat beriladi:

<meta name q "robots" content q "index, follow">

Indeksasiyalanish va havola orqali o'tish taqiqlanadi:

<meta name q "robots" content q "index, nofollow">

Indeksasiyalanish va havola orqali o'tishga ruxsat berilmaydi:

<meta name q "robots" content q "noindex, nofollow">

Bundan tashqari, <meta> yorlig'ini ishlatib, matnni kodlash turini ham tanlash mumkin:

<meta http-equiv q "Content-Type" content q "text / html; charset q utf-8">

Belgilangan vaqtdan keyin sahifani avtomatik ravishda qayta yuklash uchun <meta> tegining refresh xususiyatidan foydalanish mumkin:

<meta http-equiv q "refresh" content q "30">

Ushbu misolda sahifa 30 soniyadan keyin qayta yuklanadi. Agar siz darhol boshqa bir sahifaga o'tishini hohlasangiz, url parametridagi URLni belgilashingiz mumkin:

<meta http-equiv q "refresh" content q "0; url q http: //nammqi.uz /">

Izoh. HEAD bo'limida yana <base>, <link>, <script>, <style> kabi va boshqa teglardan foydalanish mumkin. Ushbu teglarni keyinroq ko'rib chiqamiz.

BODY bo'limi. Hujjatning asosiy qismi. Ushbu bo'lim hujjatning barcha mazmunini o'z ichiga oladi.

Strict formatda <body> tegining tarkibi blok elementlari ichida bo'lishi kerak, masalan, <p>, <div> yoki boshqalar:

```
<body>
<p>Hujjat matni</p>
<div>Hujjat matni</div>
</body>
```

<body> tegi quyidagi parametrlarga ega:

- **bgcolor** -web-sahifaning fon rangini belgilaydi. Fon rangi odatdagi rejimda oq bo'ladi. Rang olti honali o'n oltilik sanoq sistemasidagi raqamlar bo'yicha aniqlanadi. Har bir komponentning rangi (qizil, yashil va ko'k) uchun, qiymat 00 dan FF oralig'ida o'rnatiladi. Bu qiymatlar, avval "#" belgisi qo'shiladi, masalan, #FF0000 qiymati qizil rangga mos keladi, # 00FF00 to'q yashil rangga va #FF00FF - binafsha rangga (qizil va ko'k aralashmasi) mos keladi;
- **background** rasm URL manzilini ko'rsatib, hujjat uchun fon berish imkonini beradi;
- **alink**- faol havolaning rangini belgilaydi;
- **link**- o'qilmagan havola rangini belgilaydi;
- **vlink**- allaqachon ko'rilgan havolaning rangini belgilaydi;
- **text** - matn rangini belgilaydi.

Masalan, <body> tegi quyidagicha bo'lishi mumkin:



2.2.1-misol.BODY tegining hususiyati

```
<!DOCTYPE HTML>
<html><head>
<title> Sahifa sarlavhasi </ title>
<meta http-equiv q "Content-Type" content q "text / html; charset q utf-8">
</ head>
<body alink q "# FF0000" link q "# 000000" vlink q "# 000080" text q "#
000000">
Hujjat matni
</ body>
</ html>
```



2.2.1-rasm



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. *HTML hujjat tuzilishi qanday bo'ladi?*
2. *Hujjatning texnik tavsifi nima?*
3. *HEAD bo'limida qanday teglar berilishi mumkin?*
4. *<meta> tegining qo'llanilishi haqida nimalarni bilasiz?*
5. *BODY bo'limining vazifasi nima?*

2.3-§. Matnlarni bezash.



O`quv modullari

maxsus belgilar, matn shirifti, hujjatni formatlash, gorizontal to'g'ri chiziq, sarlavhalar, xat boshi

Alohida belgilar bilan ishalsh. Yuqorida aytib o'tilganidek, HTML formatlash tili hisoblanadi. Shuning uchun, alohida harflar bilan bir qatorda matnning barcha qismlarini shakllantirish muhimdir. Biroq, teglarni o'rganishdan avval, maxsus belgilarni ko'rsatish imkoniyatini ko'rib chiqaylik. Bunday belgilar, masalan, (<) va (>) bu belgilaridan kamroq foydalaniladi, chunki bu belgilar HTML teglarini ifodalaydi. Maxsus belgilarni ko'rsatish uchun HTML ekvivalentlari ishlatiladi.

Masalan, "<title> va </title> teglari orasiga sarlavha matni kiritiladi" matnini chop etish uchun "< title > va & lt; /title > teglari orasiga sarlavha matni kiritiladi" shaklida yozish talab etiladi.

Ushbu misolda biz kichik (<) belgisini < bilan va katta (>) belgisini > bilan almashtirdik.

Eng ko'p ishlatiladigan HTML ekvivalentlari ro'yxatini ko'rib chiqaylik:

- < - kichik belgisi (<);
- > - katta belgisi (>);
- & - ampersand (&);
- - noaniq bo'shliq;

- " – qo'shtirnoq (");
- © - mualliflik belgisi (©);
- ® - ro'yxatdan o'tgan tovar belgisi(®);
- ™ - Savdo belgisi (™).

Matn qismlarini belgilash.

 tegi qalin harfli matnni ko'rsatadi:

 Qalin harfli matn </ b>

 tegi o'rniga mantiqiy formatlash yorlig'ini ishlatish yaxshiroq

:

 Qalin harfli matn </ strong>

<i> tegi matnni kursiv ko'rsatadi:

<i> Kursiv matn </ i>

<i> tegi o'rniga mantiqiy formatlash tegini ishlatish yaxshiroq:

 Kursiv matn

<u> yorlig'i matnning ostiga chizilganligini ko'rsatadi:

<u> Ostga chizilgan matn </ u>

<strike> va <s> teglar matn o'rtasiga chizib qo'yadi:

<strike> O'rtasiga chizilgan matn </ strike>

<s> O'rtasiga chizilgan matn </ s>

Past va yuqori ko'rsatkichli matnlarni yaratish. <sub> tegini matnni satr sathidan pastga siljitadi va shrift hajmini kichraytiradi, pastki indeksni yaratish uchun ishlatiladi, masalan, H₂O:

Suv formulasi- H<sub>2</ sub>O

<sup> tegida matn satr sathidan yuqoriga o'tib, shrift o'lchamini kichraytiradi. Ushbu teg tez-tez ishlatiladi, masalan, m²:

Maydon o'lchov birligi - m<sup>2</ sup>

Matn shriftini belgilash. tegi shrift hajmini, turini va rangi bildiradi. U quyidagi parametrlarga ega:

- **face** - shrift turini belgilash uchun ishlatiladi:

Masalan:

 Matn </ font>

Nafaqat bitta balki vergul bilan ajratilgan bir nechta turlarni ko'rsatish mumkin. Belgilangan nom shrift turiga mos kelishi kerak. Shrift foydalanuvchining kompyuterida topilmasa, shrift ishlatilmaydi;

- **size** - hajmi 1 dan 7 gacha bo'lgan o'lchamlarda shrift hajmini belgilaydi. Standart web-brauzer tomonidan ishlatiladigan o'lcham 3 ga teng. Shrift hajmi 1 dan 7 gacha bo'lgan raqam sifatida va tegishli birliklarda qancha birlikni oshirish kerak ("Q" belgisi) yoki shtrix hajmini taglikga nisbatan kamaytirish ("- " belgisi) orqali ifodalanadi:

 Matn </ font>

 Matn </ font>

 Matn </ font>

➤ **color** shrift rangini belgilash imkonini beradi. Ranglar **bgcolor** parametri bilan bir xil tarzda o'rnatiladi :

`<font color q "#FF0000"> Matn </font>`

Raqamlar o'rniga siz rang nomlarini foydalanishingiz mumkin:

`<font color q "qizil"> matn </font>`

Eng keng tarqalgan ranglarning nomlarini yozamiz:

- black — #000000 — qora;
- white — #FFFFFF — oq;
- yellow — #FFFF00 — sariq;
- silver — #F0F0F0 — kumush;
- red — #FF0000 — qizil;
- green — #00F000 — yashil;
- gray — #F0F0F0 — kulrang;
- blue — #0000FF — kok;
- navy — #0000F0 — to'q ko'k;
- purple — #F000F0 — qirmizi.

Bundan tashqari, matnlarni formatlash uchun boshqa teglar ham mavjud: Matnni kattaroq shrift bilan chop qilish uchun `<big>` tegi qo'llaniladi.

`<big> matn </big>` katta o'lchamda

Matnni kichikroq shrift bilan chop qilish uchun quyidagi juft tegni ishlatish `<small>`:

`<small> matn </small>` kichik o'lchamda

Hujjatlarni formatlash. Oldingi qismda ko'rilgan deyarli barcha teglar hususiy formatlash belgilaridir. Qo'shimcha `<strong>` va `<em>` teglaridan ham foydalaniladi. Bu teglar mantiqiy matnni formatlash belgilari bo'lib, ular navbati bilan juda muhim va sodda qismlarga ajratish uchun ishlatiladi. Mantiqiy formatlash belgilaridan hujjatning tuzilishini ifodalash uchun foydalaniladi va turli xil web-brauzerlarda turlicha ko'rsatilishi mumkin. Keling, mantiqiy formatlashning asosiy teglarini ko'raylik:

- `<cite> ... </cite>` - atamalar va ularning nomlarini belgilashda foydalaniladi;
- `<code> ... </code>` - dastur kodining qismlarini belgilash uchun xizmat qiladi;
- `<acronym> ... </acronym>` - qisqartmalarni belgilash uchun ishlatiladi;
- `<kbd> ... </kbd>` - qismni foydalanuvchi tomonidan klaviaturadan kirish sifatida belgilaydi;
- `<q> ... </q>` - qisqa matnlarni belgilash uchun ishlatiladi;
- `<samp> ... </samp>` - dastur tomonidan chiqarilgan natijani belgilash uchun ishlatiladi;
- `<var> ... </var>` - o'zgaruvchining nomlarini belgilaydi.

Satrlarni ajratish. Satrlarni ajratish uchun bitta `<br>` tegidan foydalaniladi.

HTML hujjatiga

satr1

satr2

satr3

matn yozilsa

Web-brauzer uni bir satrda aks ettiradi: "satr1satr2satr3". Satrlarni bir-birining tagida joylashgan bo'lishini ta'minlash uchun har bir satr oxirida
 tegini qo'shishingiz kerak:

```
satr1<br> satr2 <br> satr3<br>
```

Matnni aks holida qanday yozilgan bo'lsa shu shaklda chiqarish uchun <pre> tegini ishlatish mumkin:

```
<pre>
satr1
satr2
satr3
</pre>
```

Ushbu misolda matnlar bir-birining tagiga joylashtiriladi.

Gorizontal to'g'richiziq. Bitta <hr> tegi gorizontal chizish imkonini beradi. <hr> tegi quyidagi parametrlarga ega:

- ❖ **size** - chiziq qalinligi: <hr size q "5">
- ❖ **width** - chiziq kengligi. Web-brauzer oynasining kengligi bilan bog'liq ravishda har ikkala pikselda va foizda qiymatni belgilashingiz mumkin:
<hr size q "5" width q "100"> <hr size q "5" width q "100%">
- ❖ **align** – joylashishni tekislash. Parametr quyidagi qiymatlarni qabul qilishi mumkin:
 - center – markazda joylashish
<hr sizeq "2" widthq "200" color q "red" align q "center">
 - left - chap burchak:
<hr size q "2" width q "200" color q "red" align q "left">
 - right- o'ng tomonga tjoylashgan:
<hr sizeq "2" width q "200" color q "red" align q "right">
 - noshade - bu parametrlarning mavjudligi chiziqning relefligini bekor qiladi:
<hr size q "2" width q "200" align q "center" noshade >

Sarlavhalar. Sarlavhalar oltita turli o'lchamlarga ega bo'lishi mumkin:

```
<hx> Sarlavha </ hx>
```

bu erda x - 1 dan 6 gacha bo'lgan son.

1-raqam bilan sarlavha eng katta:

```
<h1> Eng katta nom </ h1>
```

6-raqam bilan sarlavha eng kichik:

```
<h6> eng kichik sarlavha </ h6>
```

Asosiy parametr align web-brauzer oynasiga nisbatan sarlavha joylashishini belgilaydi. Quyidagi qiymatlarni qabul qilishi mumkin:

- ✓ center – markazga joylashtirish:

```
<h1 align q "center"> 1-darajali sarlavha markazda</ h1>
```

- ✓ left – chapga tekislash

```
<h2 align q "left"> chapga tekislangan 2 darajali matn</ h2>
```

- ✓ right- o'ng tekislash:

```
<h6 align q "right"> o'ng tomonga tekislangan 6 darajali matn eng kichik sarlavha </h6>
```

Xat boshini ajratish. <p> yorlig'i matnni alohida xat boshiga ajratishga imkon beradi. Web-brauzerlar paragraflarni bo'sh satr bilan bir-biridan ajratadi. Yakunlovchi </p> tegi majburiy emas.

- **align** asosiy parametr hisoblanadi va u gorizontaal tekislanishni o'rnatadi. Parametr quyidagi qiymatlarni qabul qilishi mumkin:

- **center** – markazdan tekislash;
<p align q "center"> Matn markazga joylashadi</p>
- **left** - chap tomondan tekislash;
<p align q "left"> Chap tomonga tekislangan matn </p>
- **right** - o'ngdan tekislash:
<p align q "right"> O'ng tomonga tekislangan matn </p>
- **justify** - ikki tomon bo'yicha tekislash:
<p align q "justify"> Ikki tomon bo'yicha tekislangan matn </p>

Izoh tegi. <! - va -> teglaridagi matn web-brauzer tomonidan ko'rsatilmaydi va bu oraliqqa izoxlar kiritiladi.

<! - Matn ->



Manba kodida izohlardan foydalanilsa kerakli qism tezda topiladi. Bu ayniqsa boshlang'ich web dizaynerlari uchun juda muhimdir.



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. Web sahifada mahsus belgilar qanday aks ettiriladi?
2. Quyi va yuqori ko'rsatkichlarga matn yozish qanday amalga oshiriladi?
3. Web sahifada xat boshi qanday teg bilan ifodalanadi va uning qanday parametrlari bor?
4. Matnlarni formatlash qanday amalga oshiriladi?
5. Web sahifada sarlavhalar qanday ifoda etiladi?

2.4-§. Ro'yxatlar.



O`quv modullari

Ro'yxatlar, markerlangan ro'yxatlar, raqamlangan ro'yxatlar, aniqlangan ro'yxatlar

Ro'yxat - maxsus belgilar yoki raqamlar bilan belgilangan matnlarning tartiblangan bandlaridan iborat. Har bir variantni alohida ko'rib chiqaylik.

Markerlangan ro'yxatlar. Markerlangan ro'yxatlar va teglari orasida joylashadi. Ro'yxatdagi har bir elementdan oldin tegini

joylashtirishingiz kerak. Yakunlovchi `</li>` yorlig'i ixtiyoriy. 2.4.1- misolda ro'yxatning tuzilishi keltirilgan.



2.4.1-misol. Markerlangan ro'yxat

```
<ul>
<li> Birinchi nuqta </li>
<li> Ikkinchi nuqta </li>
</ul>
```

`<ul>` tegida ro'yxat satrlari joylashgan belgini ko'rsatishga imkon beruvchi **type** parametri bor. Bu parametr quyidagi qiymatlarni qabul qilishi mumkin:

- `disc` – ichi bo'yalagan doira shaklidagi belgilar:

```
<ul type q "disc">
<li> Birinchi element </ li>
<li> Ikkinchi element </ li>
</ul>
```

- `circle` – ichi bo'sh shaklidagi belgilar:

```
<ul typeq "circle">
<li> Birinchi element </ li>
<li> Ikkinchi element </ li>
</ul>
```

- `square` – To'ldirilgan kvadrat shakldagi belgilar:

```
<ul type q "square">
<li> Birinchi element </ li>
<li> Ikkinchi element </ li>
</ul>
```

Raqamlangan ro'yxatlar. Raqamlangan ro'yxat `<ol>` va `</ ol>` tegi oralig'iga joylashtirilgan bo'ladi. Har bir ro'yxat elementidan oldin `<li>` tegini joylashtirishingiz kerak. Yakunlovchi `</ li>` yorlig'i ixtiyoriydir.

2.4.2-misolda raqamlangan ro'yxatning tuzilishi ko'rsatiladi.



2.4.2-misol. Raqamlangan ro'yxat

```
<ol>
<li> Birinchi nuqta </li>
<li> Ikkinchi nuqta </li>
</ol>
```

`<ol>` tegida ikkita parametr mavjud. Ularning birinchisi **type** - ro'yxatning satrlari harflar shaklida belgilash imkonini beradi.

Bu parametr quyidagi qiymatlarni qabul qilishi mumkin:

- `A` - satrlar katta harflar bilan belgilangan:

```
<ol type q "A">
<li> Birinchi element </ li>
<li> Ikkinchi element </ li>
```

- `</ol>`
 - a - satrlar kichik harflar bilan belgilangan:


```
<ol type q "a">
<li> Birinchi element </li>
<li> Ikkinchi element </li>
</ol>
```
 - I - satrlar katta rim raqamlar bilan belgilangan:


```
<ol type q "I">
<li> Birinchi element </li>
<li> Ikkinchi element </li>
</ol>
```
 - i - elementlar kichik rim raqamlar bilan belgilangan:


```
<ol type q "i">
<li> Birinchi element </li>
<li> Ikkinchi element </li>
</ol>
```
 - 1 - satrlar raqamlar bilan belgilangan (standart):


```
<ol type q "1">
<li> Birinchi element </li>
<li> Ikkinchi element </li>
</ol>
```

`<ol>` – tegining ikkinchi parametri – start satrlarni raqamlash boshlanishini belgilaydi:

```
<ol type q "1" start q "5">
<li> Birinchi element </li>
<li> Ikkinchi element </li>
</ol>
```

`<li>` tegi, shuningdek, ushbu ro'yxat elementining sonini o'zgartirishga imkon beruvchi **value** parametriga ega:

```
<ol type q "1">
<li> Birinchi element </li>
<li value q "5"> Ikkinchi element</li>
<li> Uchinchi elemenr </li>
</ol>
```

Ushbu misolda "Birinchi element" 1 raqami, "Ikkinchi element" - 5 raqami va "Uchinchi element" - 6 raqami bo'ladi.

Aniqlangan ro'yxatlar. Aniqlangan ro'yxatlar `<dl>` tegi yordamida tasvirlanadi. Qo'shimcha qo'shish uchun `<dt>` tegidan foydalaniladi va `<dd>` tegini qo'shish uchun. Yopiluvchi `</dt>` va `</dd>` belgilar ixtiyoriy. Aniqlangan ro'yxatlardan foydalanish 2.4.3- misolda keltirilgan.

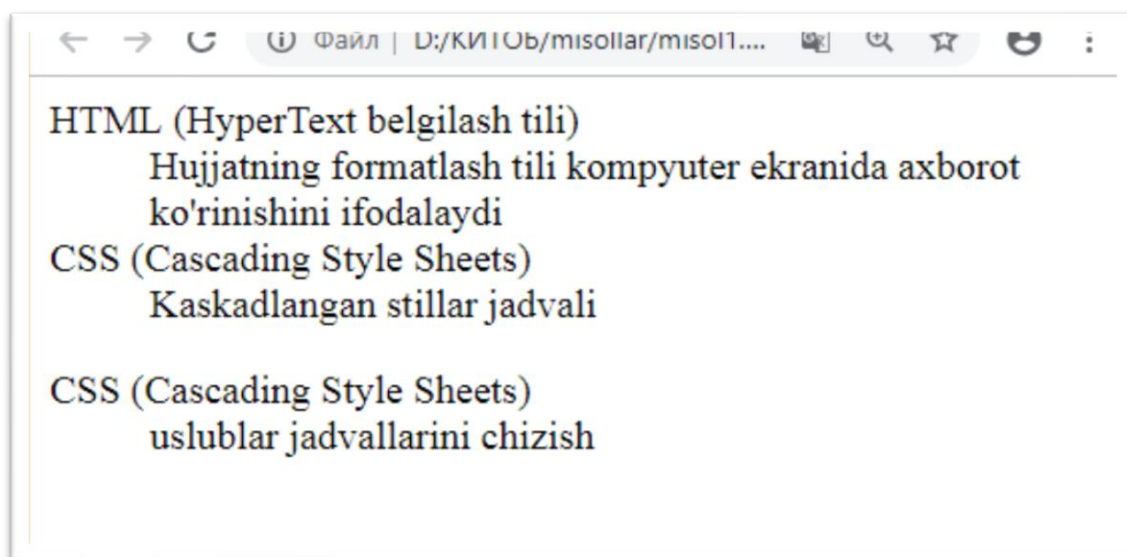


2.4.3-misol. Aniqlangan ro'yxatlar

```

<!DOCTYPE HTML>
<html>
<head>
<title> Aniqlangan ro'yxatlar</title>
<meta http-equiv q "Content-Type" content q "text / html; charset q utf-8">
</head>
<body>
<dl>
<dt> HTML (HyperText belgilash tili) </ dt>
<dd>
Hujjatning formatlash tili kompyuter ekranida axborot ko'rinishini ifodalaydi
</dd>
<dt>CSS (Cascading Style Sheets)</dt><dd>Kaskadlangan stillar jadvali</dd>
</dl>
</body></html>
</dd>
<dt>CSS (Cascading Style Sheets) </dt><dd> uslublar jadvallarini chizish
</dd>
</dl>
</body></html>

```



2.4.3-rasm



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. Ro'yxat nima?
2. Web sahifada raqmlangan ro'yxatlar qanday aks ettiriladi?
3. Web sahifada markerlangan ro'yxatlar qanday aks ettiriladi?
4. Web sahifada markerlangan ro'yxat markerlari sifatida ixtiyoriy belgini ishlatish mumkinmi?

5. Web sahifada aniqlangan ro'yxatlar qaysi HTML teglari yordamida hosil qilinadi?

2.5-§. Jadvallar.



O`quv modullari

jadval, jadval sarlavhasi, jadval qatorlari, jadval yacheykalari

HTML hujjatida jadvallar quyidagi hollarda qo'llaniladi:

- ma'lumotlarni taqdim etish vositasi sifatida;
- sahifadagi matnni va grafikani to'g'ri joylashtirishingiz mumkin bo'lgan sahifa dizayni elementi sifatida.



2.5.1-misol. HTML jadvallarining tuzilishi

```
<table border q "1" width q "200">  
<caption> Jadval sarlavhasi </caption>  
<tbody><tr>  
<td align q "center"> 1 </td><td align q "center"> 2 </td>  
</tr>  
<tr>  
<td align q "center"> 3 </td><td align q "center"> 4 </td>  
</tr></tbody>  
</table>
```

1	2
3	4

2.5.1-rasm

Hujjatga jadval joylashtirish

`<table>` tegi quyidagi parametrlarga ega:

- **border** - jadvalning chegara chiziqlarini namoyish qilishni boshqaradi, shuningdek jadval atrofida chegara qalinligini belgilaydi. Odatiy holda, jadval ko'rsatilmaydi:

`<table><!-- Bu yerda jadval chiziqlari ko'rinmaydi -->`

`<table border q "0"><!-- Bu yerda jadval chiziqlari ko'rinmaydi-->`

`<table border q "5"><!-- Bu holda jadval chiziqlai ko'rinadi, jadval atrofidagi ramka qalinligi 5 piksel -->`

➤ **cellspacing** - jadval ichidagi chiziqlarning qalinligini, qo'shni jadvallar chegaralari o'rtasidagi masofani bildiradi. Odatiy holda parametr 2 qiymatiga ega. Agar parametr 0 ga o'rnatilgan bo'lsa, qo'shni jadval ramkalari bir qatorga birlashadi:

`<table cellpaddingq"0">`

➤ **cellpadding** - hujjat ramkasi va hujjat ichidagi ma'lumotlar orasidagi bo'shliqni belgilaydi:

`<table cellpadding q "2">`

Odatiy holda qiymati 1 ga o'rnatilgan bo'ladi;

➤ **width**- jadvalning kengligi pikseldagi yoki oynaning o'lchamiga nisbatan foizini bildiradi:

`<table width q "200">`

`<table width q "100%">`

➤ **align** jadval gorizonta joylashish joyini bildiradi. Quyidagi qiymatlarni qabul qilishi mumkin:

- left - jadval chapga moslangan va matn uning atrofida o'ng tomonga o'giriladi:

`<table align q "left">`

- right – jadval o'ng tomonga moslashgan va matnni chap tomondg o'giradi:

`<table align q "o'ng">`

- center- jadval markazlashtirilgan:

`<table align q "center">`

➤ **bgcolor** - jadvalning orqa fon rangini ko'rsatadi:

`<table bgcolor q "silver">`

`<table bgcolor q "#C0C0C0">`

Jadval sarlavhasi

`<caption>` tegi jadval sarlavhasini belgilashga imkon beradi. U bitta **align** parametr ga ega. Ushbu parametr quyidagi ikki qiymatdan birini olishi mumkin:

- top - sarlavha jadval yuqori qismida joylashgan:

`<caption align q "top"> Jadval sarlavhasi </caption>`

- bottom - sarlavha jadval pastki qismida joylashgan:

`<caption align q "bottom"> jadval sarlavhasi </caption>`

Jadval qatorlari

Juft `<tr>` tegidan foydalanib, jadvalning satrlari tavsiflanadi. Quyidagi parametrlarga ega:

➤ **align** jadvalning ma'lumotlarini gorizonta ravishda joylashuvini ko'rsatadi.

Teg quyidagi qiymatlarni qabul qilishi mumkin:

- left - chap (standart):

`<tr align q "left">`

- right- o'ng:

<tr align q "o'ng">

- center - markaz:

<tr align q "center">

- justifiy – kenglik bo'yicha:

<tr align q "justify">

➤ **valign** - jadvalning yacheykalarida vertikal joylashuvini belgilaydi.

Quyidagi qiymatlarni qabul qilishi mumkin:

- top - yuqorida:

<tr valign q "top">

- middle – o'rta:

<tr valign q "middle">

- bottom - pastki qismida:

<tr valign q "bottom">

- baseline – asos chiziq bo'yicha:

<tr valign q "baseline">

Jadval yacheykalari

<td> tegidan foydalanib, jadval yacheykalari tavsiflanadi.

<td> tegi quyidagi parametrlarga ega:

- ✓ **align va valign** ma'lumotni yacheykaning joylashishning gorizontal va vertical o'rinlarini belgilaydi;
- ✓ **bgcolor** yacheykaning fon rangini bildiradi;
- ✓ **colspan** - gorizontal tarzda birlashtirilishi kerak bo'lgan yacheykalar sonini belgilaydi;
- ✓ **rowspan** -vertikal ravishda birlashtirilishi kerak bo'lgan yacheykalar sonini ko'rsatadi.



2.5.2.-misol.Yacheykalarni gorizontal tarzda birlashtirish

<table border q "1" width q "200">

<caption> Jadvalning sarlavhasi </caption>

<body>

<tr><td align q "center" colspan q "2"> 1 va 2 birlashtirildi </td></tr>

<tr><td align q "center"> 3 </td><td align q "center"> 4 </td></tr>

</body>

</table>

Jadvalning sarlavhasi	
1 va 2 birlashtirildi	
3	4

2.5.2.- rasm

Shunday qilib, bu misolda biz jadval qatorini birlashtirdik

`<td align q "center"> 1 </ td>` qatorni

`<td align q "center" colspan q "2"> 1 va 3 birlashtirildi </ td>`

`<td align q "center"> 3 </ td>` qatori o'chirildi



2.5.2-misol. Yacheykalarni vertikal ravishda birlashtirish

`<table border q "1" width q "200">`

`<caption> Jadvalning sarlavhasi </ caption>`

`<body>`

`<tr><td align q "center" rowspan q "2"> 1 va 3 birlashtirildi </ td>`

`<td align q "center"> 2 </ td></ tr><tr><td align q "center"> 4 </ td>`

`</ tr></ tbody></ table>`

Ushbu misolda biz jadval yacheykalarini birlashtirdik

Jadvalning sarlavhasi	
1 va 3 birlashtirildi	2
	4

2.5.2-rasm

`<td align q "center"> 1 </ td>` haqida `<td align q "center" rowspan q "2">`
1 va 3 birlashtirildi `</ td>` satr `<td align q "center"> 3 </ td>` o'chirilgan.

`<th>` tegi sarlavhalar bo'lgan yacheykalarni belgilashga imkon beradi. Bunday yacheykalarining tarkibi qalin va markazlashgan holda ko'rsatiladi. Boshqa tomondan, `<th>` tegi `<td>` tegiga o'xshaydi



2.5.3-misol. Jadval yacheykalarini belgilash

```
<html><head>
<title> Jadval uchun yacheykalarni tanlash </title>
<meta http-equiv q "Content-Type" contentq"text/html;
charsetqutf-8">
</head>
<body>
<table border q "1" width q "500">
<caption> Jadval sarlavhasi </caption>
<thead><tr>
<th> Markasi </th>
<th> rangi </th><th>Chiqish yili </th></tr></thead>
<tbody><tr>
<td> Cobalt</td>
<td> Qora </td>
<td> 2017 </td>
</tr>
<tr>
<td> Damas</td><td> Oq </td><td> 2015</td></tr></tbody>
</table></body></html>
```

Jadval sarlavhasi		
Markasi	rangi	Chiqish yili
Cobalt	Qora	2017
Damas	Oq	2015

2.5.3-rasm



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

- 1.Web sahifada jadvallar qanday maqsadlarda ishlatiladi?
- 2.Jadval hosil qilish uchun qanday HTML teglari ishlatiladi?
- 3.Jadval satrlari qaysi teg bilan ifodalanadi?
- 4.Jadval yacheykalari qaysi teg bilan ifodalanadi?
- 5.Jadval yacheykalarini ustun bo'ylab birlashtirish qanday amalga oshiriladi?
- 6.Jadval chiziqlarini hususiyatlarini o'zgartirish qanday bajariladi?

2.6-§. Freymlar.



O`quv modullari

freymlar, web sahifani qismlarga bo'linishi, freym strukturasi

Freymlar sizga web-brauzer oynasini bir nechta qismlarga ajratish imkonini beradi, ularning har biriga alohida HTML-hujjat o'rnatilishi mumkin.

Web-brauzer oynasini bir nechta qismlarga bo'lish

Odatda, saytdagi barcha sahifalar uchun sarlavha va navigatsiya paneli bir xil ma'lumotlarni o'z ichiga oladi va sahifalarning faqat asosiy tarkibi o'zgaradi. Kvadrat tuzilishidan foydalanib, sarlavhani bir oynada, ikkinchi navigatsiya paneli va uchinchi sahifadagi asosiy sahifa tarkibiga qo'yishingiz mumkin. Bu esa, birinchi va ikki oynani o'zgarmagan qoldirib, uchinchi mazmunini o'zgartirishga imkon beradi. Ushbu tuzilma bilan web-sahifani yaratishga harakat qilaylik. Buning uchun 5 ta fayl yarating:

- doc1.html (2.6.1-misol) - Web-sahifa sarlavhasi;
- doc2.html (2.6.2-misol) - navigatsiya paneli;
- test.html (2.6.3-misol) - ramka tuzilishini tasvirlaydigan HTML hujjat.
- bob 1.html - 1-bobning mazmuni;
- bob 2.html - 2-bobning mazmuni;



2.6.1-misol. Sarlavhani o'z ichiga olgan HTML hujjati (doc1.html)

```
<!DOCTYPE HTML>
<html><head>
<title> Sarlavha </title>
<meta http-equiv q "Content-Type" content q "text / html; charset q utf-8">
</head>
<body>
<h1> Header </h1>
</body>
</html>
```

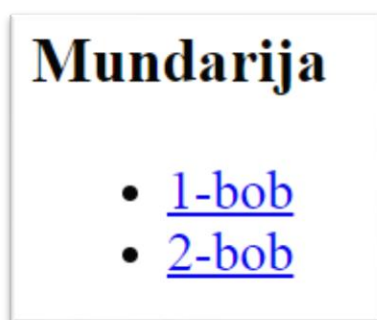


2.6.1-rasm



2.6.2-misol. Sarlavha sathini o'z ichiga olgan HTML hujjati (doc2.html)

```
<html>
<head>
<title> Navigatsiya paneli </title>
</head>
<body>
<h3> Mundarija </h3>
<ul>
<li><a hrefq"chapter1.html" targetq"chapter"> 1-bob </a> </li>
<li><a hrefq"chapter2.html" targetq"chapter"> 2-bob </a> </li>
</ul>
</body> </html>
```



2.6.2-rasm



2.6.3-misol. Ramka tuzilishini tavsiflovchi HTML hujjati (test.html)

```
<!DOCTYPE HTML>
<html>
<head>
<title> Ramkalardan foydalanish </title>
<meta http-equiv q "Content-Type" content q "text/html; charsetqutf-8"> </head>
<frameset rows q "100, *">
<frame src q "doc1.html" scrolling q "no">
<frameset cols q "20%, 80%">
<frame src q "doc2.html">
<frame src q "chapter1.html" name q "bo'lim">
</frameset> <noframes>
<p> Sizning web-brauzeringiz hech qanday ramka ko'rsatmaydi
</p> </noframes>
</frameset>
</html>
```



2.6.3-rasm

Yaratilgan barcha fayllar bitta papkada saqlanadi. So'ngra web-brauzerida test.html faylini oching. Shunday qilib, oyna uchta to'rtburchak joylarga bo'linadi. Oynaning pastki qismida navigatsiya paneli (chapda) va hujjatning asosiy qismi (o'ngda) mavjud. Alohida harakat qilayotganingizda, asosiy qismning tarkibi o'zgaradi va qolganlari o'zgarishsiz qoladi.

Keling, sarlavhaning fon rangini o'zgartiraylik. Buning uchun web-sahifaning fon kodini yozishingiz kerak.

Bilamizki, oddiy web-saytning hujjat kodini uchta usulda ko'rishingiz mumkin:

- View menyusida View HTML code ni bosing;
- Web brauzerining oynasida biron-bir joyni o'ng tugmasini bosing. Olingan kontekst menyusida View HTML-kod ni;
- Notepad yoki boshqa matn muharriridan foydalanib, manba kodini o'z ichiga olgan faylni oching.

Hujjatda ramkalar bo'lsa, dastlabki ikkita harakatlar natijalari boshqacha bo'ladi. Birinchi usul HTML-hujjatning har biri uchun hujjat kodi emas, balki faqat web-sahifaning tuzilishi uchun kodni ko'rsatadi. Boshqacha qilib aytganda, test.html faylining manba kodi ko'rsatiladi. Ikkinchidan, faqat bitta HTML hujjatning kodini ko'rsatish imkonini beradi va HTML hujjat kodi ko'rsatiladi.

Bizning holatda (doc1.html) manba kodini ko'rsatish uchun, sarlavhani o'z ichiga olgan maydonni o'ng tugmasini bosing. Ko'rsatilgan kontekst menyusida View HTML code ni tanlang.

Biz rangni almashtiring

`<body>`

`<body style q "background-color: #C0C0C0">` faylni saqlang va web-sahifani yangilang. Natijada, sarlavhaning fon rangi oqdan kul ranggacha o'zgaradi



2.6.2. Freymnlarni o'z ichiga olgan HTML hujjatning tuzilishi

HTML hujjatning ramkalar bilan tuzilishi

`<html>`

```

<head>
<title> Sarlavha </title>
<meta http-equiv q "Content-Type" tarkibi q "text / html; charset q utf-8">
</head>
<frameset rowsq"100,*">
<frame>
<frameset colsq"20%, 80%">
<frame>
<frame></frameset>
</frameset>
<p> Sizning web-brauzeringiz hech qanday ramka ko'rsatmaydi
</p></noframes></frameset></html>

```



2.6.2-rasm

<body> yorlig'i o'rniga, ramka tuzilishini tasvirlaydigan <frameset> yorlig'i ishlatiladi. Har bir alohida oyna <frame> yorlig'i bilan tavsiflanadi. Agar web-brauzer ramkalarni qo'llab-quvvatlamasa, matn <noframes> va </noframes> teglar orasida ko'rsatiladi. Ushbu teglarni batafsil ko'rib chiqing.

Freyimli struktura tuzilish tavsifi

<frameset> juftligi ramka tuzilishini tavsiflaydi. <frameset> va </ frameset> teglar ichida <frameset> va </ frameset> teglari bilan tasvirlangan faqat <frame> teglari yoki boshqa ramkalar majmui bo'lishi mumkin.

<frameset> yorlig'i quyidagi parametrlarga ega:

- satrlarni bo'laklarga bo'linishni ta'riflaydi:

<frameset rows q "100, *">

- Ustunlar ustunlarga bo'linishni anglatadi:

<frameset cols q "20%, 80%">

Qator va bo'g'inlar parametrlari qiymatlari kvadratlarning o'lchamidir. Kamida ikkita qiymat ko'rsatilishi kerak. Ro'yxatdagi barcha qiymatlar vergul bilan ajralib turadi. Olchamlari mutlaq birliklarda (piksellarda) yoki foizda ko'rsatilishi mumkin



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. Freym nima va u web sahifada qanday maqsadlarda ishlatiladi?
2. Freym hosil qilish tegini ayting?
3. Freym o'lchamlari qanday o'rnatiladi?

2.7-§. Grafika va multimedia. Giperishorat teglari.



O`quv modullari

grafika, multimedia, tovushli fayllar, giperishoratlar, ichki giperishoratlar, tashqi giperishoratlar

Grafika

Grafikalardan foydalanish web-sahifani vizuallik hususiyatini oshiradi. Rasmlar hujjatning mohiyatini va mazmunini yaxshiroq tushuntirishga yordam beradi.

Internetda quyidagi grafik formatlar qo'llaniladi:

- GIF - faqat 256 rangdan foydalanadi va oshkoralikni qo'llab-quvvatlaydi. Bundan tashqari, GIF faylida animatsiyalar bo'lishi mumkin;
- JPEG - sifatni yo'qotish bilan suratlarni siqish usuli. Shaffoflik va animatsiya qo'llab-quvvatlanmaydi;
- PNG – sifatni saqlash bilan grafikni saqlash tartibi. Shaffoflikni qo'llab-quvvatlaydi. Almashtirish sifatida GIF formatidan foydaalanilishi mumkin.



Hujjatda grafik bilan ishlash web-sahifaning yuklab olish vaqtining ko'payishiga olib keladi. Shuning uchun grafikadan faqat oqilona va samarali foydalaning.

Web-sahifada tasvir

Tasvirlar tegi yordamida web-sahifalarga kiritiladi. tegi blok tegi ichida bo'lishi kerak, masalan, <p>, <div> yoki boshqa teglar. Quyidagi parametrlari mavjud:

- src - rasm faylining URL manzili:

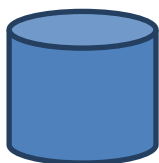
- alt - tasvirni yuklashdan oldin yoki grafik yopiq bo'lsa, sahifada ko'rsatiladigan matnli satr va rasm yuklanadi. Bunga qo'shimcha ravishda, sichqonchani tasvir ustida ko'chirsangiz, alt parametrda ko'rsatilgan matn asboblari panelidagi matn sifatida ko'rish mumkin:

- width - tasvirning pikseldagi kengligi:

```
<img src q "foto.gif" width q "480" altq "na'muna">
➤ height - tasvirning pikseldagi balandligi:
<img src q "foto.gif" width q "480" height q "60" alt q "Yordam">
```

ESLATMA!

Width va height parametrining qiymatlari tasvirning haqiqiy o'lchovlariga mos kelmasligi mumkin. Bunday holda, web-brauzer qayta masshtablashtirshni amalga oshiradi. Parametrlardan birining qiymati noto'g'ri bo'lsa, tasvir buziladi. Agar siz faqat bitta parametрни aniqlasangiz, ikkinchi qiymat haqiqiy tasvir o'lchamiga mutanosib birinchi qiymatga asosan avtomatik o'zgaradi.



Har doim width va height parametrlarini aniqlashtiring, chunki bu web-brauzga web-sahifada rasmlarni yuklashdan oldin formatlash imkonini beradi. Aks holda, har bir rasmni yuklanganda web-sahifaning boshqa elementlarini siljishiga olib keladi.

Quyidagi hususiyatlardan foydalanish mumkin:

- border - chegara qalinligi:

```
<img src q "foto.gif" border q "0" altq "na'muna">
```
- align - tasvirni matnga yoki web-sahifaning boshqa elementlariga nisbatan joylashtirish. Parametr quyidagi qiymatlarni qabul qilishi mumkin:
 - left - tasvir chapga moslangan va matn uning atrofida o'ng tomonga tekislanadi:

```
<p><img src q "foto.gif" align q "left" alt q "Yordam"> matn </ p>
```

- right - tasvir o'ng tomonga tekislangan va matn chap tomonda bo'ladi:

```
<p><img src q "foto.gif" align q "right" alt q "Yordam"> Matn </ p>
```

- top - tasvir joriy satrning yuqori qismida joylashadi:

```
<p><img src q "foto.gif" align q "top" alt q "Yordam"> Matn </ p>
```

- bottom - rasm joriy satrning pastki qismida joylashadi:

```
<p><img src q "foto.gif" align q "bottom" alt q "Yordam"> Matn </ p>
```

- middle – tasvir joriy chiziqning markaziga joylashadi:

```
<p><img src q "foto.gif" align q "middle" alt q "Yordam"> Matn </ p>
```

- hspace - tasvirdan matngacha gorizontaal siljish masofasi:

```
<p>
```

```
<img src q "foto.gif" align q "left" hspace q "20" alt q "Yordam"> Matn
</ p>
```

- vspace - rasmdan matngacha vertikal siljish masofasi:

```
<p>
```

```
<img src q "foto.gif" align q "left " vspace q "20" alt q "Yordam"> matn
</ p>
```

Tasvir fon sifatida

<body> tegi background hususiyati yordamida hujjat uchun fon tasvirini belgilash imkonini beradi:

<body background q "foto.gif" bgcolor q "blue"> Hujjatning tanasi </ body>
bgcolor parametrda fon tasvirining rangiga yaqin bo'lgan rangni ko'rsatishingiz kerak, chunki keskin o'tish, masalan, yorug'lik ohangidan qorong'i ohanggacha, sezilarli o'tishga olib keladi, chunki fon tasvir biroz kechikishi mumkin.

Multimedia. Bizda Web sahifani bezash uchun unga videoroliklar joylashtirish imkoniyati ham mavjud. Foydalanuvchi o'z brauzerida videoma'lumotni ko'rishi uchun videoma'lumot yozilgan faylni o'z kompyuteriga to'liq yuklab olishi lozim. Video fayllar katta hajm egallaganligi sababli uni tarmoq orqali kompyuterga yuklash uchun ko'proq vaqt sarflanadi. Brauzerlar odatda **AVI, Real Video va Windows Media** formatlaridagi videofayllarni tasvirlash imkoniyatiga ega. Ularni Web sahifa tarkibiga qo'shish uchun ham tegi ishlatiladi. Videofayl joylashgan manzilni ko'rsatish uchun esa **dynsrc** parametri ishlatiladi. Jimlik bo'yicha Web sahifa tarkibiga kiritilgan videoma'lumot Web sahifa yuklangandan keyin avtomatik ravishda bir martatasvirlanadi, lekin bizda tasvirlanish jarayonini boshqarish yoki bu jarayonni foydalanuvchi tomonidan boshqariladigan qilish imkoniyati mavjud. tegiga <start> parametrini qo'shish orqali biz yuklangan fideoklipni tasvirlash jarayonini bema'lol boshqarishimiz mumkin. <start> parametrining qiymatlari sifatida **mouseover** va **fileopen** kalit so'zlari ishlatiladi. **mouseover** parametri foydalanuvchi sichqoncha ko'rsatkichini videotasvir joylashgan soha ustiga olib kelishi bilan videotasvirning tasvirlanish jarayoni boshlanishini bildirsa, **fileopen** parametri esa Web sahifa kompyuterga to'liq yuklanib bo'lgandan keyin darhol tasvirlanish jarayoni boshlanishini bildiradi.

Misol:

Biz **loop** parametri orqali videorolikning necha marta tasvirlanishini boshqarishimiz mumkin. Buning uchun tasvirlanishlar sonini **loop** parametriga qiymat sifatida berishimiz zarur. Agar biz videorolikning takroriy tasvirlanishini doimiy ravishda takrorlanuvchi qilib qo'ymoqchi bo'lsak u holda **loop** parametriga **infinite** qiymatini berishimiz kerak bo'ladi. **Loopdelay** parametri takroriy tasvirlanish jarayonida tasvirlanishlar orasidagi vaqtni belgilaydi va qiymati millisekundlarda beriladi.

Misol:

Yuqoridagi misolda Web sahifa to'liq yuklanishi bilan videoklip ikki marta orasida 5 sekundlik tanaffus bilan tasvirlanadi.

 tegiga **controls** parametrini qo'yish orqali biz videofaylni to'liq o'zimiz boshqarishimiz mumkin, ya'ni Web sahifada videoklip bilan birgalikda uni boshqarish elementlari ham paydo bo'ladi. Boshqarish elementlariga tasvirni ishga tushirish, orqaga qaytarish, oldinga o'tkazish yoki to'xtatish kabi ishlarni bajaruvchi tugmalar kiradi.

<html>

<head><title>Video</title></head>

<body>

```
<p>Oddiy matn<img dynsrcq“movie.avi” startq“mouseover”loopq“2”
loopdelayq“5000”></p>
</body>
</html>
```

Videotasvirlarni Web sahifaga joylashtirish bu grafik ob'ektlarni joylashtirishning bir qismi bo'lganligi uchun **** tegining barcha parametrlarini videotasvirlar uchun ham qo'llash mumkin.

Tovushli fayllar

Web sahifaga tovushli fayllarni joylashtirish ham HTML ning multimedia imkoniyatlari doirasiga kiradi. Tovushli fayllar Web sahifada fon hisobida qatnashadi. **Tovushli fayl** Web sahifa to'liq yuklangandan keyin bajariladi. Foydalanuvchi kompyuteriga Web sahifa yuklanganda birinchi navbatda uning HTML tilida yozilgan fayli, keyin esa shu sahifada ishlatilgan grafik, video va audiofayllar yuklanadi.

Demak Web sahifaga tovushli fayl qo'yishimiz uchun biz sahifaning sarlavhasiga, ya'ni **<head>** va **</head>** teglari orasiga **<bgsound>** tegini joylashtirishimiz kerak bo'ladi. Bu tegning src parametri qiymat sifatida tovushli faylning **URL** manzilini qabul qiladi.

Brauzerlar **MIDI** va **WAV** formatdagi audiofayllarni taniydi va bimalol aytirish imkoniga ega. Boshqa formatdagi, ya'ni **MP3**, **Real Audio** kabi audiofayllarni aytirish uchun brauzerlarga qo'shimcha modullar qo'shish kerak bo'ladi. Xuddi videokliplar bilan ishlaganimizdek audiofayllar bilan ishlaganimizda ham faylning necha marta takrorlanishini **loop** parametri bilan boshqarishimiz mumkin. Qiymat sifatida esa takrorlanishlar soni beriladi. Agar audiofaylning to'xtovsiz takrorlanishi zarur bo'lsa **loop** parametriga **“-1”** qiymat berish kerak.

Misol: **<bgsound srcq“song.mid” loopq“-1”>**

Giperishoratlar.

Giperishoratlar sichqoncha tugmasini bosib tezda bir hujjatdan ikkinchisiga o'tish imkonini beradi. U barcha web-sahifalarni bitta tarmoqqa ulaydigan ko'priklar hisoblanadi.

Tashqi giperishoratlar

Tashqi giperishoratlar **<a>** tegidan foydalangan holda HTML hujjatida ko'rsatiladi.

<a> tegining asosiy parametri href. Ushbu parametr sichqonchani ko'rsatgichi bosilganda yuklanadigan web-sahifaning URL manzilini bildiradi. Ko'rsatkich matn

```
<a hrefq"http://www.mysite.com/file.html"> Havola matni </a> yoki tasvir
```

```
<a hrefq"http://www.mysite.com/file.html">
```

```
<img src q "http://www.mysite.ru/foto.gif" alt q "Yordam"></a>
```

bo'lishi mumkin.

URL manzilda & belgisi ishtirok etgan bo'lsa, uni HTML ekvivalenti **&** bilan almashtirish kerak:

`<a hrefq"index.php?idq5& nameqNik"> Ishorat matni </a>`

URL manzil absolyut va nisbiy bo'lishi mumkin.

Absolyut URL manzil

Absolyut URL manzil to'liq protokol nomi, kompyuterning domen yoki IP-manzilini, faylning yo'lini va fayl nomini o'z ichiga oladi. Masalan:

`http://www.mysite.ru/folder/file.html`

Fayl ildiz papkasida joylashgan bo'lsa, yo'lak qisqa bo'lishi mumkin:

`http://www.mysite.ru/file.html`

Fayl nomi ham bo'lmasligi mumkin.

<http://www.mysite.ru/>

`http://www.mysite.ru/folder/`

Nisbiy URL adreslar

Masalani qo'yilishiga qarab URL manzil hujjat joylashgan web-sahifasining joylashuv-pozitsiyasini hisobga olgan holda belgilanadi.

Quyidagi variantlarda berilishi mumkin:

✓ Istalgan web-sahifa havolani o'z ichiga olgan web-sahifa bilan bir xil papkada bo'lsa, URL faqatgina fayl nomini o'z ichiga olishi mumkin. Agar `http://www.mysite.uz/folder1/folder2/file1.html` da joylashgan web-sahifadan `http://www.mysite.uz/folder1/folder2/file2.html` sahifasiga o'tmoqchi bo'lsangiz, u holda havola quyidagilar bo'ladi:

`<a hrefq"file2.html"> Malumot matni </a>`

agar `http://www.mysite.uz/direct1/folder2/file1.html` da joylashgan web-sahifadan, `http://www.mysite.uz/folder1/folder2/folder3/file2.html` sahifasiga o'tishingiz kerak bo'lsa, shundan keyin havola quyidagi tarzda ko'rsatilishi mumkin:

`<a hrefq"folder3/file2.html"> Malumot matni </a>`

✓ agar `http://www.mysite.ru/direct1/folder2/file1.html` da joylashgan web-sahifadan `http://www.mysite.ru/folder1/file2.html`-ga o'tishingiz kerak bo'lsa, unda link quyidagicha: `<a hrefq"../file2.html"> Malumot matni </a>`

va `http://www.mysite.uz/folder1/folder2/folder3/file1.html` sahifasidan `http://www.mysite.uz/folder1/file2.html`-ga o'tilayotganda - bu ko'rinishda:

`<a hrefq"../..file2.html"> Malumot matni </a>`

Ko'pincha siz yangi web-brauzer oynasiga hujjatni yuklashingiz kerak. Buni amalga oshirish uchun `<a>` tegining maxsus **target** parametrda **_blank** qiymati ko'rsatiladi:

`<a hrefq"http://www.mysite.uz/file.html" targetq"_blank"> Ulanish </a>`

Maqsad parametrlarining boshqa qiymatlari ramkalarni o'rganishda ko'rib chiqiladi

Ichki giperishoratlar

Ichki giperishoratlardan foydalanib, mavjud web-sahifaning turli bo'limlariga havolalar yaratish mumkin. Hujjat juda katta bo'lsa, ichki

giperishoratlarning mavjudligi bo'limlar orasida tezlik bilan harakat qilish imkonini beradi.

```
<a hrefq"#qism1"> 1-bob </a>
```

Giperishorat<a> yorlig'i yordamida yaratiladi, lekin href parametrining o'rniga markerning nomini bildiruvchi ism parametri ishlatiladi:

```
<a nameq" qism1"></a>
```

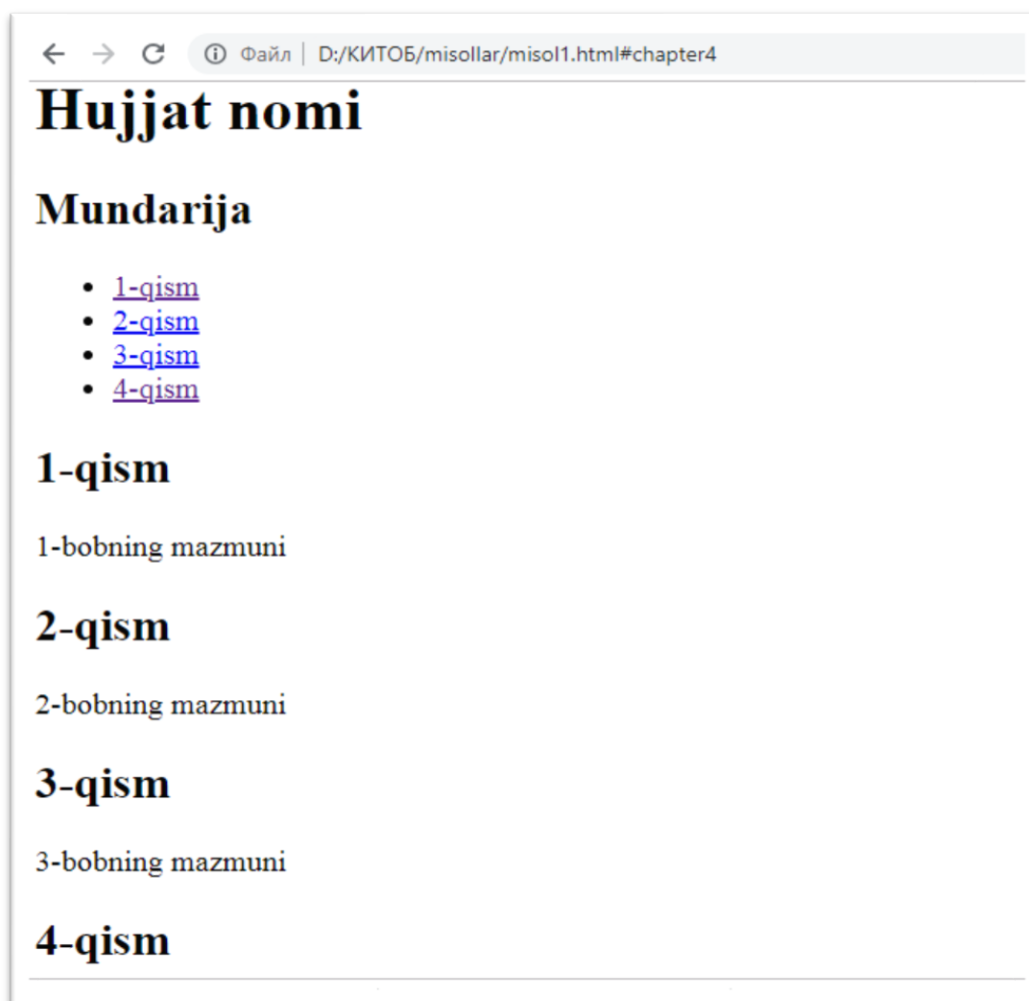
Ba'zida ko'rsatgich "yakor" deb ataladi. Siz boshqa hujjatning "bo'limlariga" ham murojaat qilishingiz mumkin. Bu quyidagicha amalga oshiriladi:

```
<a hrefq"http://www.mysite.uz/file.html#qism6">Matn</a>
```

 Hujjatning ichki havolalar bilan bog'langan tuzilishi 2.7.1-misolda ko'rsatilgan.

2.7.1-misol. Hujjatning ichki havolali tuzilishi

```
<!DOCTYPE HTML>
<html>
<head>
<title> Ichki havola yaratish </title>
<meta http-equivq"Content-Type" contentq"text/html; charsetqutf-8">
</head>
<body>
<h1>Hujjat nomi</h1>
<h2>Mundarija</h2>
<ul><li><a hrefq"#chapter1">1-qism</a> </li>
<li><a hrefq"#chapter2"> 2-qism</a> </li>
<li><a hrefq"#chapter3"> 3-qism</a> </li>
<li><a hrefq"#chapter4"> 4-qism</a> </li>
</ul>
<h2><a nameq"chapter1"></a>1-qism</h2>
<p>1-bobning mazmuni </p>
<h2><a nameq"chapter2"></a>2-qism </h2>
<p>2-bobning mazmuni </p>
<h2><a nameq"chapter3"></a>3-qism </h2>
<p>3-bobning mazmuni </p>
<h2><a nameq"chapter4"></a>4-qism </h2>
<p>4-bobning mazmuni </p>
</body>
</html>
```



2.7.1-rasm

Elektron pochta manziliga giperishoralar

Elektron pochta manziliga aloqa quyidagi tarzda bajariladi:

`<a href="mailto:mail@mysite.uz"> Matn </a>`

URL o'rniga, siz "mailto:" so'zi qo'shilgan elektron pochta manzilini ko'rsatasiz.



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. *Grafika nima va u web sahifada qanday maqsadlarda ishlatiladi?*
2. *Grafika bilan ishlash teglari va ularning xususiyatlarini ayting.*
3. *Multimedia nima va u web sahifada qaysi teg yordamida ifoda etiladi?*
4. *Gipperishorat nima?*
5. *Web sahifada gipperiishoratning qanday ko'rinishlaridan foydalaniladi?*
6. *Tahsqi gipperishorat qanday amalga oshiriladi?*
7. *Ichki gipperishorat qanday qo'yiladi?*
8. *Elektron manzilaga gipperishorat qanday qo'yiladi?*

2.8-§. Formalar



O`quv modullari

forma, boshqruv elementlari, matn va parolni kiritish maydoni

Formalar foydalanuvchi ma'lumotlarini web-serverga uzatish uchun mo'ljallangan. HTML formalar, foydalanuvchilar tomonidan, ma'lumotlarni kiritish va kiritilgan ma'lumotlardan keyinchalik foydalanish maqsadida tashkil qilingan. Formalar – matnli oynalar, tugmalar, bayroqchalar, ulagich (pereklyuchatel)lar va menyular kabi boshqariladigan elementlardan tashkil topishi mumkin. Formalar **<form>** va **</form>** teglari orasida joylashadi. Ba'zi hollarda foydalanuvchi bilan brauzer o'rtasida aktiv muloqot qilishga to'g'ri keladi, ya'ni ma'lumotlarni kiritishga va tahrir qilishga to'g'ri keladi. Bunday hollarda biz yuqorida aytganimizdek HTML hujjatning formalar deb ataluvchi bo'limidan foydalanamiz. Foydalanuvchi yuqorida ko'rsatilgan formalarni ma'lumotlar bilan to'ldiradi va uni qayta ishlash uchun serverga jo'natadi. Server bu odatda Web-server yoki elektron pochta serveri bo'lishi mumkin.

Shunday qilib har qanday formaning asosiy qismi bo'lib uning boshqariladigan elementlari hisoblanadi. Bu elementlarning har biri o'z nomiga ega va bundan tashqari bu elementlar o'zining boshlang'ich qiymatiga hamda joriy qiymatiga ega. Elementning boshlang'ich qiymati uning birinchi hosil qilinishida yoki uning qaytadan o'rnatilishida (inistializastiya) ekranda tasvirlanadi. Keyinchalik foydalanuvchi bu qiymatlarni o'zgartirishi mumkin. Foydalanuvchi tomonidan to'ldirilgan formaning qiymatlari «nomqqiymat» juftlik ko'rinishida qayta ishlanish uchun jo'natiladi. Formalar tashkil qilish form elementi yordamida amalga oshiriladi va biz uning umumiy ko'rinishi bilan tanishib chiqamiz.

Umumiy ko'rinishi:

Sintaksis : **<FORM></FORM>** (Blokli element)

Atributlar: id, class, style, title, lang, dir, xodisa

action q URL (formalarni qayta ishlash uchun ko'rsatilgan URL manzil)

method q GET | POST (formalardagi ma'lumotlarni jo'natish uchun HTTP metodi)

enctype q fayl tipi(formaga murojat qilish uchun)

accept q fayllar tiplari (bo'lishi mumkin bo'lgan fayllar tiplari)

accept-charset q kodlashtirishlar (bo'lishi mumkin bo'lgan simvollarni kodlash usullari)

name q CDATA (sstenariya uchun forma nomi)

target q freym (natijalarni hosil qilish uchun freym)

onsubmit q sstenariya (formani jo'natish)

onreset q sstenariya (formani qaytadan o'rnatish)

Ushbu ma'lumotni server tomonida qanday boshqarishi haqida PHP tilini o'rganish davomida tushuntiriladi. Hozircha formalarni yaratish uchun HTML xususiyatlarini ko'rib chiqamiz.

Saytni ro'yhatdan o'tkazish uchun forma yaratish



2.8.1-misol. Formalar bilan ishlashga doir misol

```
<html>
<head>
<title>Ariza </title>
<meta http-equiv q "Content-Type" content q "text/ html; charset q utf-8">
</head>
<body>
<h1> Ro'yxatga olish formasiga misol </h1>
<form actionq"file.php" methodq"POST" enctypeq"multipart/form-data">
<div>Login:<br>
<input typeq"text" nameq"pole1"><br>Parol:<br>
<input typeq"password" nameq"pole2" valueq" Parol "><br> URL-
manzil<br>
<input typeq"text" nameq"pole3" valueq"http://" sizeq"20"><br>Sayt
nomi:<br>
<input typeq"text" nameq"pole4" sizeq"20"><br>Sayt haqida qisqa
ma'lumot:<br>
<textarea nameq"pole5" rowsq"10" colsq"15"></textarea><br>Sayt
mavzusi :<br><select nameq"pole6">
<option valueq"0" selected>Tanlanmagan mavzu</option>
<option valueq"1">Mavzu1 </option>
<option valueq"2">Mavzu2 </option>
<option valueq"3">Mavzu2 </option></select><br>banner 88*31:<br>
<input typeq"file" nameq"pole7" sizeq"20"><br><br><input typeq"reset"
valueq"Tozalash"><input typeq"submit" valueq"Jo 'natish"></div></form>
</body>
</html>
forma.html faylni saqlang va web-brauzerida oching.
```

← → ↻ ⓘ Файл | D:/КИТОБ/misollar/forma.html

Ro'yxatga olish formasiga misol

Login:

Parol:

URL-manzil

Sayt nomi:

Sayt haqida qisqa ma'lumot:

Sayt mavzusi :

banner 88*31:
 Файл не выбран

2.8.1 -rasm

Natijada, web-brauzer oyna ostida beshta matnli maydon (ismi, parol, sayt URL-manzili, sayt nomi va sayt tavsifi) elementlari, fayl va tanlash elementlari ro'yxatidan iborat forma namoyish etadi, shuningdek, to'gridan-to'g'ri tozalash va jo'natish tugmalari ham paydo bo'ladi.

Dastur ma'lumotlarni qayta ishlaydi va saytni katalogga qo'shib, muvaffaqiyatli ro'yxatga olishni tasdiqlashi, kerakli maydon to'ldirilgan yoki noto'g'ri to'ldirilgan bo'lsa xato haqida xabar beradi. Bizning holatda, hech qanday qayta ishlash dasturi mavjud emas va forma ma'lumotlarini hech qayerga jo'natmaslik, aniqrog'i, "Fayl topilmadi" xatoga olib keladi. Sababi bizda hozircha forma ma'lumotlarini qayta ishlavchi "file.php" fayli yuq.

Boshqaruv elementlari tavsifi

<input> tegi qator, tugma yoki tasdiqlash belgisi kabi boshqarish vositalarini kiritish imkonini beradi. Ushbu teg quyidagi parametrlarga ega:

➤ **type** elementning turini bildiradi.

Ushbu maydon qiymatiga qarab, quyidagi shakl elementlari yaratiladi:

- **text** matn kiritish maydoni: `<input type="text">`

- **password**- barcha kirish belgilarining yulduzcha bilan almashtiriladigan parolni kiritish uchun matn maydoni: `<input type="password">`

- **file**- fayl nomini tanlash **Obzor** tugmasini xosil qiladi.

Web-serverga faylni yuborishga imkon beradi: `<input type="file">`

checkbox - sozlanishi yoki tanlanishi mumkin bo'lgan katakchalarni belgilash maydoni: `<input type="checkbox">`

- **radio** – bog'liq bo'lmagan tanlash tugmalarin hosil qiladi (ba'zan radio tugmalari deyiladi): `<input type="radio">`

- **reset** - bu tugma bosilganda butun shakl tozalanadi. Shaklning barcha elementlari standart qiymatlarni oladi: `<input type="reset">`

- **submit** - tugma bosilganda forma elementlariga kiritilgan ma'lumotlarni yuboradigan tugma: `<input type="submit">`

- **button** - odatdagi buyruq tugmasi: `<input type="button">`

Ushbu tugma faqat unga biriktirilgan skript yordamida foydalanish imkonini beradi.

- **hidden** - qiymat formadagi barcha ma'lumotlar bilan birga yuborilgan maxfiy element: `<input type="hidden">`

- **name**- elementning nomini belgilaydi. Har doim lotin harflari (masalan, pole) yoki lotin harflari va raqamlari birikmasi (masalan, pole1) bilan ko'rsatilishi kerak. Element nomi raqam bilan boshlanmasligi kerak: `<input type="text" name="pole1">`

- **disabled** - ariza elementiga kirishni rad etadi. Agar imkoniyat mavjud bo'lsa, element kulrang bo'ladi: `<input type="text" name="pole1" disabled>`

Qolgan parametrlar alohida har bir element uchun xosdir. Shuning uchun, har bir elementni alohida ko'rib chiqamiz.

Matn va parolni kiritish maydoni

Matn maydoni va parolni kiritish maydoni uchun quyidagi parametrlar ishlatiladi:

- **value** - standart maydonning matnini belgilaydi:

`<input type="text" name="pole1" value="http: //">`

- **maxlength** - maydonga kiritilishi mumkin bo'lgan maksimal belgilar sonini ko'rsatadi:

`<input type="text" name="pole1" value="http: //" maxlength="100">`

- **size** - kiritish maydonining aniq hajmini belgilaydi:

`<input type="text" name="pole1" value="http: //" maxlength="100" size="20">`

- **readonly** - faqat o'qish uchun qo'llaniladi. Agar parametr mavjud bo'lsa, maydonning qiymati o'zgarmaydi: `<input type="text" name="pole1" readonly>`

Jo'natish va tozlash buyruqlari

- **value** – ko'rsatilgan matnni bildiradi:

`<input type="submit" value="jo'natish">`

Yashirin maydon hidden

Yashirin maydon uchun bitta parameter mavjud:

- **value** - maydonning qiymatini belgilaydi:

`<input type="hidden" name="pole1" value="1">`

Bayroqcha belgisini o'rnatish maydoni

Elementini tavsiflashda quyidagi parametrlar qo'llaniladi:

➤ **value** - bayroqcha belgilansa, web-serverga yuboriladigan qiymat.

Agar katakcha tanlanmagan bo'lsa, qiymat uzatilmaydi. Agar parametr o'rnatilgan bo'lsa, standart qiymatdan foydalaniladi: - on:

`<input type="checkbox" name="check1" value="yes">` matn

➤ **checked** - katakchani tanlash bo'yicha tekshirganligini bildiradi:

`<input type="checkbox" name="check1" value="yes" checked>` matn

checkbox elementlari birlashtirilishi mumkin. Buning uchun **name** parametr uchun bir xil qiymatni belgilashingiz kerak. Domen nomidan so'ng serverdagi barcha qiymatlarni olish uchun kvadrat qavslar shlatilishi kerak:

`<input type="checkbox" name="check[]" value="1">` matn1 >

`<input type="checkbox" name="check[]" value="2">` matn2 >

`<input type="checkbox" name="check[]" value="3">` matn3 >

Yoqib-o'chirgich elementi

Elementini tavsiflashda quyidagi parametrlar qo'llaniladi:

value – element tanlangan bo'lsa, qiymat web-serverga uzatiladigan qiymati bildiradi: `<input type="radio" name="radio1" value="yes">` matn

Agar qiymatlarning hech biri tanlanmagan bo'lsa, ma'lumotlar uzatilmaydi;

checked – oddatiy rejimda tanlangan kalitni ko'rsatadi:

`<input type="radio" name="radio1" value="yes" checked>` matn

Kalit elementi faqat shu kabi elementlardan iborat bo'lgan guruhda mavjud bo'lib, ulardan faqat bittasini tanlash mumkin. Kalitlarni guruhga birlashtirish uchun **name** parametri va qiymat parametri uchun boshqa qiymat uchun bir xil qiymatni belgilashingiz kerak:

Jins:

`<input type="radio" name="radio1" value="erkak" checked>`erkak

`<input type="radio" name="radio1" value="ayol">` ayol

Matnli maydon

Juft `<textarea>` tegi forma ichida ko'p satrli matn kiritish uchun maydon yaratadi: `<textarea>` Standart matn `</textarea>`

Web-brauzer oynasida, maydon scrollash majmuasi bo'lgan to'rtburchaklar shaklida ko'rsatiladi. Yorliq quyidagi parametrlarga ega:

name - maydon nomi:

`<textarea name="pole2">` Standart matn `</textarea>`

cols - ko'rinadigan matn ustunlari soni:

`<textarea name="pole2" cols="15">` Standart matn `</textarea>`

rows - ko'rinadigan matn satrlari soni:

<textarea name q "pole2" cols q "15" rows q "10"> Standart matn
</textarea>

Ochiluvchi qiymatlar ro'yxati

<select>tegi ochiluvchi qiymatlar ro'yxatini hosil qiladi:

<select><option>Element1 </ option>

<option>Element2 </ option></ select>

<select> tegi quyidagi parametrlarga ega:

- **name** – ro'yhat nomlari: <selectname q "select1">

- **size**- vaqtning o'zida ko'rinadigan ro'yxat elementlarining sonini bildiradi:

<select name q "select1" size q "3">Odatiy bo'lib, hajmi 1 ga o'rnatiladi;

- **multiple** - ro'yxat bir vaqtning o'zida bir nechta elementni tanlashingiz mumkinligini ko'rsatadi. Ro'yxat nomidan keyin serverdagi barcha qiymatlarni olish uchun kvadrat qavslarni ko'rsatiladi:

<select name q "select []" size q "3" multiple>

<select> va </ select> teglar ichida <option> teglar mavjud, ular ro'yxatning har bir elementiga tavsif qilinadi.

<option> elementi quyidagi parametrlarga ega:

value - qiymat, ro'yxat elementi tanlansa, Web serverga o'tkaziladigan qiymati bildiradi. Agar parametr aniqlanmasa, ushbu elementning matni yuboriladi:

<select name q "select1">

<option value q "val1"> Element1 </ option>

<option> Element2 </ option> -ni tanlang

</ Select> ni tanlang

"Element1" elementi tanlansa, u yuboriladi

select1 q "val1"

"Element2" tanlansa, u yuboriladi

select1 q "Element2"

selected ro'yxatning qaysi elementi dastlab tanlanganligini ko'rsatadi:

<select name q "select1">

<label> tegi

<label> tegidan foydalanib, forma elementi uchun tushuntirish yozuvini belgilashingiz mumkin. Yorliqda ikkita parametr mavjud:

- **for** - matn biriktirilgan element identifikatorini belgilash imkonini beradi.

Xuddi shu identifikator form elementining id parametrda ko'rsatilishi kerak:

<label for q "pass1"> Parol *:</ label>

<input typeq "parol" nomi q "pass1" id q "pass1">

Forma elementi <label> tegi orlig'iga joylashtirilsa unda element bilan bog'lanadi va labelda parametrni ko'rsatmasa ham boladi:

<label> Parol *: <input typeq "parol" nomi q "pass1" id q"pass1"></ label>

- **accesskey** - tez bajarilishni belgilaydi. Ushbu tugmani <Alt> tugmachasi bilan bir vaqtning o'zida bosilganda kursor elementda bo'ladi:

<label accesskey q "N">

Parol *:<input typeq "password" name q "pass1" id q "pass1"></ label>



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. *Forma nima va u web sahifada qanday maqsadlarda ishlatiladi?*
2. *Forma hosil qilish uchun qanday teg ishlatiladi?*
3. *Forma hosil qilish tegining qanday parametrlari bor?*
4. *Forma elementlarini sanang.*
5. *<INPUT> tegi qanday maqsadlarda ishlatiladi va uning qanday parametrlari bor?*
6. *Matnli maydon qaysi teg bilan hosil qilinadi?*
7. *Tugma elementi qanday hosil qilinadi va uning qanday parametrlari bor?*

3-BOB. CSS ASOSLARI.

3.1-§. CSS (Cascading Style Sheets) asoslari. Selektor, klass, identifikatorlar va ularning CSSda ishlash prinsiplari. Selektor guruxlari.



O`quv modullari

CSS(Cascading Style Sheets), stillarning berilish usullari, selektor, klass, identifikator

CSS (Cascading Style Sheets) – bu stillar bilan ishlay oladigan kaskadli stillar majmuasidir. U web sahifalarga xar hil stillar berish uchun ishlatiladi. HTML xujjat ichida teglar bilan birlikda foydalanilishi mumkin. 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.

Stillar jadvali (CSS) to'liq holatda **1997** yilda yaratilib, “**WWW Consorcium**” ida qo'llab quvvatlangan va foydalanishga taqdim etilgan. O'sha paytlarda HTMLning 3 versiyasi ommabop edi. CSSni dastlab **Netscape Navigator 4.0** va **Internet Explorer 4.0** brouzerlarida ko'rish mumkin bo'lgan. Hozirda barcha brouzerlar CSSda yozilgan kodlarni tushina oladi. Stillarga oid dasturlash tilini o'rganish sodda bo'lib, yangi stillar qo'shilishi natijasida kengayib bormoqda.

CSSni HTML xujjatni ichida ishlatilishini 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 RELq “STYLE SHEET” TYPEq“TEXT/CSS” HREFq“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.

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

```
<head>< STYLE TYPEq"text/css">
```

```
<!--
```

```
A {text-decoration:none; }
```

```
-->
```

```
</--STYLE></head>
```

2. 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 berilmaydi, chunki bu usul kodlarni chalkash va tushunarsiz bo'lishiga olib keladi. Shoshilinch vaziyatlarda yoki stil qay tartibda namoyon bo'lishini ko'rib olish uchun bu usuldan foydalanish mumkin.

Undan tashqari <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, xar hil elementlar uchun bir hil tipdagi 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 hil element uchun bir necha stillarni birlashtirish mumkin.

H2 {font-weight: bold}

H2 {font-size: 14pt}

H2 {font-family: Verdana}

Barcha stillarni bittaga yig'amiz.

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 (po umolchaniyu) jarayonini hosil qilish mumkin. Shunda biror elementga stil berish esdan chiqib qolsa, <body> tegida ta'luqli stil esdan chiqqan elementga o'rnatiladi.

```
BODY { color: green;
        font-family: "Verdana";
        background: url(joke.gif) white; }
```

Selektor, Klass, Identifikator tushunchasi.

Selektor. CSS da selektor vazifasini HTML teglar bajaradi. Web sahifadagi barcha bir hil tipdagi teglarga (selektorlarga) umumiy stil beramiz. Misol uchun saytdagi barcha "havola"larning tagiga chizishni olib tashlaymiz (sizga ma'lumki, "havola" larga stil berilmasa odatiy tegiga chizilgan holatda qoladi).

<STYLE TYPE"text/css">

A {text-decoration:none; }
</STYLE>

Klass. Agar bizga, kod ichida foydalaniladigan bir xil teglarga har xil stillarni o'rnatishimiz lozim bo'lsa, klass yoki identifikatorlardan foydalanish juda qulay hisoblanadi. Dastlab klass haqida to'xtalib o'tamiz. Bir xil elementlarga har xil stil berish uchun, element nomi va nuqta (.) qo'yib, stilga biror nom beramiz. So'ng shu nomga tegishli bo'ladigan stillar ketma-ketligini yozib chiqamiz. Shundan so'ng, kod ichida kerakli bo'lgan joyda, **class** so'zi yordamida o'rnatilgan stil nomi chaqiriladi.



3.1.1-misol. Klass stillar

```
<head>  
<style TYPEq"text/css">  
H1.rd {color:red;}  
H1.be {color:red;background-color:blue;}  
</style>  
</head>  
E`lon qilingan stilni kerakli joyda chaqiramiz.  
<body>  
<h1 classq"rd">Qizil shrift</h1>  
<h1 classq"be">Ko'k fonda qizil shrift</h1>  
</body>
```



3.1.1-rasm

Shu tariqa istalgan elementga istalgan stillarni o'rnatish mumkin bo'ladi. Yuqoridagi misolda faqat bitta element(H1) uchun stil yaratdik. Agar biz biror stilni yaratib, uni istalgan element uchun ishlatmoqchi bo'lsak, u holda yaratilgan stilga element nomini qo'ymasdan, faqatgina nuqta orqali stil uchun doimiy nom beramiz va kerakli joyda chaqiramiz. Quyidagi misolni yaxshilab tahlil qilib chiqing.

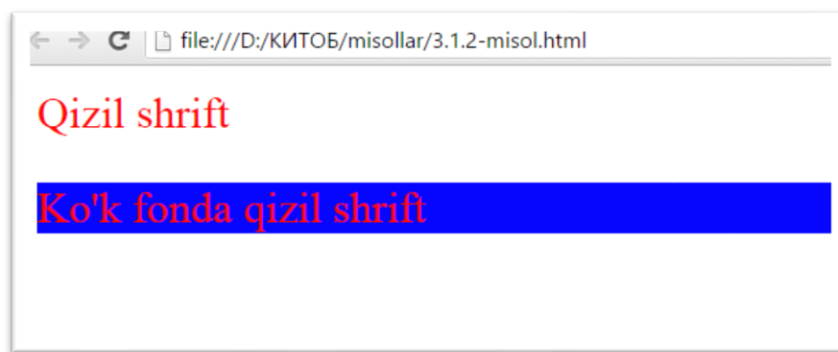


3.1.2-misol. Klass stillar

```
<STYLE TYPEq"text/css">  
.red {color: red; }  
.blue {color:red; background-color: blue}  
</STYLE>
```

Biz 2 ta stil yaratdik, "**red**" va "**blue**" deb nomladik, endi bu stillarni istalgan element uchun ishlatishimiz mumkin.

```
<P CLASSq"red">Qizil shrift</P>  
<P CLASSq"blue">Ko'k fonda qizil shrift</P>
```



3.1.2-rasm

Identifikator. Bu o'z ismi bilan etib turilibdi **IDENTIFIKATOR**, ya'ni butun web sahifasida faqatgina bitta elementga stil berish uchun ishlatiladi. Dastlab stil yozilib, unga “#” belgisi orqali nom beriladi. “#” bu identifikatorli stillar to'plami degan ma'noni beradi.

```
#matn {font-family:verdana; color:green;}
```

So'ng kod ichida bu identifikator **ID** kalit so'zi orqali chaqiriladi va yagona element uchun ishlatiladi. Bu ta'rif identifikatorga berilgan haqiqiy ta'rifdir. Lekin hozirgi kundagi ko'p brouzerlarda id orqali bir necha elementga ham stil berish mumkin bo'lib qoldi, brouzerlar bunga ruhsat berishmoqda.



3.1.3-misol. Identifikator stillar

```
<STYLE TYPEq"text/css">  
#myID {letter-spacing: 1em; }  
H1#form3 {color:red; background-color: blue}  
</STYLE>  
<BODY>  
<P IDqmyID> 1em lik yozuv</P>  
<H1 IDqform2>Qora rangda</P>
```



3.1.3-rasm

Yuqoridagi kodda "form3" identifikatori e'lon qilindi, lekin "form2" identifikatori ishlatildi. Bunda "form2" bo'lmaganligi sababli yozuv odatiy holda namoyon bo'ldi.

Hozirgi kunda identifikatordan klassda yaratilgan stillarning ma'lum bir qismini o'zgartirish uchun ishlatishmoqda. Misol uchun, bitta klass ochib, unga nom beramiz va stillar ketma-ketligini kiritamiz. HTML kod ichida shu klassni

chaqirib, undan foydalanamiz, agar bizga stillar ketmaketligini biror elementini o'zgartirish kerak bo'lib qolsa, identifikatorni ishga solamiz. Shunda butun stilni bitta elementini o'zgartirish uchun boshidan yozib o'tirmaymiz. HTML tegida klass va identifikatorni birga ishlatsak, identifikator kuchliroq hisoblanadi. Misol ko'ramiz.



3.1.4-misol. Klass va identifikator stillar

```
<style>
.klas{color:blue; font-family:verdana;}
#iden{color:red;}
</style>
<p classq"klas" idq"iden">test_klas</p>
<p idq"iden">test_klas</p>
```



3.1.4-rasm

Ikkala holatda ham matn qizil rangda chiqadi, identifikatorda o'rnatilgani kabi. Oxirgi berilgan stil natijaga ta'sir qiladi.

Sintaksis va ishlash prinsipi: ID, CLASS, STYLE. Oldingi mavzulardan CSS haqida biroz bo'lsa ham ma'lumotga ega bo'ldik. CSS kodlarini HTML kodlarida ishlatishni ko'rsatib o'tamiz. CSS natijalarini brouzerda ko'rish mumkin bo'ladi.

CSS stillar quyidagicha yoziladi:

H1{color: blue; font-size:14 px}

1. **Selektor nomi** - yozilgan stilni qaysi tegda foydalanish(misolda **H1**).
2. **Qavs** - stillar boshlanganini bildiradi.
3. **Stil xususiyati** - qaysi stil xususiyatidan foydalanishni o'rnatish (misolda **color**, **font-size**).
4. **Qiymat** - tanlangan stil xususiyat qiymati(misolda **blue**, **14 px**).

Agar bir necha stil xususiyatlaridan foydalaniladigan bo'lsak, ular nuqtali vergul (;) orqali ajratiladi, oxirida nuqtali vergul qo'yish shart emas. Stil xususiyatlari yig'indisi shaklli qavs ichiga olinishi shart({ }) va har bir stil xususiyatidan so'ng, uning qiymatini berishdan oldin ikki nuqta(:) qo'yilishi kerak.

Sintaksisi bilan tanishib chiqdik, endi HTML kodga ulashni ko'rsatib o'tamiz. Bir necha xil usulda ulash mumkin:

1. CSS stillarini alohida faylga yozib, bu faylni HTML kodida chaqirib ishlatish mumkin.

Misol uchun, "**style.css**" nomli fayl yaratib ichiga quyidagi stil xususiyat va qiymatlarini kiritamiz.

```
h1{color:blue;font-size:14px}
```

Undan so'ng, "**index.html**" fayl yaratib, bu faylni yuqorida yaratgan "**style.css**" nomli fayl bilan bitta papkaga joylashtiramiz, so'ng bu faylga quyidagi HTML teglarni kiritamiz.



3.1.5-misol. CSS bilan ishlash

```
<html>
<head>
<title>CSS bilan ishlash</title>
<link rel="stylesheet" type="text/css" href="style.css" >
</head>
<body>
<h1>Hello World CSS!</h1>
</body>
</html>
```



3.1.5-rasm

2. HTML hujjat ichida stil xususiyatlarini aniqlashtirib qo'yish mumkin.



3.1.6-misol. HTML hujjat ichida stil xususiyatlarini berish

```
<html>
<head>
<title>CSS bilan ishlash</title>
<style type="text/css">
h1{color:blue;font-size:14px}
</style>
</head>
<body>
<h1>Hello World CSS!</h1>
</body>
</html>
```



3.1.6-rasm

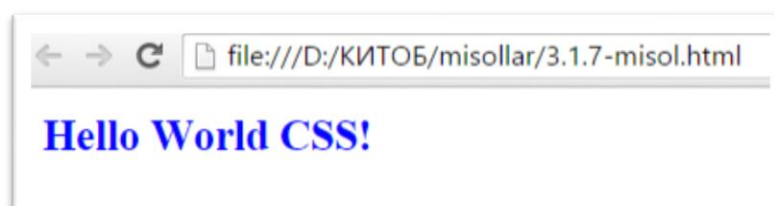
Natija, yuqoridagi bilan bir xil, bunda bitta tegga bitta stil berildi, agar o'zgarishlar qilinmasa, shu teg qaerda ishlatilsa, hamma yerda shu stil dan foydalaniladi.

3. Identifikator orqali bog'lash. O'zgarmas nomga ega bitta stillar xususiyati yoziladi va u faqat bitta element uchun ishlatiladi. Ishlatishda "**ID**" kalit so'zidan foydalaniladi, identifikator esa "#" belgisi bilan e'lon qilinadi.



3.1.7-misol. Identifikator orqali bog'lash

```
<html>
<head>
<title>CSS bilan ishlash</title>
<style type="text/css">
#matn{color:blue;font-size:14px}
</style>
</head>
<body>
<h1 id="matn">Hello World CSS!</h1>
</body>
</html>
```



3.1.7-rasm

4. Klass orqali bog'lash. Bitta klass ichida bir necha stil xususiyatlari aniqlashtirib olinadi, so'ng "**class**" kalit so'zi orqali chaqirilib foydalaniladi, nuqta(.) orqali e'lon qilinadi. Identifikatordan farqli ravishda, bir yoki bir necha element orqali ishlatish mumkin.



3.1.8-misol. Klass orqali bog'lash

```
<html>
<head>
<title>CSS bilan ishlash</title>
<style type="text/css">
.matn{ color:red; font-size:14px}
</style>
</head>
<body>
<h1 class="matn">Hello World CSS!</h1>
<h3>Hello World CSS!</h3>
<h3 class="matn">Hello World CSS!</h3>
```

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



3.1.8-rasm

5. HTML kodi ichida (**body** tegi orasida) bog'lanish, istalgan HTML kodi ichida, tegning "**style**" atributi orqali stil yaratish mumkin, bu stil faqat shu elementga xos bo'lib, boshqa element uchun ishlatib bo'lmaydi.



3.1.9-misol. HTML teg ichida stil berish

```
<html>
<head>
<title>CSS bilan ishlash</title>
<style type="text/css">
.matn{ color:red;font-size:14px}
</style>
</head>
<body>
<h1 style="color:green; font-family:verdana">Hello World CSS!</h1>
</body>
</html>
```



3.1.9-rasm



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

8. CSS (Cascading Style Sheets) stillar jadvali nima?
9. Stillar web sahifada qanday beriladi?
10. Stillarni yozilish qoiadalari qanday?
11. Klass selektorlar qanday ishlatiladi
12. Id selektorlar qanday beriladi?

3.2-§. Selektorlar va ulardan foydalanish. Havola selektorlari.



O`quv modullari

selektorlar, teglar ichida stillarni berish, stillarni qo'llanilishida ustunlik, CSS da o'lchov birliklari, havolalar bilan ishlash

Sahifaga stilnini uchta usul bilan berish mumkin: HTML tegdagi style hususiyati orolig'iga joylashtirish, HTML hujjatining sarlavhasida <style> tegi orasiga stillar jadvalini yozib, HTML tanasidaghi teglarda undan foydalanish yoki alohida faylda stillarni yozish va HTML hujjatga faylni biriktirib undagi teglarda qo'llash.

Teglar ichida stillarni berish. Stil ta'rifi style parametridan foydalangan holda har qanday tegga o'rnatiladi.

Style parametrlarini esda tuting.

```
<font styleq"font-size: 12pt">Matn</font>
```

Agar stil ta'rifi bir nechta atributlardan iborat bo'lsa, demak ular nuqtali vergul yordamida ajratib yoziladi:

```
<font styleq"font-size: 12pt; color: red">Matn</font>
```

Agar atributdan birortasi qiymati qo'shtirnoq belgilarisini talab qilinsa, u buyruq apostroflarda yozilishi mumkin:

```
<font styleq"font-size: 12pt; color: red; font-family:'Times New Roman' "> Matn</font>
```

HTML hujjat sarlavhasida stillarni yozish. Barcha zarur stillarni bir joyda to'plash mumkin. Bunday holda stillar <style> va </style> teglari orasida ko'rsatiladi. <style> tegining o'zi HTML-hujjatning HEAD bo'limida joylashgan bo'lishi kerak. Stillarni o'zi esa teglardagi class parametri yordamida ushbu tegga stil nomini ko'rsatish bilan biriktiriladi.



3.2.1-misol. Stildan foydalanish misoli

```
<!DOCTYPE HTML>
```

```
<html> <head>
```

```
<title> Stildan foydalanish misoli </title>
```

```
<meta http-equiv"Content-Type" contentq"text/html; charsetutf-8">
```

```
<style typeq"text/css">
```

```
.text1 {font-size:12pt; color:red; font-family:"Arial"}
```

```
font {font-size:12pt; color:green; font-family:"Arial"} font.text2{font-size:12pt;color:blue;font-family:"Arial"}
```

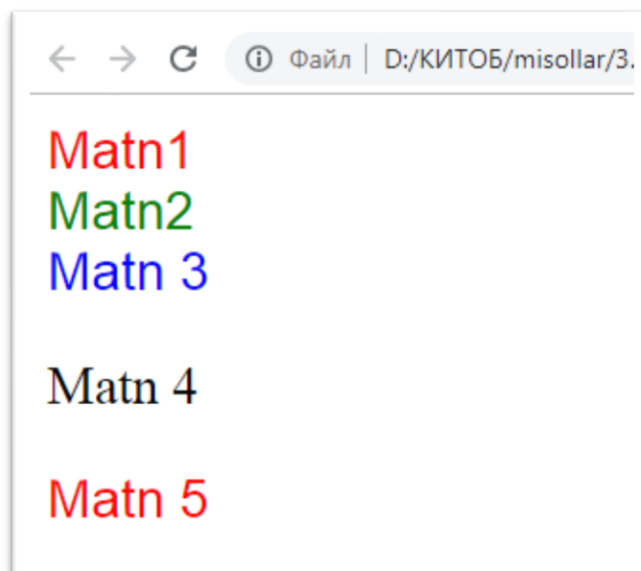
```
</style> </head>
```

```
<body>
```

```

<font classq"text1">Matn1</font><br> <!--Qizil matn-->
<font>Matn2</font> <br> <!--Yashil matn-->
<font classq"text2"> Matn 3</font><br> <!--Ko 'k matn-->
<p classq"text2"> Matn 4</p> <!--Ixtiyoriy matn-->
<p classq"text1"> Matn 5</p> <!-- Qizil matn -->
</body>
</html>

```



3.2.1-rasm

Stilni aniqlash atributlari <style> va </style> teglari orasida gulli qavs ichida ko'rsatiladi. Agar atributlar bir nechta bo'lsa ular nuqtalari vergul bilan ajratiladi:

```

<Selektor>{
<Atribut 1>:<Qiymat1>;
.....;

```

```

<Atribut N>:<QiymatN>}

```

<Selektor> sifatida quyidagi selektorlar islatilishi mumkin:

- * — barcha teglar. Barcha tashqi va ichki siljishlarni o'chiramiz:


```
* {margin:0; padding:0}
```
- Teg — ko'rsatilgan nomdagi barcha teglar:


```
font{font-size:12pt; color:green; font-family: "Arial"} <font> Matn 2 </font>
<!--Yashil matn-->
```
- . Klass — klass nomi bilan ko'rsatilgan barcha teglar:


```
.text1{font-size:12pt; color:red;font-family:"Arial"}
<font classq"text1"> Matn1</font> <!--Qizil matn--> <p
classq"text1"> Matn2 </p> <!--Qizil matn -->
"Matn1" va "Matn2" boshqa-boshqa teglar orasida joylashgan bo'lishiga
qaramay bir hil qizil rangda chop etiladi;
```
- Teg.Klass — Ko'rsatilgan klass nomidagi barcha teglar:


```
font.text2 {font-size:12pt; color:blue }
<font classq"text2">Matn1</font><!--Ko'k matn-->
```

Agar klass nomi boshqa tegda ko'rsatilsa u amal qilmaydi:

`<p classq"text2">Matn2</p>`

Bu holda “Matn2” matn qismi ko’k rangda chop etilmaydi. Chunki text2 sinf nomi faqat `<font>` tegidagini amal qiladi holos;

- #identifikator — identifikator nomi bilan ko’rsatilgan teg:

`#txt1 {color: red}`

`<p idq"txt1"> Matn</p>`

- teg#identifikator — aniqlangan tegda ko’rsatilgan identifikator:

`p#txt1 {color:red}`

`<p idq"txt1">Matn</p>`

Agar identifikator boshqa tegda ko’rsatilsa inobatga olinmaydi.

Stillar klasini birdaniga bir nechta tegga bog’lash mumkin. Bu holda selektorlar vergul bilan ajratib yoziladi:

`h1, h2 {font-family: "Arial"}`

Boshqa elementlarga biriktirish quyidagi usullar bilan bajariladi:

- selektor1 selektor2 — selektor2 parametriga mos keladigan konteyner ichida joylashgan selektor1 parametriga mos keladigan barcha elementlar:

`div a {color: red}`

Agar `<a>` teg ichida bo’lsa matn qizil bo’ladi.

`<div>:`

`<div><a hrefq"link.html"> havola </a></div>`

- selektor1 > selektor2 — selektor2 parametriga mos keladigan va selektor1 parametriga mos konteyneri ichida joylashgan barcha elementlar:

`div > a {color:red}`

Agar `<a>` tegi `<div>` tegi ichida joylashgan bo’lsa havola matni rangi qizil bo’ladi, aksincha boshqa teg ichida unday bo’lmaydi:

`<div>`

`<a hrefq"link1.html">havola 1</a><br>`

`<span> <a hrefq"link2.html">havola 2</a></span> </div>`

Ushbu misolda *Havola 2* `<span>` tegida joylashganligi uchun faqat *Havola 1* qizil rangda chiqadi;

- Selektor1 Q Selektor2 — Selektor1ga mos keluvchi konteyneiga bilan qo’shni Selektor2 ga mos element:

`div Q a { color: red }`

Agar `<a>` tegi `div` elementidan keyin kelsa havola matni rangi qizil bo’ladi:

`<div>Matn</div> <a hrefq"link.html">Havola</a>`

Zaruratga qarab bir nechta selektorlardan iborat ifodadan quyidagicha foydalanilganda:

`div span a { color: red }`

Agar `<a>` tegi `<span>` tegi ichida bu teg o’z navbatida `<div>` tegi ichida joylashgan bo’lsa matn rangi qizil bo’ladi:

`<div>`

`<a hrefq"link1.html">Havola 1</a><br> <span>`

`<a hrefq"link2.html"> Havola 2</a><br> </span> </div>`

Ushbu misolda faqat *Havola 2* matni qizil rangda chop etiladi.

Teg parametrlariga bog'lash uchun quyidagi selektorlar ishlatiladi:

- [parametr] — ko'rsatilgan parametrlarga ega elementlar:
`a[id] {color: red}`
Agar `<a>` tegi id parametriga ega bo'lsa havola matni rangi qizil bo'ladi:
`<a idq"link1" hrefq"link1.html"> Havola 1</a>`
- [Parametrq'Qiymat'] — parametri ko'rsatilgan qiymatga teng bo'lgan elementlar:
`a[hrefq"link1.html"] {color: red}`
Agar `<a>` tegining href parametri "link1.html" qiymatiga teng bo'lsa havola matni qizil rangda bo'ladi;
- [Parametr^q'Qiymat'] — parametri ko'rsatilgan qiymat bilan boshlanuvchi elementlar: `a[href*q"li"] {color: red}`
Agar `<a>` tegining href parametrining qiymati "li" bilan boshlansa havola matni rangi qizil bo'ladi;
- [Parametr\$q'Qiymat'] — parametri ko'rsatilgan qiymat bilan yakunlanadigan elementlar:
`a[href$ q ".html"] {color: red}`
Agar `<a>` tegining href parametri qiymati ".html" kengaytmasi bilan tugasi havola matni rangi qizil bo'ladi;
- [Parametr*q'Qiymat'] — Parametrda ko'rsatilgan qiymat qismini o'z ichiga olgan elementlar:
`a[href* q "link"] {color: red}`
Agar `<a>` tegining href parametri qiymatida "link" matni uchrasa havola matni qizil bo'ladi .
Selektor sifatida quyidagi psevdoelementlarni ham ko'rsatish mumkin:
- :first-letter — Birinchi harf uchun stil o'rnatadi. Barcha abzatlardagi birinchi harflarni belgilaymiz:
`p:first-letter {font-size:150%; font-weight:bold; color:red }`
- :first-line — Birinchi qator uchun stil o'rnatadi. Maslan:
`p:first-line {font-weight: bold; color:red}`
- :before va :after — mos ravishda elementning boshiga va oxiriga matn qo'shish imkonini beradi. Qo'shiladigan matn content: atributida ko'rsatilishi kerak:
`p:before {content: "before ";} p:after {content: " after";}`
`<p>Matn</p>`

Natija: before Matn after

Stillar jadvalining alohida faylda berilishi. Stillar jadvalini alohida faylda berish mumkin. Stillar jadvali odatda css kengaytmali faylda saqlanadi va uni ixtiyoriy matn muxarriri masalan Bloknout da taxrirlash mumkin. Fayl kengaytmasini esa html kengaytmasi bilan saqlash kabi css ko'rinishda saqlanadi.

Stillar jadvalini alohida style.css faylida yozilishi(3.2.2-misol) va uni asosiy test.html fayliga bog'lashni ko'rib chiqamiz (3.2.3-misol).



3.2.2-misol. style.css fayli mazmuni

```
.text1
{
    /*classq"text1" elementi uchun stil*/
    font-size: 12pt;      /* Shirift o'lchami*/
    color: red;           /* Matn rangi*/
    font-family: Arial;   /* Shirift nomi*/
}

font {
    /*Barcha FONT teglari uchun stil*/
    font-size:12pt;      /* Shirift o'lchami */
    color:green;         /* Matn rangi*/
    font-family: Arial;   /* Shirift nomi */
}

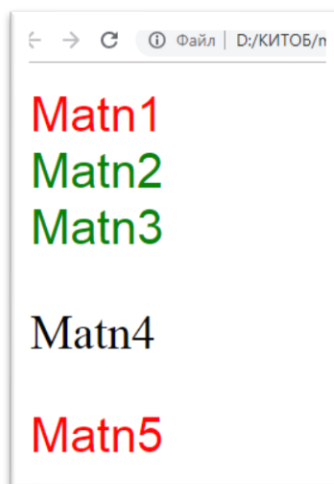
font.text2{ { /* classq"text2" sinfli FONT tegi uchun stil*/
font-size: 12pt; /* Shirift o'lchami*/
color: blue; /* Matn rangi*/
font-family: Arial /* Shirift nomi */ }
```



3.2.3-misol. test.html fayli mazmuni

```
<!DOCTYPE HTML>
<html> <head>
<title> </title>
<meta http-equivq"Content-Type" contentq"text/html; charsetqutf-8">
<link relq"stylesheet" typeq"text/css" hrefq"style.css">
</head>
<body>
<font classq"text1">Matn1</font><br><!--Qizil rangli matn-->
<font>Matn2</font><br><!--Yashil rangli matn-->
<font classq"text2">Matn3</font><br> <!--Ko'k rangli matn -->
<p classq"text2">Matn4</p><!--Ixtiyoriy rangdagi matn-->
<p classq"text1">Matn5</p><!--Qizil rangli matn-->
</body>
</html>
```

Har ikkilala faylni bir papkaga saqlaymiz va test.html faylini Web-brauzerda ochamiz.



3.2.2--rasm

Alohida stillar jadvali fayli HTML hujjatga `<link>` tegi yordamida biriktiriladi. Uning href parametrinda faylningabsolyut yoki nisbiy URL-adresi, rel parametrinda esa stylesheet qiymati ko'rsatilishi kerak. Stillar jadvali HTML hujjatga quyidagicha biriktiriladi:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

Tashqi CSS faylga bo'glanishni shuningdek quyidagicha qoida bo'yicha ham amalga oshirilishi mumkin `@import`:

```
@import url(URL-adress) [<Qurilma turi>;
```

```
@import <URL-adress> [<Qurilma turi>;
```

`@import`: qoidasi `<style>` tegi orasiga joylashgan bo'lishi kerak:

```
<style type="text/css"> @import url("style.css"); </style>
```

Majburiy bo'lmagan `<Qurilma turi>` parametri stillar jadvali bog'lanishi mo'ljallangan qurilmani ko'rsatishi mumkin. Masalan, `all` — barcha qurilmalar uchun, `print` — oldindan ko'rib chiqish va hujjatni chop etish uchun. Masalan:

```
<style type="text/css"> @import "style.css" print; </style>
```

Alohida faylda yoziladigan stillar jadvali faylidan bir nechta HTML-hujjatda foydalanilishi mumkin.

Stillarni qo'llanilishida ustunlik. Aytaylik, `style` parametrindagi `color` atributi biror rangni, `<style>` tegida boshqa rang, alohida faylda yozilgan stil esa uchichi rangni ifodalagan bo'lsin (3.2.4-misol). Bundan tashqari `<font>` tegining `color` parametri to'rtinchi rangni ifodalasin (3.2.5-misol).



3.2.4-misol. style.css fayli mazmuni

```
p {color:red}
```



3.2.5-misol. test.html fayli mazmuni

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title> Stillarni bajarilishidagi ustunlik </title>
```

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<link rel="stylesheet" type="text/css" href="style.css">
<style type="text/css">
p{color: blue}
</style>
</head>
<body>
<p style="color:green">
<font color="yellow">Matn1 </font>
</p><br>
<p style="color:green">Matn2</p>
</body> </html>

```

“Matn1” matni rangi qanday bo’ladiyu va “Matn2” matni rangi qanday bo’ladi? Stillar bajarilishidagi ustunlik ushbu savolga javob beradi.



3.2.3-rasm

- ✓ Blokni formatlash HTML kodida belgilangan bo’lsa, stillar jadvali bilan belgilangan stil bekor qilinadi;
- ✓ tegning style parametrda boshqa stil ko’rsatilgan bo’lsa, <style> tegida ko’rsatilgan uslub bekor qilinadi;
- ✓ <style> tegida boshqa stil belgilangan bo’lsa, alohida faylda yozilgan stil bekor qilinadi;

Stillar jadvalining ushu ustuvorlik ketma-ketligi kaskadlilik deyiladi.

Boshqacha qilib aytganda, alohida faylda tasvirlangan stil eng past darajaga, eng obyektga yaqin belgilangan stil esa eng yuqori ustuvorlikka ega bo’ladi. Yuqoridagi misolda “Matn1” so’zi abzatsi tegining color parametrda ko’rsatoilgan stilni oladi va so’z jigarrang bo’ladi. “Matn2” so’zi abzasi esa style parametri stiulini oladi va so’z yashil rangida bo’ladi.

Bundan tashqari, identifikator orqali belgilangan stil, class orqali belgilangan stildan ustun turadi. Maslan:

```

<style type="text/css">
#id1 {color: red }
.cls1 {color: blue }
</style>
<p id="id1" class="cls1">Matn1 </p>

```

Bu misolda “Matn1” so’zi ko’k emas, qizl rangda chiqadi.

!important hususiyati yordamida ustuvorlikni o’zgartirish mumkin. Misolimizdagi style.css fayli mazmunini o’zgartiramiz .

p {color: red !important}

Natijada stillar ustuvorligi qayta belgilanadi va “Matn2” so’zi yashil emas, qizil rangda chop etiladi. Biroq, “Matn1” so’zi tegi color parametrda ko’rsatilganidek sariq rangda qoladi.

CSS da o’lchov birliklari

CSS da o’lchamlarni aniq yoki nisbiy bo’ladi birliklarda ifodalash mumkin.

- ✓ piksel (px);
- ✓ millimetr(mm);
- ✓ santimetr (cm);
- ✓ dyum (in) — 1 in q 2.54 sm;
- ✓ punkt (pt) — 1 pt q 1/72 in;
- ✓ pika (pc) — 1 pc q 12 pt.
- ✓ Nisbiy birliklar:
- ✓ foiz(%);
- ✓ joriy shirift balandligi(em);
- ✓ Joriy shirift "x" harfi balandligi (ex).
- ✓ 0 qiymati uchun o’lchov birliklarini ko’rsatmaslik mumkin. Rang quyidagi usullardan birida ifodalanishi mumkin:
- ✓ inglizcha rang nomi bilan – blue, green va boshqa.: *p{color:red}*
- ✓ #[R][G][B]- ko’rinishdagi formatda. Bunda R– qizilga to’yinish, G–yashilga to’yinish, B–ko’k rangga to’yinish darajasi qiymati. Qiymatlar 0 dan F gacha 16 sanoq tizimidagi bir xonali sonlar bilan ifodalanadi: *p {color: #F00}*
- ✓ #[RR][GG][BB]- ko’rinishdagi formatda. Bunda RR– qizilga to’yinish, GG–yashilga to’yinish, BB–ko’k rangga to’yinish darajasi qiymati. Qiymatlar 00 dan FF gacha 16 sanoq tizimidagi ikki xonali sonlar bilan ifodalanadi: *p {color:#FF0000}*
- ✓ rgb([R],[G],[B])– ko’rinishdagi formatda. Bunda R, G, B–mos ravishda qizil, yashil va ko’k rangga to’yinish darajasini ifodalovchi 0 dan 255 gacha o’nlik sanoq tizimidagi sonlar: *p {color: rgb(255, 0,0)}*
- ✓ rgb([R%], [G%], [B%])- ko’rinishdagi formatda. Bunda R%, G%, B%–mos ravishda qizil, yashil va ko’k rangga to’yinish darajasini foizlarda ifodalanishi: *p {color:rgb(100%, 0%, 0%)}*

Yuqorida keltirilgan barcha misollarda qizil rang hosil bo’ladi.

Havolalar bilan ishlash. Biror bir saytga kirsangiz, u yerdagi havola(ssilka)ga sichqonchani olib borsangiz, bu havola o’z ko’rinishini o’zgartirishini ko’rgan bo’lsangiz kerak, ya’ni tagiga chizadi yoki havola rangi o’zgaradi va hokazo. Bular aynan CSS orqali amalga oshiriladi, to’g’riroq aytadigan bo’lsak, psevdoklasslar bu ishni amalga oshirib beradi.

Havolalar uchun quyidagi 4 ta psevdoklass ishlatiladi:

- **link** - havolaning doimiy ko’rinishi;

- **active** - havolaning aktiv holatdagi(bosilayotgandagi) ko'rinishi;
- **visited** - havolaga kirilgandagi holati;
- **hover** - sichqoncha havolaga olib borilgandagi ko'rinishi.

Psevdoklasslar quyidagicha ishlatiladi:

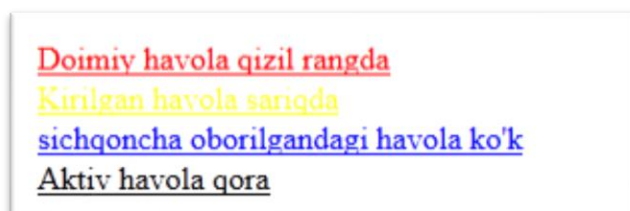
A: *psevdoklass nomi{...stil...}*



3.2.6-misol. Havolalar bilan ishlash

```
<html>
<head>
<title>Havola</title>
<style>
a:link{color:red;}
a:active{color:black;}
a:hover{color:blue;}
a:visited{color:yellow;}
</style>
</head>
<body>
<a hrefq"#1">Doimiy havola qizil rangda</a> </br>
<a hrefq"#2">Kirilgan havola sariqda</a> </br>
<a hrefq"#3">sichqoncha oborilgandagi havola ko'k</a> </br>
<a hrefq"#4" >Aktiv havola qora</a> </br>
</body>
</html>
```

Natijada hamma stil xususiyatlarini ishlatib chiqardim, ya'ni rang barang bo'lishi uchun



3.2.4-rasm

Ranglardan tashqari boshqa atributlarni ham qo'llash mumkin:

```
a:link {
color: blue ;
}
a:visited {
color:gray;
}
a:hover {
color:red ;
text-decoration:none;
```

```
font-weight:bold;
}
a:active {
color:green;
text-decoration:none;
text-transform:uppercase;
}
```

Agar havolaning barcha holatlari bir xil bo'lishini hohlasangiz quyidagicha yozishingiz mumkin.

```
a:link, a:visited, a:hover, a:active {color:black; text-decoration:none;}
```

Yuqoridagi holatni psevdoklassiz ham amalga oshirish mumkin.

```
a {color:black; text-decoration:none;}
```

Xullas, havolalarning tashqi ko'rinishi CSS orqali boshqarilar ekan.



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. Selektor nima?
2. Selektorlar qanday turlarga bo'linadi?
3. Selektorlar qanday ko'rinishlarda yozilishi mumkin?
4. Teglar ichida stillar qanday beriladi?
5. Stillar web sahifa ichida qanday yoziladi?
6. Stillarni alohida faylda berish qanday amalga oshiriladi?
7. Stillarni qo'llashda ustuvorlik qanday?
8. CSSda qanday o'lchovlar qo'llaniladi?
9. Havolalar stili qanday beriladi?

3.3-§. CSSda shriftlar bilan ishlash. CSS da matn va ro'yxatlar bilan ishlash. Blok elementlari umumiy konsepsiyalari.



O`quv modullari

shriftlarni formatlash, matnni formatlash, ro'yxatlar bilan ishlash, blok elementlari

Shriftlarni formatlash. Kaskadlandan stillar jadvali shirift nomi, rangi, o'lchami va uning stilini o'rnatish imkonini beradi. Bundan tashqari bir nechta shirift nomini ularga muqobil oila shirift oilasi nomi bilan almashtirilishi mumkin. Axir foydalanuvchi kompyuterida hamma shirift bo'lmasligi mumkin.

font-family atributi shirift nomini o'rnatish imkonini beradi:

```
p {font-family: "Arial"}
```

Ba'zi hollarga shirift foydalanuvchi kompyuterida mavjud bo'lmasligi mumkin. Shuning uchun eng yaxshisi bir nechta muqobil shiriftlar ko'rsatishdir. Shirift nomlari vergul bilan ajratib yoziladi: `p {font-family: "Verdana", "Tahoma"}`
Shuningdek, beshta asosiy shiriftlar oilasi (serif, sans-serif, cursive, fantasy yoki monospace)dan birini ko'rsatish mumkin:
`p {font-family: "Verdana", "Tahoma", sans-serif}`

font-style atributi shirift stilini o'rnatish imkonini beradi. U quyidagi qiymatlarda qo'llanilishi mumkin:

- ✓ normal — oddiy shirift:
`p {font-family: "Arial"; font-style:normal}`
- ✓ italic — kursiv shirift:
`p {font-family: "Arial"; font-style: italic}`
- ✓ oblique — egma kursiv shirift:
`p {font-family: "Arial"; font-style: oblique}`

font-size atributi shirift o'lchamini o'rnatish imkonini beradi:

`.text1 { font-size: 12pt; font-family: "Arial" }`
Absolyut yoki o'zgarmas kattaliklardan birida qiymatni ko'rsatish mumkin — xx-small, x-small, small, medium, large, x-large yoki xx-large: `.text1 {font-size: large; font-family: "Arial"}`
Bundan tashqari nisbiy kattaliklarda (masalan, foizlarda) yoki larger yoki smaller konstantalaridan birida qiymatni ko'rsatish ham mumkin:
`.text1 {font-size: 150%; font-family: "Arial"}`
`.text2 {font-size: smaller; font-family:"Arial"}`

color atributi shirift rangini belgilash imkonini yaratadi:

`.text1 {font-size:12pt; font-family:"Arial"; color:red}`
`.text2 {font-size:12pt; font-family:"Arial"; color: #00FF00}`
`.text3 {font-size:12pt; font-family:"Arial"; color:rgb(255, 0,0)}`

font-weight atributi shirift qalainligini boshqarish imkonini beradi. Quyidagi qiymatlar ishlatilishi mumkin:

- ✓ 100, 200, 300, 400, 500, 600, 700, 800, 900 — 100 qiymati eng ingichka, 900 — esa eng qalin shiriftga mos keladi:
`p {font-family:"Arial"; font-style:italic; font-weight:700}`
- ✓ normal — normal shirift. 400 qalinlikdagi shiriftga mos keladi:
`p {font-family:"Arial"; font-style:italic; font-weight:normal}`
- ✓ bold — qalin shirift. 700 qalinlikdagi shiriftga mos keladi:
`p {font-family:"Arial"; font-style:italic; font-weight:bold}`

Matnni formatlash. Matn uchun shiriftni tavsiflashdan tashqari uning bir nechta qo'shimcha parametrlari — belgilar so'zlar va qatorlar oralig'idagi masofa hamda vertical va gorizontal tekislanishlar, birinchi qatordagi siljish oraligi kabilarni ham belgilash imkoni mavjud.

letter-spacing atributi matndagi belgilar oralig'i masofasini o'rnatadi. U quyidagi qiymatlarni olishi mumkin:

✓ normal — oddiy qiymat:

p { letter-spacing: normal; font-style: italic; font-weight: normal }

✓ CSS qo'llaydigan absolyut kattaliklar yordamida:

p {font-size: 12pt; color: red; letter-spacing: 5mm }

word-spacing atributi so'zlar oralig'idagi masofani o'rnatish uchun qollaniladi. U quyidagi qiymatlarni olishi mumkin:

✓ normal — oddiy qiymat:

p {word-spacing: normal; font-style: italic; font-weight: normal }

✓ CSS qo'llaydigan absolyut kattaliklar yordamida:

p { font-size: 12pt; color: red; word-spacing: 5mm }

text-indent atributi abzats uchun siljish masofasini o'rnatish uchun foydalaniladi. Siljish masofasini absolyut yoki nisbiy kattaliklardan foydalanib o'rnatish mumkin:

p {text-indent: 10mm; font-style: italic; font-weight: normal }

line-height atributi ikki qator orasidagi masofani o'rnatish uchun foydalaniladi. U quyidagi qiymatlarda qo'llanilishi mumkin:

✓ normal — oddiy qiymat:

p {line-height: normal; font-style: italic; font-weight: normal }

✓ CSS qo'llaydigan absolyut yoki nisbiy kattaliklar yordamida:

p {font-size: 12pt; color: red; font-family: "Arial"; line-height: 5mm }

text-align atributi matnni gorizontal tekislanishini o'rnatish uchun qo'llaniladi. Ushbu atribut quyidagi qiymatlarda qo'llanilishi mumkin:

✓ center — markazdan tekislash:

<p style="text-align: center">Markazdan tekislanuvchi abzas</p>

✓ left — chap tomondan tekislanish:

<p style="text-align: left"> Chap tomondan tekislanuvchi abzas </p>

✓ right — o'ng tomondan tekislanish:

<p style="text-align: right"> O'ng tomondan tekislanuvchi abzas</p>

✓ justify — har ikki tomondan tekislanuvchi abzas:

<p style="text-align: justify"> Har ikki tomondan tekislanuvchi abzas </p>

vertical-align atributi o'zi joylashgan obyektga – masalan jadval yasheykasiga nisbatan vertikal tekislanishni o'rnatadi. U quyidagi qiymatlarni qo'llanilaydi:

✓ baseline — asos bo'yicha:

td {font-size: 12pt; color: red; vertical-align: baseline }

✓ middle — markazdan:

td {font-size: 12pt; color: red; vertical-align: middle }

✓ top — yuqoridan:

td {font-size: 12pt; color: red; vertical-align: top }

✓ bottom — yuqorisidan:

td {font-size:12pt; color: red; vertical-align: bottom}

text-decoration atributi matnni ostiga, ustiga yoki ustidan chizish imkonini beradi. U quyidagi qiyjatlarda qo'llaniladi:

- ✓ none — odatiy matn (po umolchaniyu):
<p styleq"text-decoration: none">Matn</p>
- ✓ underline — ostiga chizilgan matn:
<p styleq"text-decoration: underline">Ostiga chizilgan matn</p>
- ✓ overline — matn ustiga chizish:
<p styleq"text-decoration:overline">Ustiga chizilgan matn</p>
- ✓ line-through — matn ustidan chizish:
<p styleq"text-decoration: line-through">Ustidan chizilgan matn</p>
- ✓ blink — matnni pirpiratish:
<p styleq"text-decoration: Pirpirovchi matn</p>

text-transform atributi belgi registrini o'zgartirish imkoninin beradi. U quyidagi qiyjatlarda qo'llaniladi:

- ✓ capitalize — har bir so'zning birinchi harfini bosh harfga o'zgartiradi;
- ✓ uppercase — barcha belgilarni bosh harfga o'zgartiradi;
- ✓ lowercase — barcha harflarni kichik harfga o'tkazadi;
- ✓ none — o'zgartirilmaydi. Masalan:

<h1 styleq"text-transform: capitalize">bir nechta so'zdan iborat sarlavha</h1> <h1 styleq"text-transform: uppercase">sarlavha2</h1>
<h1 styleq"text-transform: lowercase">SARLAVHA3</h1>

Natija:

Bir Nechta So'zdan Iborat Sarlavha

SARLAVHA2

sarlavha3

white-space atributi probellarni qayta ishlash turini o'rnatishga imkon beradi. Odatda bir nechta probellar Web brauzer oynasida birdaniga bitta probel sifatida chop etiladi. Atribut quyidagi qiyjatlarda qo'llaniladi:

- ✓ normal — matn odatiy ko'rinishda chop etiladi:

<p styleq"white-space: normal">

Qator 1

Qator 2

</p>

Web brauzer oynasidagi natija:

Qator 1 Qator 2

- ✓ pre — barcha probel va qator tashlashlar saqlanadi:

<p styleq"white-space: pre">

Qator 1

Qator 2

</p>

Web brauzer oynasidagi natija:

Qator 1

Qator 2

✓ nowrap — qatorlarni o'tkazishda HTML-kod inkor qilinadi. Agar qatorda

 tegi bo'lsa, qator shu joydan yangi qatorga o'tkaziladi:

```
<p style="white-space: nowrap">
```

Stroka 1

Stroka 2
 Stroka 3

```
</p>
```

Web brauzer oynasidagi natija:

Stroka 1 Stroka 2 Stroka 3

Endi stillarga oid bu xususiyatlarni misollar orqali ko'rib chiqamiz.

Text-align - matnni tekislash, HTML dagi "**align**" attribut kabi ishlaydi, qiymatlari ham bir xil:

left - chap tomonga tekislash(odatiy hol);

right - o'ng tomonga tekislash;

center - markazga tekislash;

justify - matn uzunligi bo'yicha tekislash.

```
h1{text-align:center;}
```

Text-decoration - matnni chizish kerak bo'lsa, shu xususiyat ishlatiladi, uning qiymatlari quyidagilar:

none - hech qanday chizishni amalga oshirmaydi;

underline - matnni tegiga chizish;

overline - matnni ustig chizish;

line-through - matnni chizib tashlash.

Barcha qiymatlarga misol ko'ramiz:



3.3.1-misol. text-decoration xususiyati

```
<html>
```

```
<head>
```

```
<title> Text-decoration xususiyati </title>
```

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

```
#a1{
```

```
text-decoration:none;
```

```
}
```

```
#a2{
```

```
text-decoration:underline;
```

```
}
```

```
#a3{
```

```
text-decoration:overline;
```

```
}
```

```
#a4{
```

```
text-decoration:line-through;
```

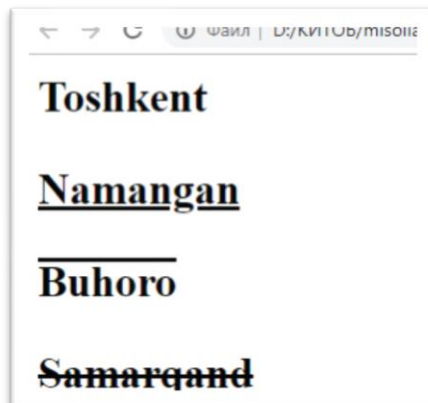
```
}
```

```

</style>
</head>
<body>
<h3 idq"a1">Toshkent</h3>
<h3 idq"a2">Namangan</h3>
<h3 idq"a3">Buhoro</h3>
<h3 idq"a4">Samarqand</h3>
</body>
</html>

```

Natija



3.3.1-rasm

Text-indent - birinchi qatorning boshida joy qoldirish, ya'ni abzasni o'rnatish uchun ishlatiladi. Qiymatni pikselida berish qulay va ishonchli.



3.3.2-misol. Text-indent xususiyati

```

<html>
<head>
<title> Text-indent </title>
<style typeq"text/css">
p {
text-align:justify;
text-indent: 40px;
}
</style>
</head>
<body>

```

<p> Xorazmiy dunyo faniga g'oyat katta hissa qo'shdi. U algebra fanining asoschisi bo'ldi. "Algebra" so'zining o'zi esa uning "Al-kitob al-muxtasar fi hisob al-jabr va al-muqobala" nomli risolasidan olingan.

Uning arifmetika risolasi hind raqamlariga asoslangan bo'lib, hozirgi kunda biz foydalanadigan o'nlik pozitsion hisoblash sistemasi va shu sistemadagi amallarning Ovro'poda tarqalishiga sabab bo'ldi. Olimning "al-Xorazmiy" nomi esa "algoritm" shaklida fanda abadiy o'rnashib qoldi. Uning geografiyaga doir

asari esa arab tilida o'nlab geografik asarlarning yaratilishiga zamin yaratdi.

</p>

</body>

</html>

Natija

Xorazmiy dunyo faniga g'oyat katta hissa qo'shdi. U algebra fanining asoschisi bo'ldi. "Algebra" so'zining o'zi esa uning "Al-kitob al-muxtasar fi hisob al-jabr va al-muqobala" nomli risolasidan olingan. Uning arifmetika risolasi hind raqamlariga asoslangan bo'lib, hozirgi kunda biz foydalanadigan o'nlik pozitsion hisoblash sistemasi va shu sistemadagi amallarning Ovro'poda tarqalishiga sabab bo'ldi. Olimning "al-Xorazmiy" nomi esa "algoritim" shaklida fanda abadiy o'mashib qoldi. Uning geografiyaga doir asari esa arab tilida o'nlab geografik asarlarning yaratilishiga zamin yaratdi. Xorazmiyning "Zij"i Ovro'poda ham, Sharq mamlakatlarida ham astronomiyaning rivojlanish yo'lini ko'rsatib berdi. Lekin afsuski, fanning bir necha tarmoqlariga asos solgan, "o'z davrining eng buyuk matematigi va agar barcha shart-sharoitlar nazarga olinsa, hamma davrlarning ham eng buyuklaridan biri" (J. Sarton) bo'lgan bunday siymoning hayoti haqida ma'lumotlar deyarli saqlanmagan.

3.3.2-rasm

Text-transform - matndagi yozuv harflarini katta-kichik qilib o'zgartirish stil xususiyati. 4 xildagi qiymatlarni qabul qiladi:

capitalize - matndagi har bir so'zning birinchi harfini katta harf qilib chiqaradi;

uppercase - matndagi barcha harflarni katta harfga almashtirib chiqaradi;

lowercase - matndagi barcha harflarni kichik harfga almashtirib chiqaradi;

none - registrni o'zgartirmaydi, ya'ni qanday yozilgan bo'lsa shunday chiqaradi.



3.3.3-misol. Text-transform xususiyati

<html>

<head>

<title>text </title>

</head>

<body>

<p style="text-transform:capitalize">Axmad al-Farg'oniy </p>

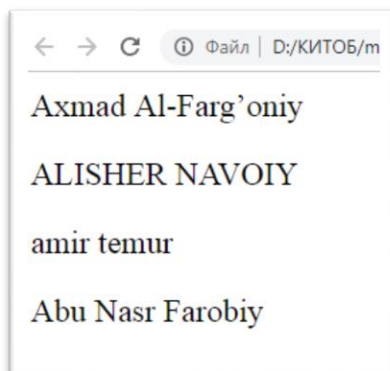
<p style="text-transform:uppercase">Alisher Navoiy</p>

<p style="text-transform:lowercase">Amir Temur</p>

<p style="text-transform:none">Abu Nasr Farobiy </p>

</body>

</html>



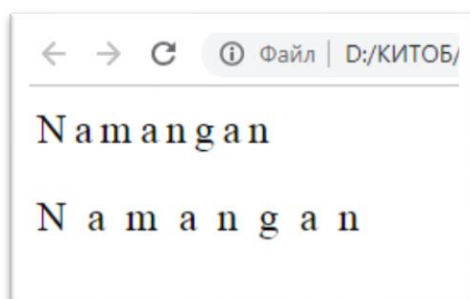
3.3.3--rasm

Letter-spacing - matndagi harflar orasidagi masofani o'rnatish, qiymatlarni pikselda beramiz.



3.3.4-misol. Letter-spacing xususiyati

```
<html>
<head>
<title> Letter-spacing </title>
</head>
<body>
<p styleq"letter-spacing:3px">Namangan</p>
<p styleq"letter-spacing:8px"> Namangan </p>
</body>
</html>
```



3.3.4-rasm

Word-spacing - matndagi so'zlar orasidagi masofani o'rnatish, albatta pikselda qiymatlarni beramiz.



3.3.5-misol. Word-spacing xususiyati

```
<html>
<head>
<title>Word </title>
</head>
<body>
<h3 styleq"word-spacing:2px">Namangan muhandislik qurilish instituti</h3>
<h3 styleq"word-spacing:16px">Namangan muhandislik qurilish
instituti</h3>
</body>
</html>
```



3.3.5-rasm

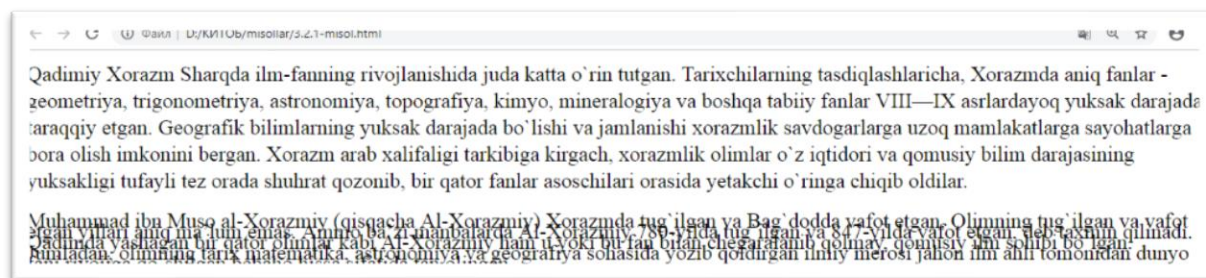
Line-height – qator orasidagi masofani oʻrnatish, pikselda qiymat kiritamiz.



3.3.6-misol. *Line-height* xususiyati

```
<html>
<head>
<title>Line </title>
</head>
<body>
<p styleq"line-height:20px">Qadimiy Xorazm Sharqda ilm-fanning rivojlanishida juda katta o`rin tutgan. Tarixchilarning tasdiqlashlaricha, Xorazmda aniq fanlar - geometriya, trigonometriya, astronomiya, topografiya, kimyo, mineralo-giya va boshqa tabiiy fanlar VIII—IX asrlardayoq yuksak darajada taraqqiy etgan. Geogra-fik bilimlarning yuksak darajada bo`lishi va jamlanishi xorazmlik savdogarlarga uzoq mamlakatlarga sayohatlarga bora olish imkonini bergan. Xorazm arab xalifaligi tarkibiga kirgach, xorazmlik olimlar o`z iqtidori va qomusiy bilim darajasining yuksakligi tufayli tez orada shuhrat qozonib, bir qator fanlar asoschilari orasida yetakchi o`ringa chiqib oldilar.</p>
<p styleq"line-height:8px"> Muhammad ibn Muso al-Xorazmiy (qisqacha Al-Xorazmiy) Xorazmda tug`ilgan va Bag`dodda vafot etgan. Olimning tug`ilgan va vafot etgan yillari aniq ma`lum emas. Ammo ba`zi manbalarda Al-Xorazmiy 780-yilda tug`ilgan va 847-yilda vafot etgan, deb taxmin qilinadi. Qadimda yashagan bir qator olimlar kabi Al-Xorazmiy ham u yoki bu fan bilan chegaralanib qolmay, qomusiy ilm sohibi bo`lgan. Jumladan, olimning tarix matematika, astronomiya va geo-grafiya sohasida yozib qoldirgan ilmiy merosi jahon ilm ahli tomonidan dunyo fani rivojiga qo`shilgan bebaho hissa sifatida tan olingan.
</p>
</body>
</html>
```

Natija



3.3.6-rasm

White-space - so'zlar orasidagi probellarni qanday chiqarishni o'rnatish. Odatda HTML gippermattin tilida barcha probellar bitta probel ko'rinishida chiqadi, pre tegidan tashqari(pre tegi probellarni hisobga olib, shriftni ham sal o'zgartirib chiqaradi), "**white-space**" xususiyati "**pre**" tegi bajargan vazifani bajaradi, faqat shriftni o'zgartirmaydi.

"**White-space**"ning qiymatlari:

normal - oddiy ko'rinish, matndagi probellar bitta probel ko'rinishida chiqadi va so'zlar keyingi qatorga avtomat o'tadi;

nowrap - matndagi probellar hisobga olinmaydi, so'zlar avtomat keyingi qatorga o'tmaydi, "br" tegi orqali keyingi qatorga o'tish kerak;

pre - matndagi probellar va keyingi qatorga o'tishlar hisobga olinadi, agar kodda keyingi qatorga o'tish bo'lmasa, brauzerda o'tkazgich(prokrutka) paydo bo'ladi;

pre-line - matndagi probellar hisobga olinmaydi, belgilangan chegaradan so'zlar chiqib ketsa, avtomat keyingi qatorga chiqadi;

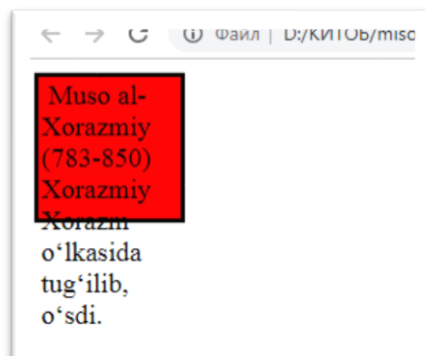
pre-wrap - matndagi probellar va keyingi qatorga o'tishlar saqlanib qoladi, agar matn belgilangan joydan chiqib ketsa, avtomat keyingi qatorga o'tadi.



3.3.7-misol. *White-space* xususiyati

```
<html>
<head>
  <title> White-space </title>
  <style>
.sloy1 {
border:2px solid black;
padding:2px;
height:80px;
width:80px;
white-space:pre-wrap;
background-color:red;
}
  </style>
</head>
<body>
<div classq"sloy1"> Muso al-Xorazmiy (783-850)
```

```
Xorazmiy Xorazm o'lkasida tug'ilib, o'sdi. </div>
</body>
</html>
```



3.3.7-rasm

Yozuv matn ichiga sig'madi va keyingi qatorga o'tdi.

Direction - matnni chiqarish yo'nalishini o'rnatish, 2 xil qiymati mavjud:

ltr - chapdan o'nga qarab chiqarish;

rtl - o'ngdan chapga qarab chiqarish.



3.3.8-misol. *Direction* xususiyati

```
<html>
<head>
<title>Direction </title>
</head>
<body>
<h4 styleq"direction:ltr">Hello World!</h4>
<h4 styleq"direction:rtl">Hello World!</h4>
</body>
</html>
```



3.3.8-rasm

unicode-bdi - matnni qanday tartibda chiqarishni o'rnatish, qiymatlari:

normal - brauzer o'zi qanday chiqarishni hal qiladi;

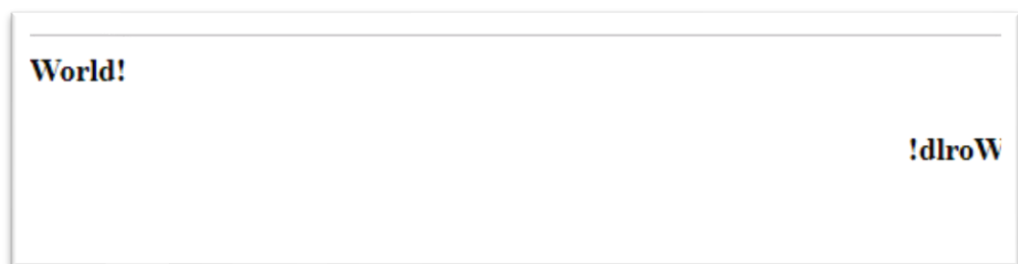
embed - "**direction**" xususiyatini qayta aniqlashtiradi, "**direction**"da qanday yozilgan bo'lsa shunaqa chiqadi;

bidi-override - biz uchun qiziqarli qiymat, simvollar ketma-ketligi tartibi o'zgartirib chiqaradi.



3.3.9-misol. unicode-bdi xususiyati

```
<html>
<head>
<title>Rang </title>
</head>
<body>
<h4 style="direction:ltr; unicode-bdi:embed">World!</h4>
<h4 style="direction:rtl; unicode-bdi:bidi-override">World!</h4>
</body>
</html>
```



3.3.9-rasm

Ro'yxatlar bilan ishlash

CSS stillar bilan ishlash tilida ham HTML gipermatn tiliga o'xshab ro'yxatlar tushunchasi mavjud. Ro'yxatlar uchun uncha ko'p stil xususiyatlari ajratilmagan, shu sababli maqolamiz qisqa bo'ladi, lekin kerakli xususiyatlar bilan tanishamiz. Ro'yxatlarni, CSS xususiyatlarini ishlatmasdan HTML gipermatn tilida ham hosil qilishingiz mumkin, lekin CSSdagidek chiroyli chiqmaydi, shuning uchun maslahat, ikkala hilni ham ko'rib chiqing va ma'qulidan foydalaning. Ro'yxatlar uchun quyidagi stil xususiyatlarini ishlatamiz:

- *list-style-type*;
- *list-style-position*;
- *list-style-image*;
- *list-style*.

Bu xususiyatlar universal bo'lib, tartiblangan va tartiblanmagan ro'yxatlar uchun ishlatilaveriladi. Bular orqali tartibli va tartibsiz ro'yxatlar hosil qilish mumkin, keling ular bilan tanishib chiqamiz.

List-style-type - ro'yxatlar uchun markerlarni aniqlashtirish uchun ishlatiladi, markerlar raqam, harf, kvadrat, dumaloq va boshqa ko'rinishlarda bo'lishi mumkin. Markerlarni aniqlashtirish uchun quyidagi qiymatlar ishlatiladi:

disk - marker sifatida bo'yalgan doira;

circle - marker sifatida bo'yalmagan aylana;

square - marker sifatida bo'yalgan kvadrat.

decimal - marker sifatida oddiy arab raqamlar: 1, 2, 3, 4, 5...

decimal-leading-zero - marker sifatida oddiy arab raqamlar "0" bilan: 01, 02, 03, 04, 05...

upper-roman - marker sifatida katta rim raqamlari: I, II, III, IV,...

lower-roman - marker sifatida kichik rim raqamlari: i, ii, iii, iv,...

upper-alpha - marker sifatida katta ingliz harflari: A, B, C, D,...

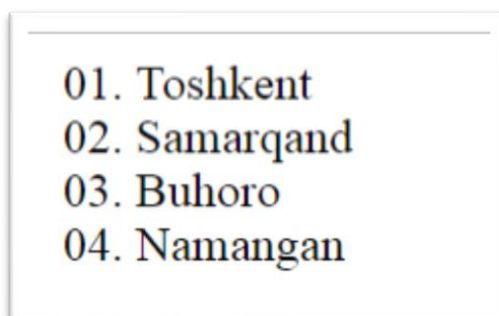
lower-alpha - marker sifatida kichik ingliz harflari: a, b, c, d,...

none - ro'yxatni markersiz qilish.



3.3.10-misol. *list-style-type* dan foydalanish misoli

```
<html>
<head>
  <title>List</title>
  <style>
    ol {
      list-style-type: decimal-leading-zero;
    }
  </style>
</head>
<body>
  <ol>
    <li>Toshkent</li>
    <li>Samarqand</li>
    <li>Buhoro</li>
    <li>Namangan</li>
  </ol>
</body>
</html>
```



3.3.10-rasm

List-style-position - bu stil xususiyati markerni joylashishini aniqlashtiradi, 2 xil qiymat qabul qilishi mumkin:

outside - markerni ro'yxat bloki tashqarisiga joylashtirish;

inside - markerni ro'yxat bloki ichiga joylashtirish.

inside

1. Birinchi qator
2. Ikkinchi qator
3. Uchunchi qator

outside

1. Birinchi qator
2. Ikkinchi qator
3. Uchunchi qator

3.3.10-rasm

Bu stil eskirgan bo'lib, yangi brauzerlarda ishlamasligi mumkin.

List-style-image - ro'yxatning eng qiziq xususiyati bo'lib, markerga o'zingiz hohlagan rasmni qo'yishingiz mumkin bo'ladi. "**url**" kalit so'zidan keyin, rasm adresini yozasiz.



3.3.11-misol. List-style-image dan foydalanish misoli

```
<html>
<head>
  <title>List</title>
  <style>
    ol {
      list-style-image: url(rasm.png);
    }
  </style>
</head>
<body>
  <ol>
    <li>Alex Fergusion</li>
    <li>David Moyes</li>
    <li>Lui Van Gal</li>
  </ol>
</body>
</html>
```



3.3.11-rasm

List-style - yuqoridagi ro'yxatga oid barcha xususiyatlarni birlashtiruvchi xususiyat.

```
ul {
list-style-type:square;
list-style-position: inside;
list-style-image: url(rasm.png);
}
```

Shu 5 qatorli kodni, "**list-style**" orqali quyidagicha qisqartirish mumkin.

```
ul {list-style:square inside url(rasm.png) }
```

BLOK tushunchasi

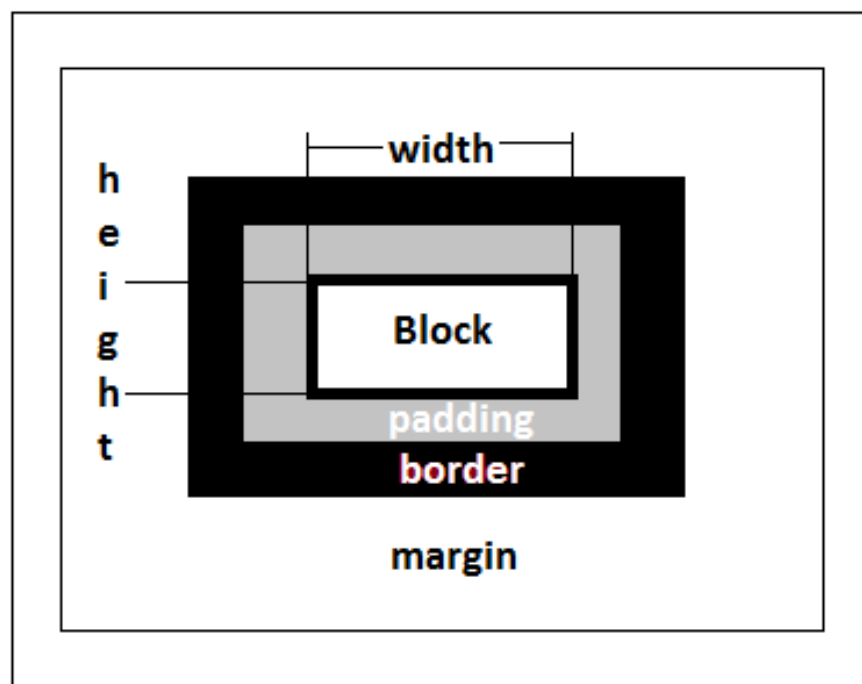
Web sohasida blok tushunchasi juda kerakli atama hisoblanadi. Saytning barcha brauzerlarda bir xilda ochilishi, rasm va yozuvlarni o'ylaganingizdek namoyon bo'lishi aynan blok holatiga bog'liq. Blokga tegishli stillar xususiyatini ko'rib chiqamiz. Dastlab, umumiy blok nimaligini bilib olamiz.

Blok bu - ma'lum bir chegaraga ega bo'lgan va alohida qism sifatida qaraladigan ma'lum bir axborotni o'zida saqlaydigan joydir.

HTML gipermatnli belgilash tilini oladigan bo'lsak, u yerda ham blok va qator elementlari tushunchasi bor. Blokli elementlar, alohida strukturaga ega bo'lib, to'rtburchak chegaraga bilan o'ralgan bo'ladi. Ikkita blokli elementni bir qatorga joylashtirib bo'lmaydi. HTML da blokli elementga misol qilib, "**H1...H6**", "**P**" va "**DIV**" teglarini olish mumkin. Har safar bu teglarni ishlatganda brauzer har biri uchun alohida qator ajratib beradi.

Qatorli elementlarga esa "**strong**", "**em**", "**i**" kabi teglarni misol qilish mumkin. Bu teglardan bir qatorda bemalol foydalansa bo'ladi.

CSS stillar texnologiyasida ham blok tushunchasi mavjud va u ham o'z strukturasi va to'rtburchak chegarasiga ega. Strukturali blok ichida qatorli blok ham bo'lishi mumkin (blok ichidagi yozuvlar), qatorli blok yozuv ko'rinishida ham bo'lishi mumkin. Strukturali blok ichida ob'ektki blok bo'lishi mumkin. Yuqoridagi nazariy ma'lumotlarni quyida bitta rasm orqali tushunish mumkin.



3.3.12-rasm

Bloklarning o'z xususiyatlari bor. Misol uchun, blok uzunligi (width), blok balandligi (height), chegarasi(border) va hokazo.

Har bir blok qandaydir tashkil etuvchilardan iborat bo'lishi shart. Ya'ni yozuv, rasm yoki boshqa axborot. Blokning bu qismi **kontent** deb yuritiladi. Misol uchun "**P**" tegi uchun kontent yozuv hisoblanadi.

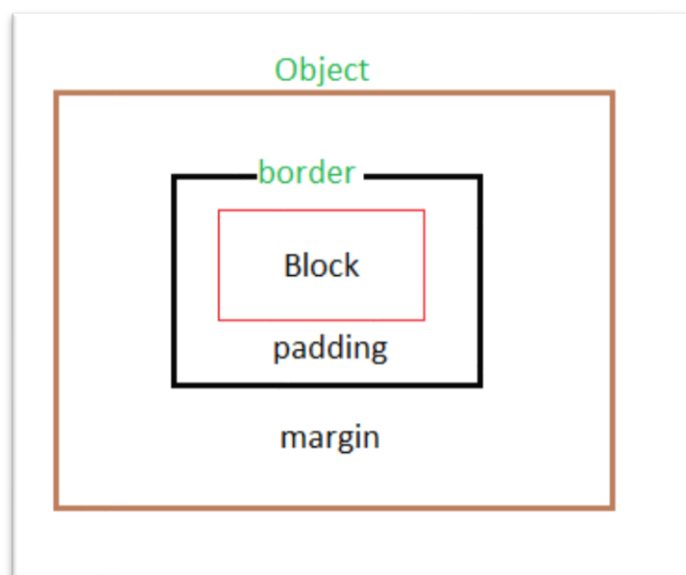
Blok atrofida qandaydir boshqa blok yoki ob'ektlar bo'lishi mumkin, blokdan ulargacha bo'lgan masofa "**margin**" kalit so'zi bilan ifodalanadi.

Blok ajralib turishi uchun, uning chegarasi bo'lishi lozim, bu chegaragacha masofa "**border**" so'zi bilan ifodalanadi.

Yuqoridagilardan tashqari blokning ichki bo'sh oralig'i bo'lishi mumkin, ya'ni blokdan blok chegarasigacha bo'lgan masofa, bu masofa "**padding**" kalit so'zi bilan ifodalanadi.

Blokning oraliq masofalari: margin, padding.

Blokning o'z ramkasi(chegarasi)gacha bo'lgan oraliq masofa va ramkasidan atrofidagi biror ob'ektgacha bo'lgan masofa kabi xususiyati bor. Bu ikki oraliq masofani tushinish uchun quyidagi rasmni keltirib o'tamiz. Bu yerda blok sifatida biror blokli teg (p, div, ...) bo'lishi mumkin.



3.3.13-rasm

Rasmdan ko'rinib turibdiki, blokdan o'z ramkasigacha bo'lgan masofa "**padding**" deb, blok ramkasidan biror ob'ekt(element) yoki element mavjud bo'lmasa hujjat oxirigacha bo'lgan masofa "**margin**" deb yuritilar ekan. Bu ikki xususiyatni kod ichida ishlatish, blokni web sahifada aniq va tushunarli bo'lishini ta'minlaydi. Misol ko'ramiz va hammasi tushunarli bo'ladi.



3.3.12-misol. Bloklarda margin va paddingdan foydalanish misoli

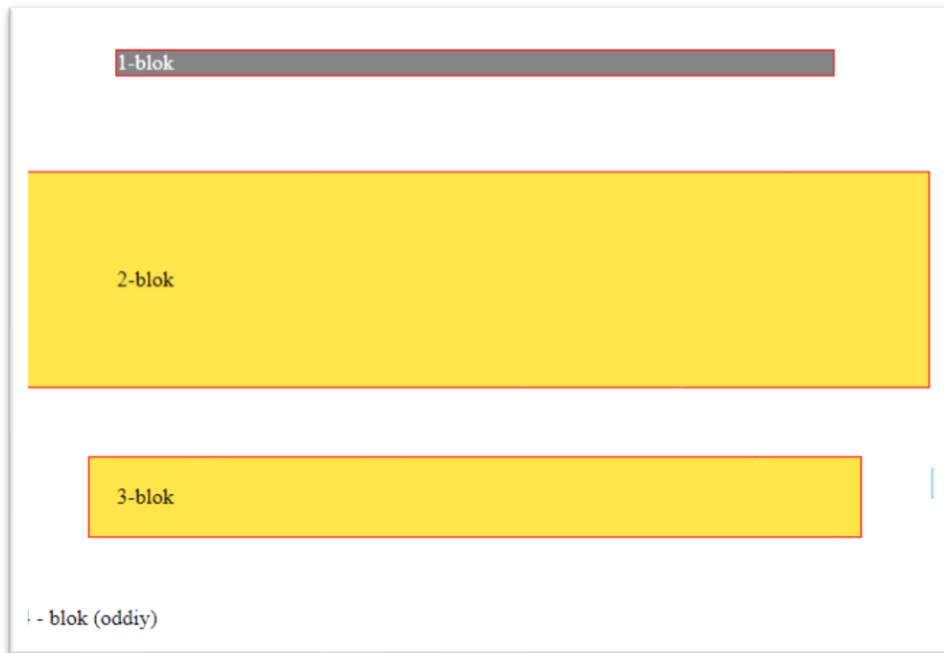
```
<html>
<head>
  <title>Margin Padding</title>
  <style>
.p1 {
background-color : gray;
border:1px solid red;
margin:70px;
color:white;
}
.p2 {
background-color : #FFE446;
border:1px solid red;
padding:70px;
}
.p3 {
background-color : #FFE446;
border:1px solid red;
margin:50px;
padding:20px;
}
```

```

</style>
</head>
<body>
<p classq"p1">1-blok</p>
<p classq"p2">2-blok</p>
<p classq"p3">3-blok </p>
<p>4 - blok (oddiy)</p>

</body>
</html>

```



3.3.14-rasm

Oldingi mavzularda ishlatilgan quyidagi kalit so'zlar ham bu xususiyat uchun ishlaydi:

- left* - chap tomondan joy qoldirish (*margin-left, padding-left*)
- right* - o'ng tomondan joy qoldirish (*margin-right, padding-right*)
- top* - yuqori qismdan joy qoldirish (*margin-top, padding-top*)
- bottom* - pastki qismdan joy qoldirish (*margin-bottom, padding-bottom*).

Endi, kalit so'zlar orqali bo'sh joy qoldirishga oid misolni ko'ramiz.



3.3.13-misol. Bloklarni joylashtirish

```

<html>
<head>
<title>Margin Padding</title>
<style>
.d1 {
background-color : gray;

```

```

border:1px solid red;
margin-top:70px;
margin-bottom:10px;
color:white;
}
.d2 {
background-color:#FFE446;
border:1px solid red;
padding-left:70px;
padding-right:10px;
}
</style>
</head>
<body>
<p classq"d1">Namangan</p>
<p classq"d2">Toshkent</p>
</body>
</html>

```



3.3.15-rasm

Yuqoridagi CSS kodni qisqartirib quyidagicha yozish mumkin:

```

.d1{margin: 70px 0 10px 0;}
.d1{padding: 0 10px 0 70px;}

```

Ko'rinib turibdiki, xususiyatlarni berilishi ketma-ketligi soat strelkasi yo'nalishi bo'yicha, ya'ni yuqori qism (**top**), o'ng tomon (**right**), pastki qism (**bottom**), chap tomon (**left**).

Qaysi paytda qaysi birini (margin, padding) ishlatishni, o'zingiz hal qilishingiz kerak. Faqat shuni aytishim mumkinki, "**padding**" blok ichida bo'ladi, "**margin**" blok tashqarisida.

Blok o'lchamlari: width, height. Blokga oid xususiyatlarni ko'rib chiqishni davom ettiramiz. Bu safar yana bir asosiy xususiyatlardan biri – blokning balandligi (**height**) va uning uzunligi (**width**) haqida gaplashamiz. Agar sezgan bo'lsangiz, blokning uzunligi va balandligi avtomat aniqlashtiriladi, yozuv qancha

katta bo'lsa (yoki blokni boshqa tashkil etuvchi), shuncha blokning uzunlik va balandlik katta bo'ladi.

Shunday bo'lsa ham, biz CSS orqali blokning uzunlik va balandligini o'zimiz o'rnatishimiz mumkin. Demak bu ish uchun biz quyidagi ikkita xususiyatdan foydalanamiz:

- *height* - blokning balandligi;
- *width* - blokning uzunligi.

Odatda, CSS da blok sifatida "**div**" tegi ishlatiladi, lekin paragraf(p tegi), ro'yxatlarni ham blok sifatida ishlatish mumkin.

Nazariy ma'lumotlar bilan tanishib chiqdik, endi misol ko'ramiz.



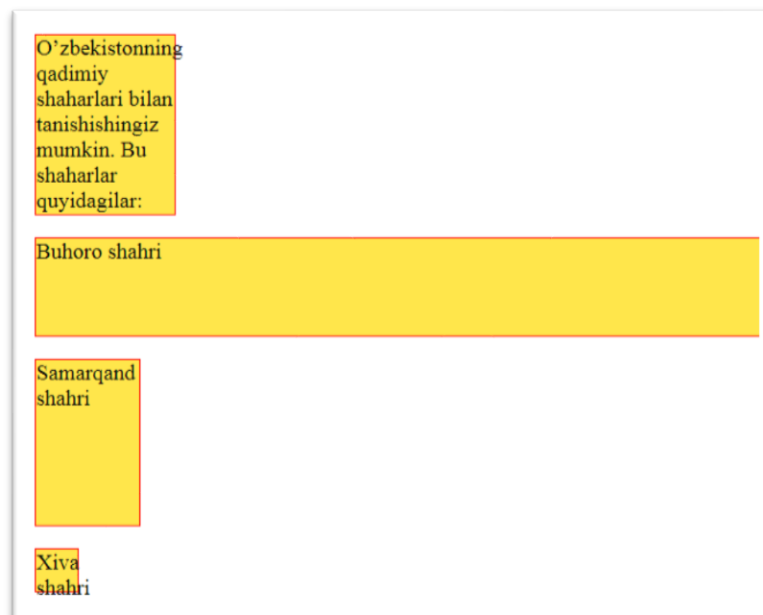
3.3.14-misol. Blok o'lchamlari

```
<html>
<head>
  <title> Blok o'lchamlari </title>
  <style>
.mu1 {
width: 100px;
border: 1px solid red;
background: #FFE446;
}
.mu2 {
height: 70px;
border: 1px solid red;
background: #FFE446;
}
.mu3 {
width: 75px;
height: 120px;
border: 1px solid red;
background: #FFE446;
}
.mu4 {
width: 30px;
height: 30px;
border: 1px solid red;
background: #FFE446;
}
  </style>
</head>
<body>
<p class="mu1">O'zbekistonning qadimiy shaharlari bilan tanishishingiz
mumkin. Bu shaharlar quyidagilar:</p>
```

```

<p classq"mu2">Buhoro shahri</p>
<p classq"mu3">Samarqand shahri</p>
<p classq"mu4">Xiva shahri</p>
</body>
</html>

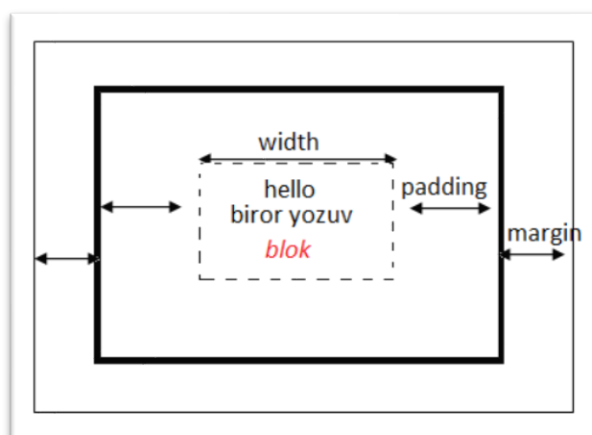
```



3.3.16-rasm

Har xil o'lchamdagi bloklar hosil qildik, agar biz yozgan yozuv blok ichiga sig'masa, ko'rib turganingizdek blok tashqarisida ham yozuvning davomi yozilaverar ekan. Shuning uchun bloklar hosil qilishda hamma narsalarni hisobga oling.

Blok uzunligi va balandligini berish bilan siz faqat yozuv uchun ajratilgan joyni aniqlashtirgan bo'lasiz. Oraliq masofalarni belgilovchi CSS xususiyatlar (**margin**, **padding**) bu o'lchamlarga kirmaydi. Siz kiritgan o'lchamdan keyin o'rnatgan oraliq masofa qiymatichalik joy tashlab ketiladi, rasmda ko'radigan bo'lsak:



3.3.17-rasm

Yuqoridagi o'lcham beruvchi CSS xususiyatlardan tashqari yana quyidagi o'lchov beruvchi xususiyatlar mavjud:

- *max-width* - blokning maksimal uzunligi;
- *min-width* - blokning minimal uzunligi;
- *max-height* - blokning maksimal balandligi;
- *min-height* - blokning minimal balandligi.

Bu xususiyatlar blokning maksimal va minimal o'lchovlarini aniqlashtirib beradi, biror kod orqali bu maksimal va minimal o'lchovlardan chetga chiqib bo'lmaydi. Beriladigan o'lchovlar faqat shu oraliqga tushishi lozim. Misol,

```
.size {  
height:80px;  
width:600px;  
max-width: 300px;  
border:1px solid red;  
background-color:orange;  
}
```

Bu holda blok uzunligi 300 piksel bo'ladi, biz o'rnatgan 600 piksel ishlamaydi. Menimcha hammasi tushunarli, tushunarsiz narsalar pochta yoki kommentariyada ko'rib chiqiladi.

Bloklarni sahifa bo'ylab joylashtirish: POSITION. CSS bizga istalagan ob'ektni brauzerning istalgan joyida chiqarish imkonini beradi. Buning uchun shu blok koordinatalrini ko'rsatishimiz kerak holos. Blok deganda "**div**" tegidan tashqari ro'yxatlar, paragraflar, sarlavhalar, rasmlar, umumiy qilib aytganda, blokli model ko'rinishidagi barcha teglar tushuniladi. Koordinatalarni o'rnatib, bloklarni joylashtirish **pozisiyalash** (joylashinish) deyiladi. Joylashtirish 2 xil ko'rinishda bo'ladi va bu xil ko'rinishlar uchun quyidagi CSS xususiyatlarini ishlatamiz.

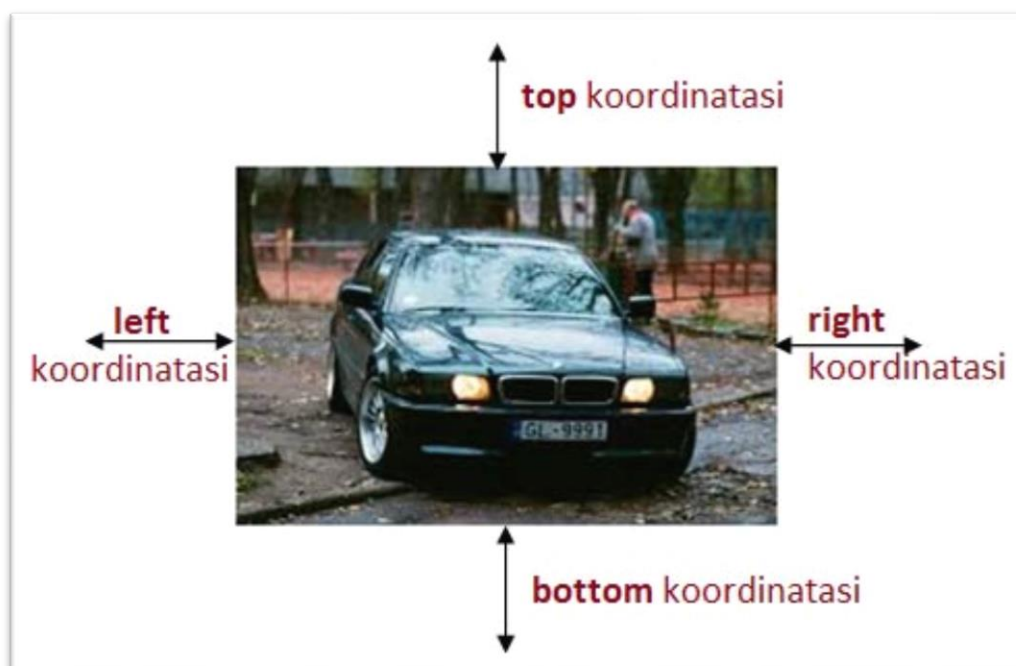
1. Absolyut joylashtirish (*position:absolute*;) .

2. Biror narsaga nisbatan joylashtirish (*position:relative*;) .

Absolyut joylashtirish. Odatda, joylashtirishda ushbu tur joylashtirish ishlatiladi. CSSda joylashtiruv vazifasini "**Position**" kalit so'zi amalga oshirib beradi, ushbu joylashtiruvni absolyut qilish uchun esa "**absolute**" qiymati beriladi. Undan so'ng quyidagi kalit so'zlar orqali koordinatalar beriladi:

- *top*;
- *left*;
- *right*;
- *bottom*.

Bu kalit so'zlar, brauzerda qanday namoyon bo'lishini quyidagi rasmda ko'rish mumkin.



3.3.18-rasm

Absolyut joylashtiruvning yaxshi tomoni shundaki, blok aynan siz bergan koordinatalarda joylashadi, o'zi oldindan aniqlashtirib qo'yilgan joy (html kodda berilgan joy) esa avtomat o'chadi. Uning o'rnida boshqa narsalar joylashishi mumkin. Misol ko'ramiz, chunki so'z bilan tushuntirish biroz qiyinroq. Dastlab, oddiy rasm qo'yish misolini ko'ramiz.



3.3.15-misol. Bloklarda position dan foyadalanish

```
<html>
<head>
<title>Position</title>
</head>
<body>
<img srcq"rasm.jpg"/>
<h1>HTML ni Notepad yoki TextEdit da yozing HTML ni
quyidagi professional muharrirlar ... . </h1>
</body>
</html>
```



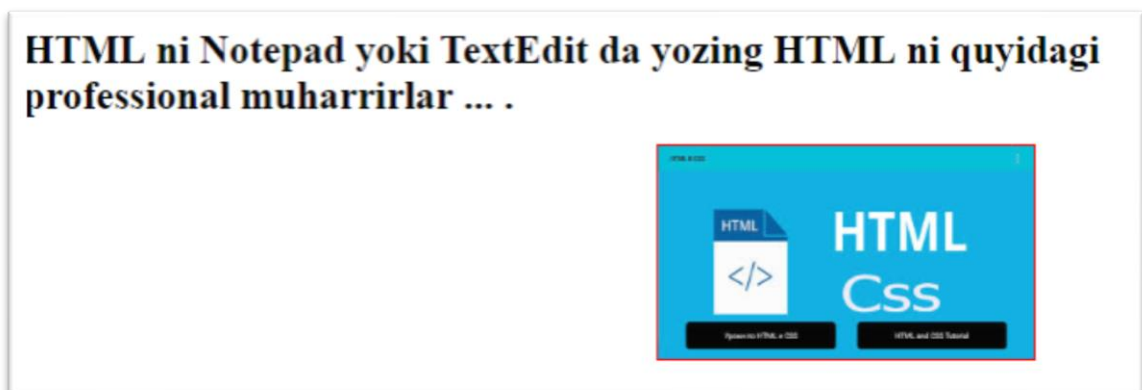
3.3.19-rasm

Endi shu misoldagi rasmni koordinata bo'yicha joylashtiramiz.



3.3.16-misol. Bloklarda position dan foyadalanish

```
<html>
<head>
<title>Position</title>
<style>
.smile{
border: 1px solid red;
position:absolute;
top:100px;
right:100px;
}
</style>
</head>
<body>
<img classq"smile" srcq"rasm.jpg"/>
<h1>HTML ni Notepad yoki TextEdit da yozing HTML ni quyidagi professional
muharrirlar ... . </h1>
</body>
</html>
```



3.3.20-rasm

E`tibor bersak, rasm o`rnatilgan koordinataga ko`chdi va rasm o`rnida biz yozgan yozuv paydo bo`ldi. Rasm koordinatasi tepadan 100 piksel va o`ng tomondan 100 piksel koordinatada joylashdi.

Odatda koordinata faqat ikkita qiymatda beriladi:

```
.mu{  
position:absolute;  
bottom:300px;  
left:100px;  
}
```

yoki

```
.mu{  
position:absolute;  
top:100px;  
right:100px;  
}
```

Blok brauzerning biror burchagiga nisbatan olinadi, shu sababli ikkita koordinata qiymati beriladi.

Agar, blok brauzerga yopishib turishini hohlasangiz, ya`ni brauzer kattaligi o`zgarganda aylantirgich (prokrutka) chiqsa ham, blok aylanmasdan qotib turish kerak bo`lsa, "**absolute**" qiymat o`rniga "**fixed**" qiymat berish lozim.

```
.mu{  
position:fixed;  
top:100px;  
right:100px;  
}
```

Bu ikki qiymatni misollar orqali, ishlatib ko`ring.

Biror narsaga nisbatan joylashtirish. Bu tur joylashtirishda, blokning dastlabki holatidan boshlab, koordinata hisoblanadi (absolyut joylashtirish kabi brauzerning yuqori qismidan emas). Undan tashqari blokning dastlabki holati o`zgarishsiz saqlab qolinadi, odatda bo`sh joy bo`lib ko`rinib turadi. Bu joylashtirishni amalga oshirish uchun, "**position**" xususiyatiga "**relative**" qiymati beriladi. Yuqoridagi misolni "**relative**" orqali yozamiz.



3.3.17-misol. Nisbiy joylashtirish

```
<html>  
<head>  
<title>Position</title>  
<style>  
.smile{  
border: 1px solid red;  
position:relative;  
top:100px;  
right:10px;  
}
```

```

</style>
</head>
<body>
<img classq"smile" srcq"rasm.jpg"/>
<h1>HTML ni Notepad yoki TextEdit da yozing HTML ni
quyidagi professional muharrirlar ... . </h1>

</body>
</html>

```



3.3.21-rasm

Misoldan ko'rinib turibdiki, rasm tegi birinchi yozilgan, demak rasm brauzerning chap yuqori qismida joylashishi kerak, uning tagida esa yozuv. Biz koordinatani berganimizdan so'ng, rasm boshlang'ich holatidan berilgan qiymat bo'yicha surildi va dastlabki holati bo'sh bo'lib qoldi. Yozuv ham uning joyiga o'tmadi, absolyut joylashtirish bo'lganida, yozuv rasmning oldingi holatiga o'tar edi.



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. Shrifltar stili qanday beriladi?
2. Stillar yordamida matnlarni formatlash qanday amalga oshiriladi?
3. Stillar jadvali yordamida ro'yxatlar qanday formatlanadi?
4. Blok nima?
5. Stillar uchun blokni ahamiyati qanday?
6. Blok eni va bo'yi qanay beriladi?
7. Blokning oraliq masofalari qanday ifodalanadi?
8. Position nima?
9. Absolyut joylashtirish nima?
10. Nisbiy joylashtirish nima?

3.4-§. Blok va kontenlarni joylashtirish: clip, visibility, overflow, vertical-align. Sloylar bilan ishlash.



O`quv modullari

Bloklarni joylashtirish, clip, visibility, overflow, sloylar bilan ishlash

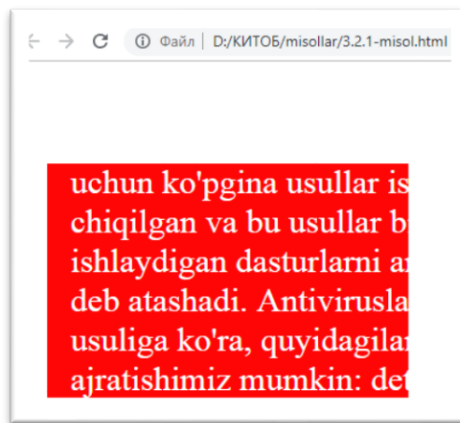
Blok va kontenlarni joylashtirish: clip, visibility, overflow, vertical-align. Bloklar joylashuvi mavzusidagi navbatdagi CSS xususiyatimi "**clip**" deb nomlanadi va bu xususiyat blok ichidagi kontentni qancha qismi namoyon bo'lishini aniqlashtirib beradi. Bu xususiyat ichiga koordinata nuqtalari beriladi va shu nuqtalar orasida yotgan kontent qismi ko'rinadi holos.

Kontent faqatgina to'rtburchak ko'rinishda kesib olinishi mumkin va beriladigan koordinata o'qlari quyidagi ketma-ketlikda beriladi: **Y1, X1, Y2, X2**. Bu koordinata o'qlari oldidan "**rect**" degan kalit so'z ishlatiladi. Tushunarli bo'lishi uchun misol ko'ramiz.



3.4.1-misol. Blok va kontenlarni joylashtirish

```
<html>
<head>
<title>Clip</title>
<style>
clip{
position: absolute;
clip: rect(40px, 180px, 150px, 10px);
width: 250px;
color: white;
background: red;
border: 1px solid blue;
padding: 20px;
}
</style>
</head>
<body>
<div classq"clip"> Hozirgi vaqtda viruslarni yo'qotish uchun ko'pgina usullar
ishlab chiqilgan va bu usullar bilan ishlaydigan dasturlarni antiviruslar deb
atashadi. Antiviruslarni, qo'llanish usuliga ko'ra, quyidagilarga ajratishimiz
mumkin: detektorlar, faglar, vaktsinalar, privivkalar, revizorlar.</div>
</body>
</html>
```



3.4.1-rasm

Natija tushunarli bo'lishi uchun, koordinatalarni ko'rsatib o'tamiz.

Navbatdagi CSS xususiyati bloklarni ko'rinarli va ko'rinmas holga keltiradi, ya'ni "**visibility**" xususiyati. Bu amalni "**display**" orqali ham amalga oshirish mumkin, lekin bu ikkisini farqi bor. "**Visibility**" orqali blokni ko'rinmas holga olib kelsa, blok uchun ajratilgan joyi hech kim ishlata olmaydi, ya'ni blok joyi saqlanib qoladi, "**display**" xususiyatida esa blok joyi saqlanmaydi. Bu xususiyatni quyidagi asosiy qiymatlari mavjud:

visible - blokni ko'rinarli holga olib keladi;

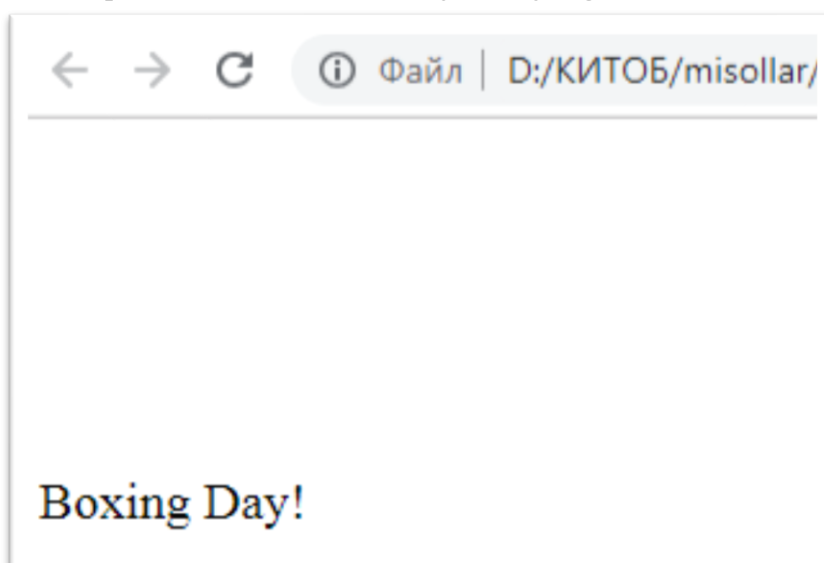
hidden - blokni ko'rinmas holga olib keladi;



3.4.2-misol. Visible hususiyati

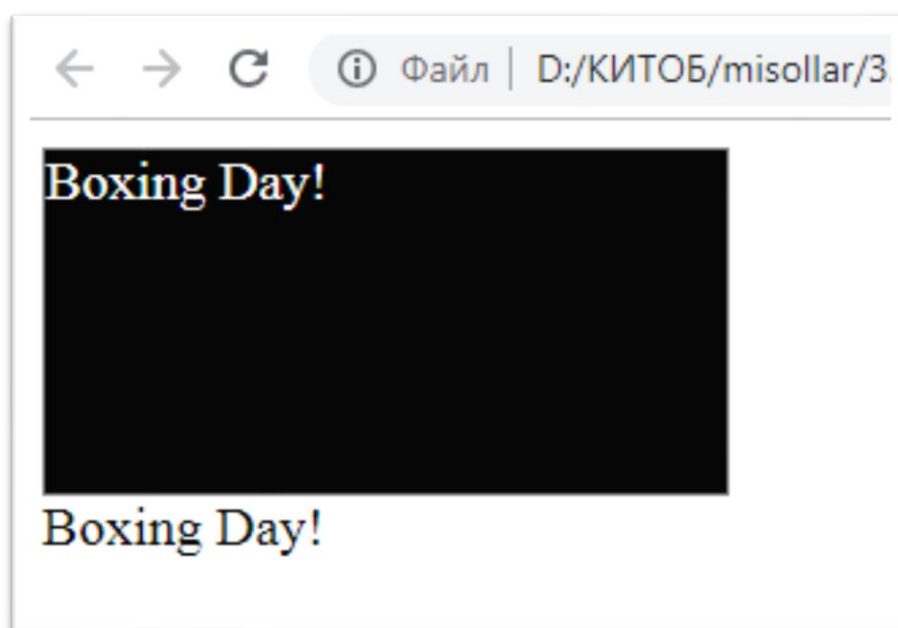
```
<html>
<head>
<title>Visibility</title>
<style>
.vis{
height:150px;
width:200px;
background:black;
color:white;
border:1px solid gray;
visibility:hidden;
}
</style>
</head>
<body>
<div classq"vis">Boxing Day!</div>
<div>Boxing Day!</div>
</body>
</html>
```

Natijada, "**vis**" klasi bilan yozilgan yozuv ko'rinmaydi, lekin **150x200** o'lchamdagi joy bo'sh qoldiriladi va undan keyin keyingi elementlar chiqariladi.



3.4.2-rasm

Agar, "**visibility:visible**" qilib blokni ko'rintirsak, quyidagi ko'rinishda bo'ladi.



3.4.3-rasm

Bloklarni hosil qildik, uni ichini to'latdik ham deylik, agar blok ichidagi kontent blok ichiga sig'masa nima bo'ladi? Bu holni ham CSS o'ylab qo'ygan. Aynan kontent blok ichiga sig'masa nima qilish kerakligini "**overflow**" xususiyati orqali berib qo'yishimiz mumkin, uning quyidashi qiymatlari mavjud:

visible - kontent blok ichiga sig'masa ham to'liq ko'rsatish;

hidden - kontentning faqat blok ichiga sig'gan qismi ko'rinadi holos;

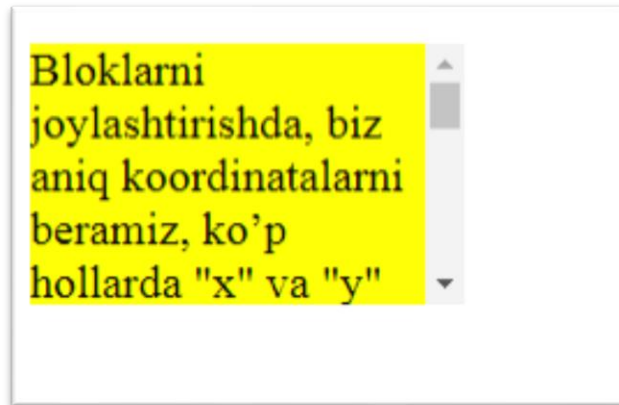
scroll - har doim o'tkazgich(prokrutka) ko'rsatish;

auto - o'tkazgich faqat kerak bo'lganda paydo bo'ladi.



3.4.3-misol. *Overflow* hususiyati

```
<html>
<head>
<title>Over</title>
<style>
.over{
    height:90px;
    width:150px;
    background:yellow;
    overflow:auto;
}
</style>
</head>
<body>
<div classq"over">
    Bloklarni joylashtirishda, biz aniq koordinatalarni beramiz, ko'p hollarda
    "x" va "y" nuqtalarni, yoki boshlang'ich holatdan surilishlarni (position).
</div>
</body>
</html>
```



3.4.4-rasm

Qiymatdan ko'rinib turibdiki **auto** bo'lsa o'tkazgich paydo bo'ladi. Bu ko'pincha kerak bo'ladi.

Keyingi CSS xususiyat, vertikal bo'ylab joylashtirish deb nomlanadi, ya'ni "**vertical-align**". Elementni vertikal bo'ylab tekislash yoki shu elementni istalgancha yuqori va pastga o'rnatish mumkin. Quyidagi qiymatlarni qabul qiladi:
baseline - elementni vertikal bo'ylab asosiy chiziqga o'rnatish(odatiy hol);
bottom - eng quyida turgan element bilan bir chiziqga joylashtirish;
middle - elementni vertikal bo'ylab asosiy chiziq markazidan boshlab o'rnatish;
sub - quyi indeks ko'rinishida joylashtirish;
super - yuqori yendeks ko'rinishida joylashtirish;

text-bottom - elementning eng quyi chiqig'i bo'ylab joylashtirish;
text-top - elementning eng yuqori chizig'i bo'ylab joylashtirish;
top - eng yuqori turgan element bilan bir chiziqqa joylashtirish;
 Bu qiymatlardan tashqari foizlar, piksellar ko'rinishida ham qiymat berish mumkin, mening fikrimcha piksellar orqali berganingiz qulayroq.



3.4.4-misol. *vertical-align* xususiyati

```
<html>
<head>
<title> vertical-align xususiyati </title>
</head>
<body>
<div styleq"font-size:30px;">
<span styleq"vertical-align:sub">W</span>eb
<span styleq"vertical-align:middle;">D</span>a
<span style q "vertical-align:bottom;">S</span>
<span styleq"vertical-align:10px">T</span>
<span styleq"vertical-align:8px">U</span>
<span styleq"vertical-align:6px">R</span>
<span styleq"vertical-align:4px">L</span>
<span styleq"vertical-align:2">A</span>
<span styleq"vertical-align:middle">Sh</span>
</div>
</body>
</html>
```



3.4.5-rasm

Sloylar bilan ishlash

Sloy bu - qobiq, ya'ni alohida alohida blokdir. Agar 3 ta blok bo'lsa va ularning o'lchamlari bir-xil bo'lsa va siz ularni ustma-ust joylasangiz, eng quyi sloyl(blok) ko'rinaydi. Bunda, ustma-ust joylashuv bo'lib qoladi, lekin sloyl mavjud bo'lib turaveradi.

Bloklarni joylashtirishda, biz aniq koordinatalarni beramiz, ko'p hollarda "x" va "y" nuqtalarni, yoki boshlang'ich holatdan surilishlarni (**position**). Mavzuda yana bir nuqta - "z" ni kiritib, har bir blokni ustma-ust joylashuvini aniqlashtiramiz. Sloylarni boshqarishni CSSning "**z-index**" xususiyati orqali amalga oshiramiz.

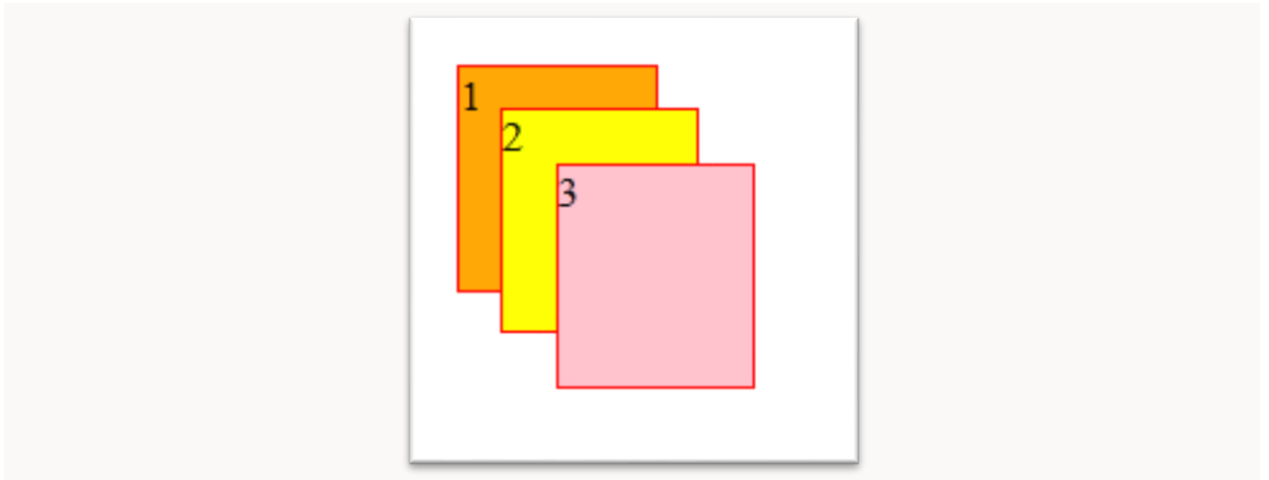
Blokning koordinatalari bilan birga ularning joylashuvini raqamlaymiz. Shu nomer orqali qaysi sloy eng yuqorida va qaysinisi eng quyida bo'lishini ko'rsatamiz. Misol ko'ramiz, dastlab bloklarni raqamlamaymiz va natijani ko'ramiz.



3.4.5-misol. Sloylar bilan ishlash

```
<html>
<head>
<title>Sloy</title>
<style>
.sloy1 {
position:absolute;
left:15px;
top:15px;
height:80px;
width:70px;
border:1px solid red;
background-color:orange;
}
.sloy2 {
position:absolute;
left:30px;
top:30px;
height:80px;
width:70px;
border:1px solid red;
background-color:yellow;
}
.sloy3 {
position:absolute;
left:50px;
top:50px;
height:80px;
width:70px;
border:1px solid red;
background-color:pink;
}
</style>
</head>
<body>
<div classq"sloy1">1</div>
<div classq"sloy2">2</div>
```

```
<div classq"sloy3">3</div>
</body>
</html>
```



3.4.6-rasm

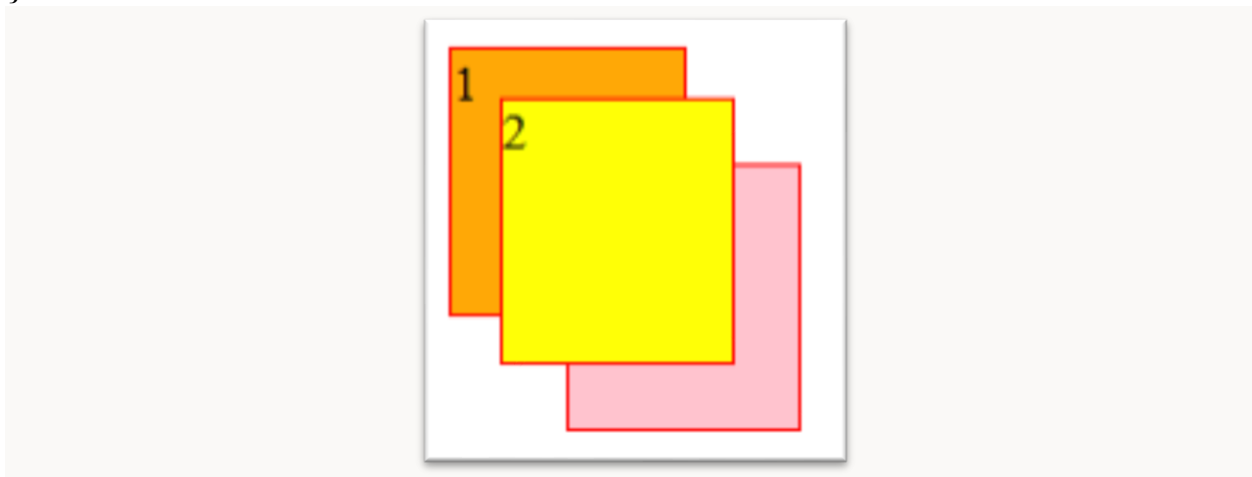
Kod natijasidan ko'rinib turibdiki, qaysi sloy oxirida yozilsa, o'sha blok eng yuqorida namoyon bo'ladi. Endi, yuqoridagi kodning CSS qismini o'zgartiramiz, ya'ni qo'shamiz va quyidagiga ega bo'lamiz.



3.4.6-misol. Z-index xususiyati

```
.sloy1 {
position:absolute;
left:15px;
top:15px;
height:80px;
width:70px;
border:1px solid red;
background-color:orange;
z-index:2;
}
.sloy2 {
position:absolute;
left:30px;
top:30px;
height:80px;
width:70px;
border:1px solid red;
background-color:yellow;
z-index:3;
}
.sloy3 {
position:absolute;
```

```
left:50px;  
top:50px;  
height:80px;  
width:70px;  
border:1px solid red;  
background-color:pink;  
z-index:1;  
}
```



3.4.7-rasm

Qaysi blokning raqami (z-index:3) katta bo'lsa, o'sha blok eng yuqorida ko'rinadi, misolda ikkinchi blokning raqami eng katta.

Yaxshilab o'ylab ko'rilsa, bu xususiyat bilan ancha chiroyli ishlarni amalga oshirish mumkin bo'ladi, chiroyli sahifalar, soya shaklidagi rasmlar, tushuvchi menyular va hakazo.



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. Bloklar sahifada qanday joylashtiriladi?
2. Clip qanday hususiyat?
3. Overflow qanday hususiyat?
4. Sloy nima?
5. Sloylarni joylashtirish tartibi qanday beriladi?

4-BOB. JAVASCRIPTDA DASTURLASH.

4.1-§. JavaScriptning asosiy tushunchalari: xossalar, metodlar, xodisalar.



O`quv modullari

xossalar, metodlar, xodisalar, interaktiv, kod, script, izox, kontent, brauzer, hujjat, parameter, uslub.

JavaScript — bu Web-sahifalarning foydalanuvchi harakati, ta'siri bilan interaktiv tahrirlash imkonini beruvchi tildir.

Ko'rsatmalar izchil interpretatorlar bilan oddiy web-brauzerda bajariladi. Boshqa so'z bilan aytganda, dastur kodi HTML hujjatga tadbiq qilinadi va kliyent tomonidan bajariladi. Dastur bajarilishi uchun endi web sahfa qayta yuklanishi kerak bo'lmaydi. Qandaydir hodisalarning paydo bo'lishi natijasida dastur bajariladi. Masalan, forma ma'lumotlarini jo'natish oldidan ularning ruxsat etilgan qiymatlari tekshirilishi mumkin, agar qiymatlar kutilganiga muvofiq bo'lmasa ma'lumot yuborish bekor qilinadi.

JavaScriptda birinchi dastur. Dasturlarni o'rganishda odatda "Hello, World" ("Salom, Dunyo") matnini chop etish bilan boshlanadi. Biz ham ananani buzmaymiz.



4.1.1-misol. Birinchi dastur.

```
<!DOCTYPE HTML>
<html> <head>
<title>Birinchi dastur</title>
<meta http-equivq"Content-Type" contentq"text/html; charsetqutf-8">
</head>
<body>
<script typeq"text/javascript">
<!--
document.write("Salom, Dunyo");
//-- >
</script> <noscript>
<p> Sizning brauzeringiz JavaScriptni qo'llamaydi </p> </noscript> </body>
</html>
```

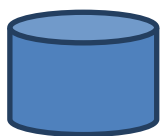


4.1.1-rasm

Oddiy **Blokn**ot da kodlarni yozamiz va HTML formatda test.html nomi bilan Ekranda quyidagi hollardan biri bo'lishi mumkin.

- Web-brauzer oynasida "Salom, Dunyo" yozuvi aks etadi — demak, hammasi joyida;
- Web-brauzer oynasida "Sizning brauzeringiz JavaScriptni qo'llamaydi" va Web-brauzerning «**Zapustit skripto`?**» so'rovini beradi — demak, Web-brauzerni sozlash va **Podtverjdat zapusk skriptov** bandi qarshisiga bayroqcha o'rnatiladi. **Razreshit zapusk spriptov** bandi qarshisida bayroq o'rnatilsa har safar , "Da" javobini berish kerak bo'ladi;
- Web-brauzer oynasida "Sizning brauzeringiz JavaScriptni qo'llamaydi" yozuvi aks etsa va Web-brauzer hech qanday savol bermasa— demak, Web-brauzera sozlanishida “**Zapretit zapusk skriptov**” bandi qarshisiga bayroq o'rnatilgan. **Razreshit zapusk spriptov** bandiga bayroq o'rnatish kerak;
- Web-brauzer oynasida hech qanday yozuv bo'lmasa — demak, dastur kodini yozishda xatoga yo'l qo'yilgan. JavaScriptda o'zgaruvhclarni nomlashda kichik va katta harflar farqlanadi. Bundan tashqari, har bir harf, har bir qo'shtirnoq alohida ahamiyat kasb etadi. Bir harfda ham yanglishish butun dasturning ishlamasligiga yoki xato ishlashiga sabab bo'ladi.

Demak, biz birinchi muammoga duch keldik - ixtiyoriy foydalanuvchi web-brauzerni sozlash bo'limidan skriptni ishlatishni o'chirib qo'yishi mumkin. Biroq bu yagona muammo emas. Bunga sabab har bir web-brauzerda maxsus kod yozish kerak bo'ladi.



Bu kitobdagi barcha script misollar Microsoft Internet Explorer uchun yozilgan va ular boshqa brauzerlarda ishlamasligi mumkin. Buni yodda saqlash kerak.

Dastur HTML hujjatga `<script>` tegi yordamida tadbiiq qilinadi. *Type* parametrda MIME-tip `text/javascript` qiymati ko'rsatiladi. Bundan tashqari qaysi tildan foalanilayotgani **language** parametrda qaysi tildan foydalanilayotgani ko'rsatish kerak bo'ladi (bizni misolimizda — JavaScript). Mazkur parameter HTMLning oldingi versiyalarida ishlatilar edi, hozirda esa type parametri bilan bir paytda muvofiqlik uchun foydalanialadi:

```
<script typeq"text/javascript" languageq"JavaScript">
```

Agar Web-brauzer JavaScript qo'llamasa yoki brauzer sozlanishida ta'qiqlangan bo'lsa, ko'rinadigan ma'lumot <noscript> va </noscript> teglari orasidagi yozuv chop etiladi.

```
<!--
```

```
document.write("Salom, Dunyo");
```

```
//-- >
```

Hozirgi vaqta barcha web brauzerlar <script> tegini taniydi. Shuning uchun izoh belgisiga hojat yuq.

`document.write("Salom, Dunyo");` - bu qator ifoda deb ataladi va uning bajarilishidan "Salom, Dunyo" matni web brauzer oynasida paydo bo'ladi. JavaScriptda har bir ifoda nuqta vergul bilan ajratiladi.

Izoxlar. JavaScript «`//`» belgisidan keyinga yozuvni bir qatorli izoh hisoblaydi: *// bir qatorli izox*

Bir qatorli izox ifodadan keyin yozilishi mumkin:

```
document.write("Hello, world"); // Bir qatorli izox
```

Bundan tashqari ko'p qatorli izohlar mavjud. U «`/*`» belgisi bilan boshlanib «`*/`» belgisi bilan tugaydi:

```
/*
```

```
Ko'p qatorli izox
```

```
*/
```

alert() uslubi. Bu ma'lumotlar oynasi va OK tugmasini ko'rsatish uchun xizmat qiladi.



4.1.2-misol. alert() uslubi.

```
<!DOCTYPE HTML>
```

```
<html> <head>
```

```
<title> alert() uslubi</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
</head>
```

```
<body>
```

```
<script type="text/javascript">
```

```
<!--
```

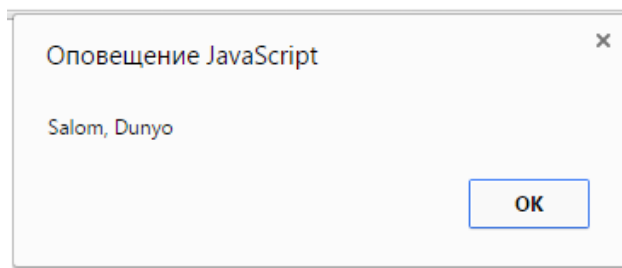
```
window.alert("Salom, Dunyo");
```

```
//-- >
```

```
</script>
```

```
</body>
```

```
</html>
```



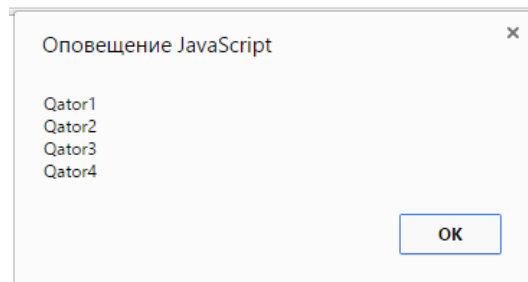
4.1.2-rasm

Xabarni g`n belgisi yordamida qatorlarga bo'lish mumkin.



4.1.3-misol. Xabarni qatorga bo'lish.

```
<!DOCTYPE HTML>
<html> <head>
<title> Xabarni qatorga bo'lish </title>
<meta http-equivq"Content-Type" contentq"text/html; charsetqutf-8">
</head>
<body>
<script typeq"text/javascript">
<!--
window.alert("Qator1g`nQator2g`nQator3g`nQator4");
//-- >
</script>
</body>
</html>
```



4.1.3-rasm

OK va Cancel tugmali xabar oynasi. confirm() uslubi OK va Cancel tugmali xabar oynasini ko'rsatish uchun ishlatiladi.



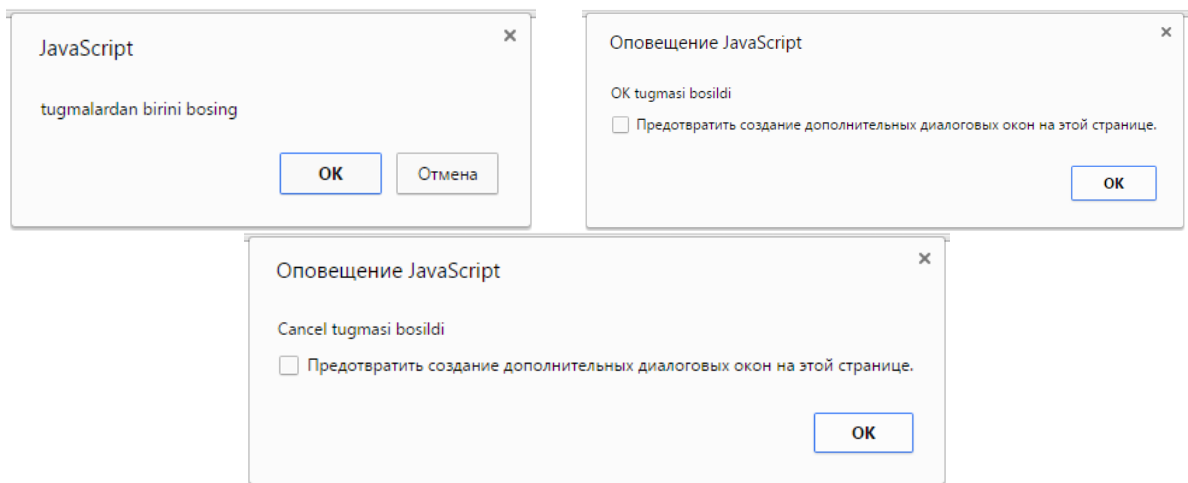
4.1.4-misol. confirm() uslubi.

```
<!DOCTYPE HTML>
<html>
<head>
<title>confirm uslubi</title>
<meta http-equivq"Content-Type" contentq"text/html; charsetqutf-8">
```

```

</head>
<body>
<script type="text/javascript">
<!--if (window.confirm("tugmalardan birini bosing"))
{ window.alert("OK tugmasi bosildi");
}
  else { window.alert("Cancel tugmasi bosildi");
}
//-- >
</script>
</body>
</html>

```



4.1.4-rasm

OK va Cancel tugmali kiritish oynasi. Prompt () uslubi kiritish maydoni va OK, Cancel tugmalari joylashlan xabar oynasini chaqirish uchun ishlatiladi. U agar Ok tugmasi bosilsa kiritilgan qiymatni o'zlashtiradi yoki Cancel bosilsa *null*ga o'zgaruvchi qiymati teng bo'ladi.



4.1.5-misol. prompt() uslubi.

```

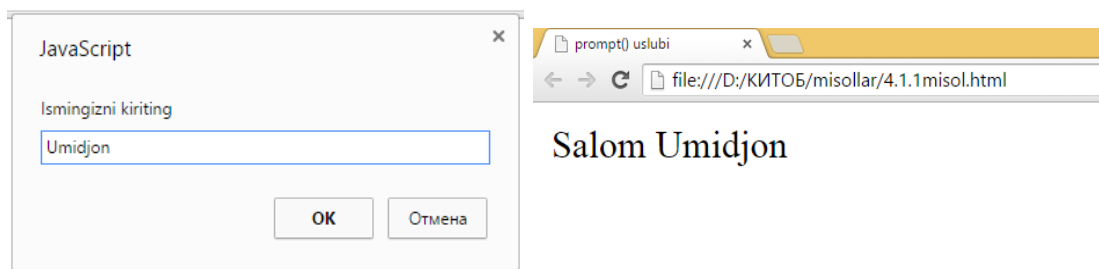
<!DOCTYPE HTML>
<html>
<head>
<title> prompt() uslubi </title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
<script type="text/javascript">
<!--
var n = window.prompt("Ismingizni kiriting", "Bu qiymat ixtiyoriy");
if (n == null) {

```

```

    document.write("Siz Cancel ni bosdingiz");
}
else {
    document.write("Salom " + Qn);
}
//-->
</script>
</body>
</html>

```



4.1.5-rasm



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. JavaScript qanday dasturlash tili?
2. JavaScript buyruqlari HTMLda qanday yoziladi?
3. JavaScript kodlarning ishlashi web brauzerga bog'liqmi?
4. Web brauzer JavaScript uchun qanday sozlanadi?
5. JavaScriptda dialogli xabar oynasi qanday chaqiriladi?
6. JavaScriptda dialogli kiritish oynasi qaysi uslub yordamida chaqiriladi?

4.2-§. JavaScriptda ma'lumot tipi va o'zgaruvchilar, ob'ektlar, ifodalar, arifmetik amallar.



O'quv modullari

o'zgaruvchi, ob'ekt, ifoda, deterministik, identifikasiya, vizualizatsiya, initsializatsiya, qiymat, ma'lumot tiplari, interpretator, funksiya, tip o'zgartirish

O'zgaruvchi – dasturda ma'lumotlarni saqlashda foydalaniladigan xotiradan ajratilgan joy. Har bir o'zgaruvchi takrorlanmas nom bilan nomlanishi va bu nom lotin alifbosi xarflari, raqamlar, belgilarda iborat bo'lishi mumkin. Birinchi belgi ixtiyoriy harf, ixtiyoriy belgi bo'lishi mumkin. O'zgaruvchi nomida \$ belgisi ham ishtirok etishi mumkin. O'zgaruvchi nomlari sifatida JavaScript tilining kalit so'zlari (zaxiraga olingan so'zlar)dan foydalanib bo'lmaydi.

O'zgaruvchining to'g'ri nomlanishi:

x, strName, y1, _nam

O'zgaruvchining noto'g'ri nomlanishi:

ly, frame

Oxirgi o'zgaruvchi nomi JavaScriptning kalit so'zi ishlatilgani uchun xato. O'zgaruvchi nomini ko'rsatishda katta va kichik haqrlar farqlanishini hisobga olish kerak. Masalan, *strName* va *strname* — boshqa boshqa o'zgaruvchilar. Dasturda o'zgaruvchilar var kalit so'zi ordamida e'lon qilinadi. *var strName;*

Bir qatorda birdaniga bir nechta o'zgaruvchini vergul bilan ajratib e'lon qilish mumkin.

```
var x, strName, y1, _name, frame1;
```

Ma'lumot tipi va o'zgaruvchilarni e'lon qilish. O'zgaruvchilar tipini aniqlash.

- ✓ *number* — butun son yoki suzuvchi vergulli son (kasr son);
- ✓ *string* — qator;
- ✓ *boolean* — ma'lumotlarning mantiqiy tipi. Uning qiymati true (chin) yoki false (yolg'on) bo'lishi mumkin;
- ✓ *function* — funksiya. JavaScript tilida qandaydir o'zgaruvchi qimatini funksiya yordamida berish mumkin. Bunday nomlanish uchun funksiya aylanma qavssiz ko'rsatilishi kerak. Bundan tashqari, funksia xususiyat va uslublar bo'lishi kerak;
- ✓ *object* — massivlar, ob'ektlar hamda **null** o'zgaruvchisi bilan beriladi.

Initsializatsiya jarayonida JavaScript o'zgaruvchilari avtomatik ravishda ajratadi. O'zgaruvchilar initsializatsiya nima? Bu o'zgaruvchiga boshlang'ich qiymatni o'zlashtirish amalidir.

O'zgaruvchilarga qiymatni o'zlashtirish «q» operatori yordamida bajariladi.

Number1 q 7; // Number1 o'zgaruvchisi 7 qimatini o'zlashtirmoqda;

Number2 q 7.8; // Number2 o'zgaruvchisi 7.8 suzuvchi vergulli qimatini o'zlashtirmoqda;

String1 q "Qator"; // String1 o'zgaruvchisi Qator o'zgaruvchisini o'zlashtirmoqda;

String2 q "Qator"; // String2 o'zgaruvchisi Qator o'zgaruvchisini o'zlashtirmoqda;

Boolean1 q true; // Boolean1 o'zgaruvchisi mantiqi qiymat true ni o'zlashtirmoqda

Str1 q null; // Str1 o'zgaruvchisi hech qanday ma'lumotga ega emas, bo'shatilgan.

O'zgaruvchilarga to'g'ridan to'g'ri uning e'lonida qiymat berish mumkin:

```
var str1 q "Qator";
```

```
var str2 q "Qator", Number1 q 7;
```

// bir nechta o'zgaruvchilarning boshlang'ich qiymatlarni to'g'ridan-to'g'ri berish mumkin.

Agar dasturda o'zgaruvchilarga murojat amalga oshmasa, o'zgaruvchilar e'lonida qandaydir kichik xatolikka yo'l qo'yilgan. Agar o'zgaruvchi e'lon

qilingan bo'lib, biroq u boshlang'ich qiymatni o'zlashtirmasa, qiymat *undefined*ga teng bo'lib qoladi.

typeof operatori o'zgaruvchilar tipini aniqlaydi. Quyidagi misolda buni ko'rib chiqamiz.

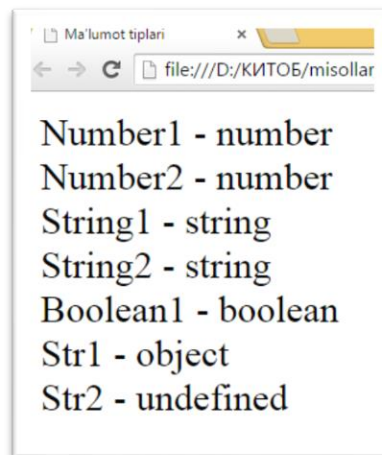


4.2.1-misol. Ma'lumot tiplari.

```
<!DOCTYPE HTML>

<html>

<head>
<title> ma'lumot tiplari </title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
<script type="text/javascript">
<!--
var Number1 q 7;
var Number2 q 7.8;
var String1 q "Qator";
var String2 q 'Qator',
var Boolean1 q true;
var Str1 q null, Str2;
document.write("Number1 - " Q typeof (Number1) Q "<br>");
document.write("Number2 - " Q typeof (Number2) Q "<br>");
document.write("String1 - " Q typeof (String1) Q "<br>");
document.write("String2 - " Q typeof (String2) Q "<br>"); //
Qavsni yozmaslik ham mumkin
document.write("Boolean1 - " Q typeof Boolean1 Q "<br>");
document.write("Str1 - " Q typeof Str1 Q "<br>");
document.write("Str2 - " Q typeof Str2;
//-- >
</script>
</body>
</html>
```



4.2.1-rasm

Ma'lumotlar tipini o'zgartirish.



Muammoli savol. Agar songa qatorni qo'shish kerak bo'lsa nima bo'ladi?

```
var Str q "5";
var Number1 q 3;
var Str2 q Number1 Q Str; // o'zgaruvchi qatordan iborat bo'ladi "35"
var Str3 q Str Q Number1; // o'zgaruvchi qatordan iborat bo'ladi "53"
```

Bu holda interpretator ma'lumotlar tipining nomuvofiqligiga duch keladi va ma'lumotlar tipini bir hil holatga olib keladi, so'ng amal bajariladi. Bizning misolimizda Number1 o'zgaruvchisi, number (sonli) tipga tegishli edi, string (qator) tipiga o'zgartirilib, keyin qatorni qo'shish amali bajariladi.



Muammoli savol. Agar sondan qator ayrilsa, songa qator ko'paytirilsa, son qatorga bo'linsa nima bo'ladi?

```
var Number1 q 15;
var Str q "5";
var Str2 q Number1 - Str; // O'zgaruvchi sondan iborat bo'ladi 10
var Str3 q Number1 * Str; // O'zgaruvchi sondan iborat bo'ladi 75
var Str4 q Number1 / Str; // O'zgaruvchi sondan iborat bo'ladi 3
```

Shunday qilib interpretator qatorni songa aylantirishga urinadi va keyin ifodani hisoblaydi.

Qator va son qanday ketma-ketlikda ko'rsatilgani muhim emas:

```
var Str5 q Str * Number1; // O'zgaruvchi qiymati bari-bir 75
```



Muammoli savol. Qator o'zgaruvchisida bir dona harf qatnashsa nima bo'ladi?

```
var Number1 q 15;
```

```
var Str q " Qator";
```

```
var Str2 q Number1 - Str; // O'zgaruvchi qiymati NaN (Not a number)
```

Bu holda interpretator qatorni o'zgartira olmaydi va o'zgaruvchiga NaN(Not a number, son emas) o'zlashtiradi.

Bir tomondan interpretator ma'lumotlar tipini o'zgartirib yuborgani biz uchun yaxshi. Boshqa tomondan esa rejalashtirilmagan natijalarga erishib qolishimiz mumkin. Bu holda o'zimiz o'zgaruvchini o'zgartirilishi kerak bo'lgan tipga o'tkazishimiz zarur bo'ladi.

Ma'lumotlar tipini o'zgartirish uchun quyidagi funksiyalardan foydalaniladi:

✓ **parseInt(<qator>, [<asos>])** qatorni butun son tipiga o'zgartiradi. Qator son tipiga o'tgaziladi, agar ikkinchi majburiy bo'lmagan "Asos" o'zgaruvchisi yozilmasa sonni o'nlik sanoq sistemasida deb hisoblanadi. Agar qatorni sonda tasvirlab bo'lmasa NaN qiymati qaytariladi:

```
var Number1 q 15;
```

```
var Str q "5";
```

```
var Str5 q "FF";
```

```
var Str2 q Number1 - parseInt(Str); // O'zgaruvchi qiymati 10 ga teng
```

```
var Str3 q Number1 - parseInt(Str5, 16); // O'zgaruvchi qiymati -240 ga teng
```

```
var Str4 q Number1 Q parseInt(Str); // O'zgaruvchi qiymati 20 ga teng
```

✓ **parseFloat(<qator>)** qatorni suzuvchi vergulli songa o'zgartiradi:

```
var Str q "5.2";
```

```
var Str2 q parseFloat(Str); // O'zgaruvchi qiymati 5.2 soniga teng
```

✓ **eval(Str)** qatorni huddi JavaScript dagi odatiy ifoda kabi hisoblaydi:

```
var Str q "3 Q 5";
```

```
var Str2 q eval(Str); // O'zgaruvchining qiymati 8 ga teng
```

Quyidagi misolda ma'lumotlar tipini o'gartirishni yi'g'indini hisoblash misolida ko'rib chiqamiz.



4.2.2-misol. Ikki son yig'indisini hisoblash.

```
<html>
```

```
<head>
```

```
    <meta charsetq"utf-8">
```

```
    <title>Tiplarni o'zgartirish</title>
```

```
</head>
```

```
<body>
```

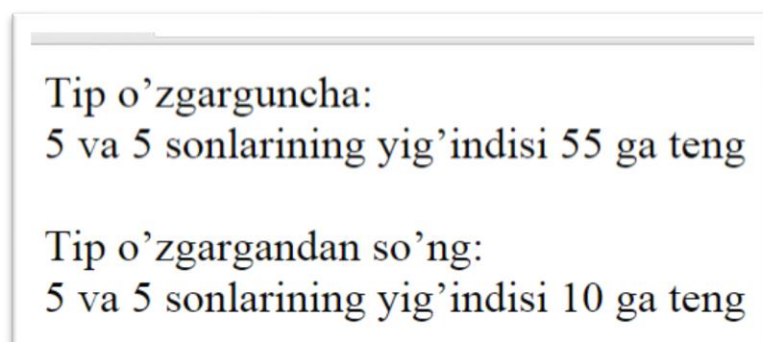
```
<script typeq"text/javascript">
```

```

var Str1, Str2, Sum1, Sum2, msg;
Str1 q window.prompt("Ikki son yi`indisini hisoblash. Birinchi sonni kiriting", "");
if (Str1qqnull){
document.write("Siz Otmena tugmaisni bosdingiz");
}
else {
Str2 q window.prompt("Ikki son yi`indisini hisoblash. Ikkinchi sonni kiriting", "");
if (Str2qqnull) {
document.write("Siz Otmena tugmaisni bosdingiz");
}
else { Sum1 q Str1 Q Str2;
msg q "Tip o'zgarguncha: <br> "Q Str1 Q " va " Q Str2 Q " sonlarining yig'indisi
";
msg Qq Sum1 Q " ga teng ";
Sum2 q parseInt(Str1) Q parseInt(Str2);
msg Qq "<br> <br>Tip o'zgargandan so'ng: <br>";
msg Qq Str1 Q " va " Q Str2 Q " sonlarining yig'indisi " Q Sum2 Q " ga teng" ;
document.write(msg);
}
}
</script>
</body>
</html>

```

Agar web-brauzerning dialogli oynasiga 5 va 5 ni kiritsak quyidagicha natija ko'rsatiladi:



Shunday qilib, dialogli oynadan kiritilgan qiymat qatordan iborat bo'ladi. Ikki son yig'indisini hisoblash uchun kiritilgan qiymatni kerakli tipga o'zgartirish kerak bo'ladi, aks holda biz oddiy qator ko'rinishidagi natijaga erishamiz.

JavaScript operatorlari. Operatorlar ma'lumotlar ustida biror amal bajarish uchun ishlatiladi. Masalan, o'zlashtirish operatori ma'lumotlarni o'zgaruvchiga o'zlashtirish, matematik operatorlar turli arifmetik amallar bajarish, qatorlarni ulash (konkatenasiya) operatorlari ikki qatorni bir qatorga birlashtirish uchun xizmat qiladi.

Operatorlar bir yoki ikki o'z o'zgaruvchi, o'zgarmas, yoki boshqa ifoda yoki tarkibiy operatorlarini ifoda qiluvchi qiymatlarni oladi va berilganlar bo'yicha bitta natijani qaytaradi. JavaScriptda qo'llaniladigan operatorlarni batafsilroq ko'rib chiqamiz.

Matematik operatorlar

- Q — qo'shish: $Z \leftarrow X + Y$;
- — ayirish: $Z \leftarrow X - Y$;
- $*$ — ko'paytirish: $Z \leftarrow X * Y$;
- $/$ — bo'lish: $Z \leftarrow X / Y$;
- $\%$ — modulli bo'lish (qoldiq qismini qaytaradi): $Z \leftarrow X \% Y$;
- QQ — inkrementa operatori. O'zgaruvchi qiymatini 1 ga orttiradi:
 ZQQ ; // $Z \leftarrow Z + 1$ ga teng kuchli;
- $--$ — dekrementa operatori. O'zgaruvchi qiymatini 1 ga kamaytiradi:
 $Z--$; // $Z \leftarrow Z - 1$ ga teng kuchli;

inkrementa va dekrementa operatorlari postfiks yoki prefiks shaklda ishlatilishi mumkin:

ZQQ ; $Z--$; // Postfiks shakl
 QQZ ; $--Z$; // Prefiks shakl

Nima farqi bor? Postfiks shaklda (zQQ) boshqa amal bajarilgach oshiriladi yoki kamaytiriladi, prefiks (QQZ) shaklda esa qiymat orttiriladi yoki kamaytiriladi keyin amal bajariladi. Farqini misol yordamida ko'rish mumkin.



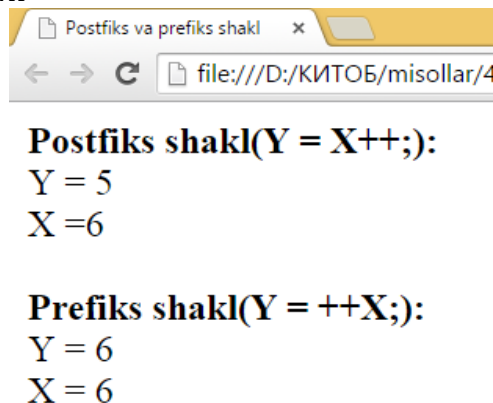
4.2.3-misol. Postfiks va prefiks shakl

```
<!DOCTYPE HTML>
<html> <head>
<title> Postfiks va prefiks shakl </title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"> </head>
<body>
<script type="text/javascript">
var X, Y;
X = 5;
```

```

Y q XQQ; // Y q 5, X q 6
var msg;
msg q "<b>Postfiks shakl(Y q XQQ;):<" Q "/b> <br> Y q ";
msgQq Y Q "<br> X q" Q X Q "<br><br>";
X q 5;
Y q QQX; // Y q 6, X q 6
msg Qq "<b> Prefiks shakl(Y q QQX;):<" Q "/b><br> Y q ";
msg Qq Y Q "<br>X q " Q X;
document.write(msg);
</script> </body> </html>

```



4.2.3-rasm

O'zlashtirish operatorlari.

- q o'zgaruvchilarga qiymat o'zlashtirish: Z q 5;
- Q q ko'rsatilgan qiymatga oshirish: Z Qq 5; // Z q Z Q 5 ga teng kuchli;
- - q ko'rsatilgan qiymatga kamaytirish: Z -q 5; // Z q Z - 5 ga teng kuchli;
- * q ko'rsatilgan qiymatga ko'paytirish: Z *q 5; // Z q Z * 5 ga teng kuchli;
- / q ko'rsatilgan qiymatga bo'lish: Z /q 5; // Z q Z / 5 ga teng kuchli;
- % q ko'rsatilgan qiymatga qoldiqli bo'lish: Z %q 5; // Z q Z%5 ga teng kuchli;

Ikkilik operatorlari.

- ~ — ikkilik inversiya: Z q ~X;
- & — ikkilik VA: Z q X & Y;
- | — Ikkilik YOKI: Z q X | Y;
- * — ikkilik istisnoli YOKI: Z q X * Y;
- << — chapga siljish – chapga bir yoki undan ortiq razryadga kichik razryadli nol bilan to'ldirilib siljish: Z q X << Y;
- >> — o'nga siljish – o'nga katta razryadli nol bilan to'ldirilib siljish:
- Z q X >> Y;

- >>> — o'ngga razryadsiz siljish: (sdvig vpravo bez ucheta znaka — sdvig vpravo na odin ili bolee razryadov s zapolneniem starshix razryadov nulyami:)
- Z q X >>> Y;

Qatorlarni qayta ishlash operatori.

Q — qatorlarni birlashtirish operatori:

var Str q " Qator1" Q " Qator2"; // Str o'zgaruvchisi qiymati "Qator1 Qator2" ga teng bo'ladi

Ko'pincha o'zgaruvchi nomi bilan berilgan qatorlarni qayta ishlashga to'g'ri keladi. Agar,

var X q " Qator1";

var Z q "Qiymat X ga teng";

ko'rinishida yozilgan bo'lsa o'zgaruvchisining qiymati "Qiymat X ga teng" bo'ladi.

Agar u boshqacharoq

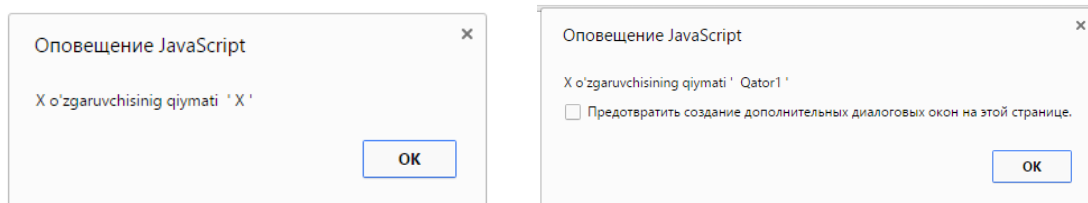
var X q " Qator1";

var Z q "O'zgaruvchi qiymati " Q X; ko'rinishda yozilgan bo'lsa z o'zgaruvchisining qiymati "O'zgaruvchi qiymati Qator1" ko'rinishda bo'ladi.



4.2.4-misol. O'zgaruvchi qiymatini chop etish

```
<!DOCTYPE HTML>
<html>
<head>
<title> O'zgaruvchi qiymatini chop etish </title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
<script type="text/javascript">
var X q " Qator1";
window.alert("X o'zgaruvchisining qiymati ' X ' "); // X o'zgaruvchisining qiymati
'X' chop etiladi
window.alert("X o'zgaruvchisining qiymati ' " Q X Q " ' "); // "X
o'zgaruvchisining qiymati 'Qator1'" chop etiladi
</script>
</body>
</html>
```



4.2.4-rasm

Operatorlarni bajarilishidagi ustunlik (bajarilish tartibi)

Ushbu ifoda qanday ketma-ketlikda bajariladi? $X \text{ q } 5 \text{ Q } 10 * 3 / 2$;

Bu operatorlarning bajarilish tartibiga bo'g'liq. Ifoda quyidagi ketma-ketlikda bajariladi:

Misolimizdagi ifodaning bajarilishi quyidagi ketma-ketlikda bo'ladi:

1. 10 soni 3 ga ko'patiriladi.
2. Olingan qimat 2 ga bo'linadi.
3. Olingan natija 5 ga qo'shiladi.
4. Natijaviy qiymat X ga o'zlashtiriladi.

Qavslar qo'yilsa amal bajarilish tartibi o'zgaradi.

Quyidagi ifodada hisoblash tartibi boshqaja bo'ladi:

$X \text{ q } (5 \text{ Q } 10) * 3 / 2$;

1. 5 soniga 10 soni qo'shiladi.
2. Olingan natija 3 ga ko'paytiriladi.
3. Olingan natija 2 ga bo'linadi.
4. Natija X o'zgaruvchisiga o'zlashtiriladi.

Operatorlarni bajarilishini kamayish tartibida ko'rib chiqamiz:

- ✓ !, ~, QQ, -- — inkor, ikkilik inversiya, increment, decrement;
- ✓ *, /, % — ko'paytirish, bo'lish, qoldiqli bo'lish;
- ✓ Q, - — qo'shish va ayirish;
- ✓ <<, >>, >>> — ikkilik siljish;
- ✓ & — ikkilik VA;
- ✓ * — ikkilik istisno YOKI;
- ✓ | — ikkilik YOKI;
- ✓ q, Qq, -q, *q, /q, %q — o'zlashtirish.

Maxsus belgilar — bu alohida belgilar ketma-ketligi bo'lib, ular chop etilmayda va qandaydir oddiy amalni bajaradi.

g`n (qatorga bo'lish) belgisi yordamida dialogli onasidagi xabarni birnecha qatorda ko'rishimiz mumkin.



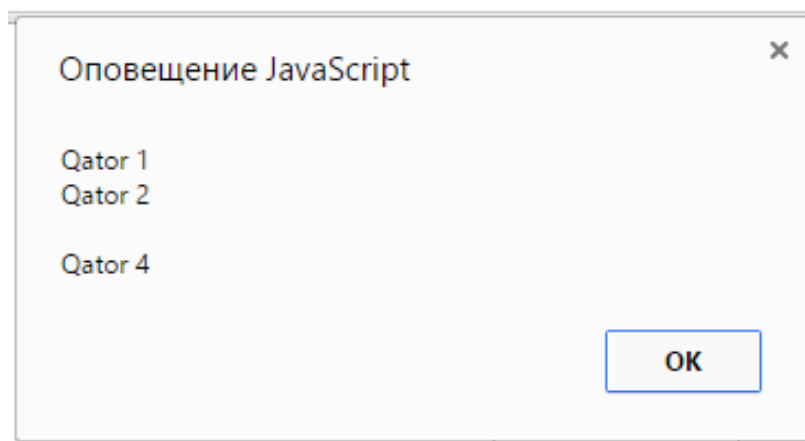
4.2.5-misol. Maxsus belgilar

<!DOCTYPE HTML>

```

<html>
<head>
<title>Maxsus belgilar</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
<script type="text/javascript">
window.alert("Qator 1\nQator 2\nQator 4");
</script>
</body>
</html>

```



4.2.5-rasm

JavaScriptdagi maxsus belgilarni sanab o'tamiz:

- g` — keying qatorga o'tkazish;
- n
- g` — Enter tugmasini bosish (yangi qatordan);
- r
- g` — sahifaga o'tish;
- f
- g` — tabulyatsiya belgisi;
- t
- g` — apostrof;
- ,
- g` — qo'shtirnoq;
- "
- g` — teskari drob belgisi (obratnaya kosaya cherta).
- g`



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. JavaScriptda qanday ma'lumot tiplari ishlatiladi?
2. JavaScriptda o'zgaruvchilar qanday e'lon qilinadi?
3. JavaScriptda o'zgaruvchilar tipi qanday aniqlanadi?

4. JavaScriptda o'zgaruvchilar tipi qanday o'zgartiriladi?
5. JavaScriptda amallarni bajarilish tartibi qanday?
6. JavaScriptda maxsus belgilar qanday ifodalanadi?

4.3-§. JavaScriptda jarayonlarini boshqarish elementlari (shartli va sikl operatorlari).



O'quv modullari

operator, kod, shart, Mantiqiy ifoda, true, false, parametr, ob'ekt klassifikatsiyasi, **shartli operator**, tanlash operatori, sikl operatori, **taqqoslash operatorlari**, identifikatsiya.

Shartli operator. Kodlar blokining shartlar bo'yicha bajarilishi. Shartli operatorlar tegishli mantiqiy ifodaning qiymatiga ko'ra dasturning alohida bir qismini bajarilish yoki bajarilmasligini ta'minlaydi. Mantiqiy ifoda faqat ikkita qiymat qabul qiladi: *true* (rost) yoki *false* (yolg'on).

Taqqoslash operatorlari. Taqqoslash operatorlari mantiqiy ifodalarda qo'llaniladi. Ularni sanab o'tamiz:

- ✓ qq — teng;
- ✓ qqq — qat'iy teng;
- ✓ !q — teng emas;
- ✓ !qq — qat'iy teng emas;
- ✓ < — kichik;
- ✓ > — katta;
- ✓ <q — kichik yoki teng;
- ✓ >q — katta yoki teng.



Muammoli savol. qq (teng) va qqq (qat'iy teng) operatorlari orasida qanday farq bor?

qq operatoridan foydalanishda agar taqqoslanayotgan ifodalar turli tipda bo'lsa, interpretator ifodalarni bir tipga o'tkazishga harakat qiladi va ularni tekshiradi.

qqq operatori esa turli tiplar taqqoslanayotgan bo'lsa birdaniga *false* (yolg'on) qiymatni qaytaradi.

Bundan tashqari mantiqiy ifoda qiymati ! operatori yordamida quyidagi ko'rinishda inkorlanishi mumkin:

!(Var1 qq Var2)

Agar Var1 va Var2 o'zgaruvchilari teng bo'lsa, rost qiymat qaytariladi. Bu ifodadan oldin ! belgisi ishlatilsa ifoda qiymati yolg'on ga aylanadi.

Bir nechta mantiqiy ifodalarni quyidagi operatorlar yordamida birlashtirish mumkin:

✓ && — mantiqiy VA;
 ✓ || — mantiqiy YOKI.
 (Var1 qq Var2) && (Var2 !q Var3)
 (Var1 qq Var2) || (Var3 qq Var4)

Birinchi ifoda har ikki mantiqiy amal natijasi true bo'lsa true qiymat qabul qiladi, ikkinchisida esa hech bo'lmaganda bittasi true bo'lsa true bo'ladi.

|| operatori ko'pincha funksiyalarda zaruriy (majburiy) bo'lamagan parametrlarini yaratish uchun foydalaniladi. Agar birinchi ifoda true bo'lmasa, ikkinchi ifoda bajariladi:

```
function f_print(str) {
    strqstr || "Qiymat tanlanmagan";
    window.alert(str); }
f_print();//"Qiymat tanlanmagan"
f_print("Qiymat ko'rsatilgan"); //"Qiymat ko'rsatilgan"
```

if...else shart operatori. Foydalanuvchi qiymat kiritishini tekshirish.
 Yo'naltirish operatoridan biz oldinroq dialogli oynada tugmaning bosilishini tekshirish misolida ko'rib chiqdik. Bunda Ok tugmasining bosilishi true qiymatini qabul qilishini bilib olish mumkin (Misol 3.17).



4.3.1-misol. Muloqot oynasidan qaysi tugma bosilganini tekshirish.

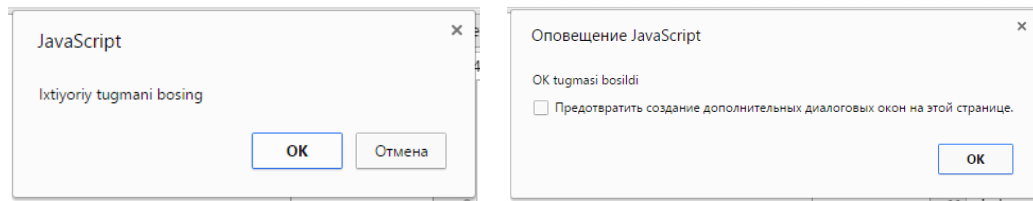
```
<!DOCTYPE HTML>
<html> <head>
<title>Xabar oynasi</title>
<meta http-equivq"Content-Type" contentq"text/html; charsetqutf-8"> </head>
<body>
<script typeq"text/javascript">

if (window.confirm("Ixtiyoriy tugmani bosing")) { window.alert("OK tugmasi
bosildi");
}
else {
window.alert(" Cancel tugmasi bosildi ");
}

</script>
```

</body>

</html>



4.3.1-rasm

Diqqat qiling, mantiqiy ifodada taqqoslash operatorlari ishlatilmayapti:

```
if (window.confirm(" Ixtiyoriy tugmani bosing")) {
```

Ushbu yozub quyidagi yozuvga teng kuchli:

```
if (window.confirm("Ixtiyoriy tugmani bosing") qq true) {
```

Ifodaning tengligi true (rost) bo'lsa ixtiyoriy tanlanadi.

if...else operatori umumiy holda quyidagi ko'rinishda bo'ladi:

```
if (<Mantiqiy ifoda>) {
```

*<Operatorlar to'plami, agar mantiqiy ifoda rost qiymat qabul qilsa bajariladi>
[else {<Operatorlar to'plami, agar mantiqiy ifoda yolg'on qiymat qabul qilsa bajariladi> }]*

Quyidagi misolni ko'rib chiqamiz :



4.3.2-misol. Sonni juftlikka tekshirish

```
<!DOCTYPE HTML>
```

```
<html> <head>
```

```
<title>Sonning juftligini tekshirish</title>
```

```
<meta http-equivq"Content-Type" contentq"text/html; charsetqwindows-1251">
```

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

```
<script typeq"text/javascript">
```

```
var x q window.prompt("Sonni kiriting", "");
```

```
if (xqqnull) {
```

```
document.write("Otmenga tugmasini bosdingiz");
```

```
}
```

```
else {
```

```
if ((parseInt(x))%2qq0) {
```

```
document.write("Juft son");
```

```
}
```

```
else {
```

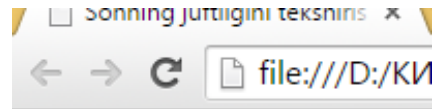
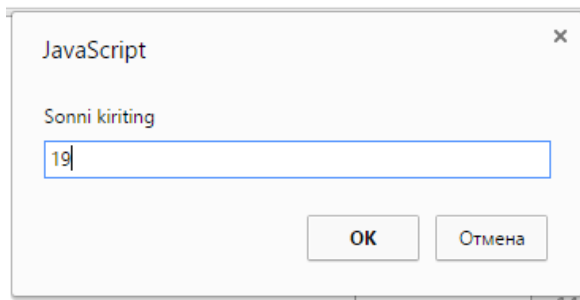
```
document.write("Toq son");
```

```
}
```

```

}
</script>
</body> </html>

```



Toq son

4.3.2-rasm

Misoldan ko'rinib turibdiki, bir shartli operator boshqa shartli operatorning ichiga joylashishi mumkin. Bundan tashqari Bundan tashqari, agar shart ichida faqat bitta ifodadan tashkil topgan bo'lsa, figurali qavsni ko'rsatish shart emas:

```
if ((parseInt(x))%2==0) document.write("Juft son"); else document.write("Toq son");
```

bundan tashqari else umuman ishtirok etmasligi ham mumkin:

```
if ((parseInt(x))%2==0) document.write("Juft son");
```

? operatori. Juftlikka tekshirish.

? operatori umumiy ko'rinishda quyidagicha yoziladi:

<O'zgaruvchi> q (<Mantiqiy ifoda>) ? <agar Rost> : <agar Yolg'on>;

Juftlikka tekshirish dasturi (Misol 3.18)ni qayata tuzamiz va bunda if...else o'rniga ? operatoridan foydalanamiz.



4.3.3-misol. ? operatori yordamida sonni juftlikka tekshirish

```

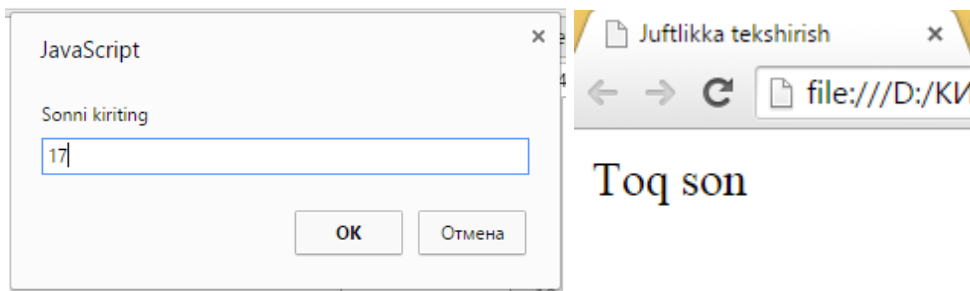
<!DOCTYPE HTML>
<html> <head>
<title>Juftlikka tekshirish</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head> <body>
<script type="text/javascript">
var x q window.prompt("Sonni kiriting", "");
if (x==null) {
document.write(" Siz Otmenga ni bosdingiz");
}
else {
var msg q ((parseInt(x))%2==0) ? "Juft son" : "Toq son";
document.write(msg);

```

```

}
</script>
</body>
</html>

```



4.3.3-rasm

switch tanlash operatori. switch tanlash operatori quyidagi shaklda yoziladi:

```

switch (<O'zgaruvchi yoki ifoda>) {
case <Qiymat 1>:<Ifoda 1>; break;
case <Qiymat 2>:< Ifoda 2>;break;
...
case <Qiymat n>:< Ifoda n>;break;
default: < Qiymat nQ1 >;
}

```

Oldingi dasturimizni if...else va ? o'rniga switch operatoridan foydalanib qayata yozamiz (4.3.4-misol).



4.3.4-misol. switch operatoridabn foydalanib sonni jutlikka tekshirish

```

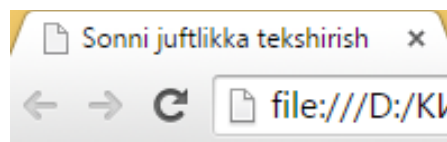
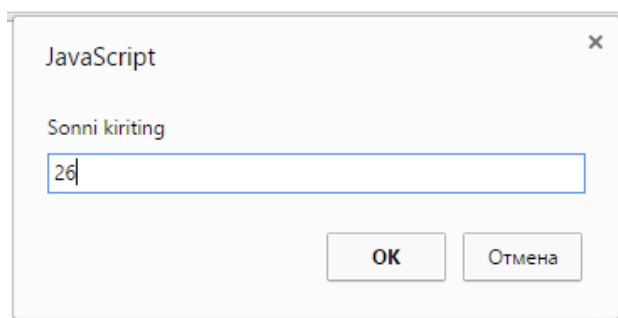
<!DOCTYPE HTML>
<html> <head>
<title>Sonni juftlikka tekshirish</title>
<meta http-equiv"Content-Type" contentq"text/html; charsetqutf-8">
</head>
<body>
<script typeq"text/javascript">
var x q window.prompt("Sonni kiriting", "");
if (xqnull) {
document.write("Siz Otmena ni bosdingiz ");
}
else {
switch ((parseInt(x))%2)
{ case 0:document.write("Juft son "); break;
case 1: document.write("Toq son"); break;

```

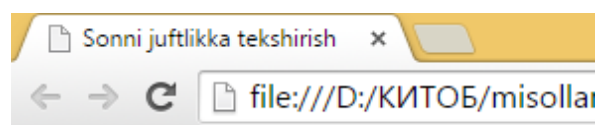
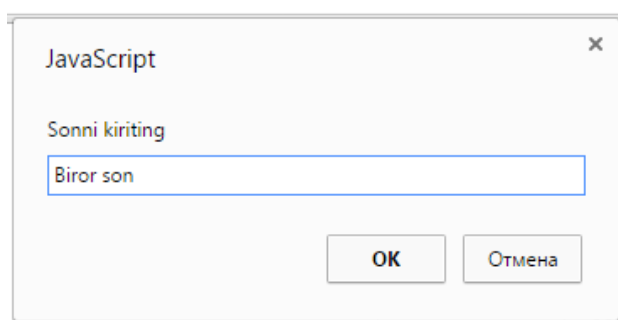
```

default:
document.write("Kiritilgan qiymat sonli emas");
}
}
</script>
</body>
</html>

```



Juft son



Kiritilgan qiymat sonli emas

4.3.4-rasm

Shunday qilib, **switch** operatori yana bir qo'shimcha tekshirishga imkon yaratdi. Agar foydalanuvchi son o'rniga qatorni kiritib yuborishi mumkin. Oldingi misolimizda bu tekshirishni bajarmagan edik va tasodifan qator kiritib yuborilsa natija "Toq son" ko'rinishda chiqar edi. Bu esa xato.

Switch operatoriga qaytamiz. Switch operatorida mantiqiy ifoda o'rniga o'zgaruvchi yoki arifmetik ifoda ishtirok etadi. O'zgaruvchi yoki ifodaning qiymatiga mos case bloklaridan biri bajariladi. Agar qiymat hech bir case bloke qiymatiga mos bo'lmasa, default bloke bajariladi. Break operatori switch tanlash operatoridan chiqib ketishni ta'minlaydi. Bu nima uchun kerak? case blokining oxirida break ko'rsatilmasa, keyingi case bloke ham bajariladi. Agar misolimizdagi barcha break operatorlarini o'chirib tashlasak, web brauzerda quyidagi natijani ko'ramiz (tashqaridan juft son kiritilsa):

Juft son Toq son Kiritilgan qiymat sonli emas

Demak, break operatori albatta har bir case blokining oxirida ko'rsatilishi kerak.

Sikllar. Kodlar blokinig ko'p marta bajarilishi. Aytaylik, 1 dan 100 gacha natular sonlarni ketma-ket chop etish kerak bo'lsin. Buning uchun bilishimiz bo'yicha quyidagi kodni 100 marta yozib chiqishimiz kerak bo'ladi:

```
document.write("1<br>");
document.write("2 <br>");
... document.write("100<br>");
```

Buni sikl yordamida qisqa va oson hal qilish mumkin:
for (var iq1; i<101; document.write(iQ"
");

for sikl operatori. for sikl operatori ifodaning aniq belgilangan son miqdori bo'icha takrorlanib bajarilishi uchun ishlatiladi. Quyidagi formatda yoziladi:

```
for (<Boshlang'ich qiymat>; <Shart>; <Orttirish>)
{ <Ifoda yoki operatorlar ketma-ketligi> }
```

Bunda:

- ✓ <Boshlang'ich qiymat> o'zgaruvchiga boshlang'ich qiymat beriladi;
- ✓ <Shart> mantiqiy ifodadan iborat bo'lib, agar uning qiymati true bo'lsa sikl ichidagi ifoda bajariladi;
- ✓ <Orttirish> o'zgaruvchiga qadam beriladi, ya'ni boshlang'ich qiymat o'zgaruvchisi ma'lum miqdordagi songa orttiriladi.

For sikli odatda quyidagicha bajariladi:

1. O'zgaruvchi-hisoblagich boshlang'ich qiymat oladi.
2. Shart tekshiriladi – agar u rost bo'lsa sikl ichidagi ifoda bajariladi, aksincha yolg'on bo'lsa sikldan chiqiladi.
3. O'zgaruvchi-hisoblagich <Orttirish> ko'rsatkichi bo'yicha o'zgaradi.
4. 2- qadamga o'tiladi.
5. Sikl <Shart> false bo'lmaguncha bajariladi. Agar shart false bo'lmasa sikl to'xtovsiz davom etadi.

<Orttirish> nafaqat o'suvchi balki kamayuvchi ham bo'lishi mumkin. 100 dan 1 gacha butun sonlarni chop etaylik:

```
For (var iq100; i>0; i--) document.write(iQ"<br>");
```

<Orttirish> nafaqat bir birlikka boshqa butun sonlarda oshirilib yoki kamaytirilishi mumkin. 1 dan 100 gacha juft sonlarni chop etaylik:

```
for (var iq2; i<101; iQq2) document.write(i Q "<br>");
```

Ifoda <Shart> parametrda ko'rsatilgan har bir iteratsiya hisoblanadi.

Massiv elementlarini chop etishni ko'rib chiqaylik:

```
var Mass q [1, 2, 3];
```

```
for (var iq0; i<Mass.length; iQQ){
```

```
if (iQQ0) {
```

```
    Mass.push(4); // Yangi elementlarni qo'shamiz
```

```
    Mass.push(5); // isbot uchun
```

```
}
```

```
document.write(Mass[i] Q " "); } // Natija: 1 2 3 4 5
```

Bu misolda <Shart> parametrida length hususiyatini kop'rsatdik, sikl ichida esa massivga yangi element qo'shdik. Natijada yangi elementlar kiritilib massivning barcha elementlarini o'z ichiga olgan yangi massiv pado bo'ldi:

```
var Mass q [1, 2, 3];  
for (var i q 0, cqMass.length; i < c; i QQ) {  
    if (i qq 0) {  
        Mass.push(4); // Yangi elementlarni qo'shamiz  
        Mass.push(5); // isbot uchun  
    }  
    document.write(Mass[i] Q " "); } // Natija: 1 2 3
```

While sikli. While sikli unda ko'rsatilgan mantiqiy ifada rost bo'lsa bajariladi. Quyidagi umumiy formatda yoziladi:

```
<Boshlang'ich qiymat>;  
while (<Shart>) {  
    <ifoda>;  
    <Orttirish>;  
}
```

While sikli quyidagi tartibda bajariladi:

1. O'zgaruvchi – hisoblagich boshlang'ich qiymat o'zlashtiradi.
 2. Shart tekshiriladi – agar u rost bo'lsa, sikl ichi bajariladi, aksincha bo'lsa sikl tugatiladi.
 3. O'zgaruvchi – hisoblagich <Orttirish> ifodasida ko'rsatilgandek o'zgaradi.
 4. 2-bandga o'tiladi.
- while siklidan foydalanib 1 dan 100 gacha bo'lgan sonlarni chop etaylik (4.3.5-misol).



4.3.5-misol. While sikli

```
<!DOCTYPE html>  
<html>  
<body>  
<script type="text/javascript">  
var i q 1;  
while (i < 101)  
{  
    document.write(i Q " ");  
    i QQ;  
}  
</script>  
</body>  
</html>
```

1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60; 61; 62; 63; 64; 65; 66; 67; 68; 69; 70; 71; 72; 73; 74; 75; 76; 77; 78; 79; 80; 81; 82; 83; 84; 85; 86; 87; 88; 89; 90; 91; 92; 93; 94; 95; 96; 97; 98; 99; 100;

4.3.5-rasm

DIQQAT! Agar <Orttirish> ko'rsatilmasa sikl to'xtovsiz bajariladi. <Orttirish> faqat arifmetik ifoda bo'lishi shart emas. Masalan, ma'lumotlarbazasi bilan ishlashda <Orttirish> qiymati qatorlarnio o'zgartirish sifatida aniqlanishimumkin. Bunda <Boshlang'ish qiymat> ma'lumotlar bazasining birinchi qatori bo'ladi.

Do while sikli. do...while siklida ham mantiqi ifoda rost bo'lganda sikl bajariladi. While siklidan farqi unda shart sikl oxirida tekshiriladi. Demak, do..while sikli kamida bir marta albatta bajariladi. Quyidagi umumiy formatda yoziladi:

<Boshlang'ish qiymat>;

do {

<Ifoda>; <Orttirish>; } while

(<shart>);

do..while sikli quyidagi tartibda bajariladi:

1. O'zgaruvchi – hisoblagich boshlang'ich qiymat o'zlashtiradi.
2. Sikl ichidagi ifoda bajariladi.
3. O'zgaruvchi – hisoblagich <Orttirish> ifodasida ko'rsatilgandek o'zgaradi.
4. Shart tekshiriladi – agar u rost bo'lsa 2-bandga o'tiladi, aksincha bo'lsa sikl tugatiladi.

do...while siklidan foydalanib 1 dan 100 gacha bo'lgan sonlarni chop etaylik.



4.3.6-misol. Do...while sikli

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
var i q 1;
```

```
do {
```

```
document.write(i Q " ");
```

```
iQQ;
```

```
} while (i<101);
```

```
</script>
```

```
</body>
```

</html>

1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60; 61; 62; 63; 64; 65; 66; 67; 68; 69; 70; 71; 72; 73; 74; 75; 76; 77; 78; 79; 80; 81; 82; 83; 84; 85; 86; 87; 88; 89; 90; 91; 92; 93; 94; 95; 96; 97; 98; 99; 100;

4.3.6-rasm

DIQQAT! Agar <Orttirish> ko'rsatilmasa sikl to'xtovsiz bajariladi.

Continue operatori. Keyingi siklning navbatdagi iterasiyasiga o'tish. Continue operatori sikl ichidagi ifodalar oxirigacha bajarilmasdan navbatdagi iterasiyasiga o'tish imkonini beradi. Bu operator ixtiyoriy siklda qo'llanilishi mumkin.

1 dan 100 gacha bo'lgan va 5 dan 10 gacha bo'lgan sonlardan tashqari sonlarni chop etish.



4.3.7-misol. continue operatoridan foydalanish

```
<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">
for (var iq1; i<101;iQQ){
  if (i>4 && i<11) continue;
  document.write(iQ " ");
}
</script>
</body>
</html>
```

1; 2; 3; 4; 11; 12; 13; 14; 15; 16; 17; 18; 19; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44; 45; 46; 47; 48; 49; 50; 51; 52; 53; 54; 55; 56; 57; 58; 59; 60; 61; 62; 63; 64; 65; 66; 67; 68; 69; 70; 71; 72; 73; 74; 75; 76; 77; 78; 79; 80; 81; 82; 83; 84; 85; 86; 87; 88; 89; 90; 91; 92; 93; 94; 95; 96; 97; 98; 99; 100;

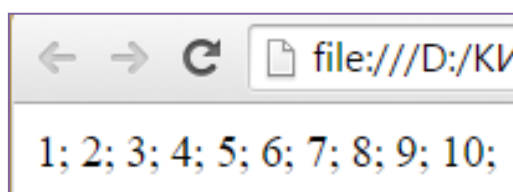
4.3.7-rasm

Break operatori. Siklni to'xtatish. break operatori siklni zudlik bilan to'xtatish uchun ishlatiladi. 1 dan 100 gacha sonlarni yana bir yo'l bilan chop etaylik.



4.3.8-misol. Siklning to'xtatilishi.

```
<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">
for (var i=1; true; i++) { if (i>10) break; document.write(i + " "); }
</script> </body>
</html>
```



4.3.8-rasm

Bu yerda biz shart sifatida true ni ishlatdik, bu esa biz sikl ichida break operatoridan foydalanmaganimizda siklni adoqsiz bajarilishiga olib kelar edi.

Break operatori dasturning emas, siklning bajarilishini to'xtatadi. Sikldan keyingi navbatdagi ifoda bajarilaveradi.



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. JavaScriptda taqqoslar amallari qanday?
2. JavaScriptda ? operatoridan foydalanishtartibini tushintiring?
3. JavaScriptda switch tanlash operatori ishlatilishini tushintiring?
4. JavaScriptda sikllar qanday ifoda qilinadi?
5. JavaScriptda for uslubi yordamida sikllarni tashkil etish qanday bajariladi?
6. JavaScriptda while va do.. while qanday tartibda ishlatiladi?
7. Qanday hollarda for, qanday hollarda while, qanday xollarda do... while ishlatgan qulay?

4.4-§. Funksiyalar. Dasturning qismlarga bo'linishi.



O`quv modullari

*Funksiya, sinf, hususiyat, metod, global sinf, number sinfi, string sinfi, date sinfi, array sinfi, **assosiativ massiv**, math sinfi*

Funksiya — dasturning ixtiyoriy qismidan murojaat qilish mumkin bo'lgan JavaScriptning qism dasturidir. Funksiya Function kalit so'zi yordamida quyidagi umumiy ko'rinishda yoziladi:

```
function <Funksiya nomi>([<Parametrlar>]){<Funksiya tanasi> [return <Qiymat>]}
```

Asosiy tushunchalar.

Funksiya takrorlanmas nom bilan nomlanishi kerak. Funksiyani nomlashda uchun funksiya bajaradigan masala mohiyatidan kelib chiqish maqsadga muvofiq. Ko'rgazmali bo'lishi uchun barcha funksiya nomlari bu kitobda "f_" bilan boshlanadi.

Funksiya nomidan keyin aylanma qavs ichida bir yoki bir nechta parametrlar vergul bilan ajratilganholda ko'rsatilishi kerak. Parametrlar umuman ishtirok etmasligi ham mumkin. Bunday holda faqat aylanma qavs yoziladi holos.

Figurali qavs ichida JavaScript buyruqlari funksiya tanasi sifatida yoziladi. Bundan tashqari, funksiyaga murojaat joyini belgilash uchun return kalit so'zidan foydalaniladi.

Parametrsiz funksiyaga misol:

```
function f_Alert_OK() {  
window.alert("Amallar muvaffaqiyatli bajarilgani haqida habar");
```

Parametrli funksiyalarga misol:

```
function f_Alert(msg) {  
window.alert(msg);
```

```
}
```

Parametrli funksiyalarga misol, ikki son yig'indisini hisoblash:

```
function f_Sum(x, y) {
```

```
var z = x + y;
```

```
return z;
```

```
}
```

Return tuzilmasi nafaqat o'zgaruvchi nomini balki, ifoda qiymatini ham qaytaradi:

```
function f_Sum(x, y) {
```

```
return (x + y);
```

```
}
```

Dasturda funksiyaga quyidagicha murojaat qilinadi:

```
f_Alert_OK();
```

```
f_Alert("Habar");
Var1 q f_Sum(5, 2); // Var1 o'zgaruvchisi 7 qiymatini o'zlashtiradi
return <qiymat> dan keying ifoda hech qachon bajarilmaydi:
function f_Sum(x, y) {
return (x Q y);
window.alert("Xabar"); // Bu ifoda hech qachon bajarilmaydi
}
```

Funksiyaning qiymatidan o'tadigan o'zgaruvchining nomi funktsiyaning ichidagi o'zgarmaydigan nomiga mos kelmasligi mumkin:

```
function f_Sum(x, y)
{
    return (x Q y);
}
```

```
var Var3, Var1 q 5;
var Var2 q 2;
Var3 q f_Sum (Var1, Var2);
```

Funksiyaga murojaat biror o'zgaruvchida saqlanishi mumkin. Bunday nomlanish aylanma qavssiz bo'ladi:

```
function test() {
    window.alert("Bu funksiya test()");
}
```

```
var x;
x q test; // funksiyaga murojaat o'zlashtirildi
x(); // test() funksiyasiga x o'zgaruvchisi orqali murojaat qilinmoqda
```

Bundan tashqari, funksiya umuman nomsiz bo'lishi mumkin. Misolimizda anonym funksiyaga havola o'zgaruvchida saqlanmoqda:

```
var x q function() { // anonym funksiyaga o'zgaruvchi havolasi
window.alert("Xabar");
};
x(); // anonim funksiyaga x o'zgaruvchisi orqali murojaat.
```

Ichki o'tgan funktsiyaga havola qaytarish so'zida qiymat sifatida qaytarilishi mumkin. Ichki ichki funktsiyani chaqirish uchun, ikki marta ko'rsatiladi:

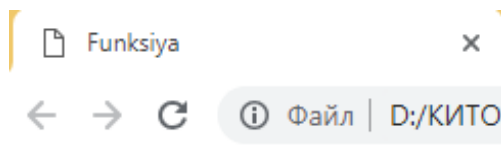
```
var x q function() { // anonim funksiyaga havola bog'lanmoqda
return function() { // qiymat funksiasiga havola qaytarilmoqda
window.alert("Bu qiymat funksiya");
};
};
x(); // X o'zgaruvchisi orqali qiymat funksiyaga murojaat qilinmoqda
```

HTML-hujjatda funktsiyaning joylashishi.

Odatda funksiya HTML hujjatning HEAD bo'limiga joylashtiriladi yoki alohida *.js kengatmali faylda yozilib saqlanadi. Funksiya BODY bo'limida joylashishi mumkin. (4.4.1 va 4.4.2-misollar).

4.4.1-misol. Funksiyaning HEAD bo'limida joylashishi.

```
<!DOCTYPE HTML>
<html>
<head> <title>Funksiya</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script type="text/javascript">
function f_Sum(x, y){
    return (x + y); }
</script>
</head>
<body>
<script type="text/javascript">
var Var3, Var1 = 7, Var2 = 3;
Var3 = f_Sum(Var1, Var2);
document.write(Var3);
</script>
</body>
</html>
```



10

4.4.1-rasm

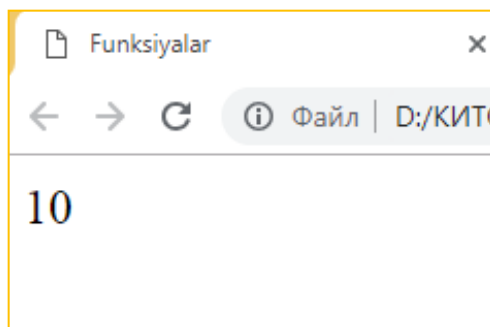
4.4.2-misol. Funksiyaning alohida script.js faylida joylashishiga misol

```
<!DOCTYPE HTML>
<html> <head> <title>Funksiyalar</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script type="text/javascript" src="script.js"></script>
</head>
<body>
<script type="text/javascript">
var Var3, Var1 = 7, Var2 = 3;
Var3 = f_Sum(Var1, Var2);
document.write(Var3);
</script>
```

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

script.js faylining tarkibi

```
function f_Sum(x, y) {
return (x Q y);
}
```



4.4.2-rasm

script.js faylini Bloknот yordamida yaratish mumkin.

Rekursia. Faktorialni hisoblash.

Rekursia – bu funksiyaning o'zida-o'ziga murojat qila olish imkoniyatidir. Bir tomondan bu qulay, boshqa tomondan agar chiqish sharti chorasi ko'rilmagan bo'lsa siklni davomiyligi cheksiz bo'lib qoladi. Misolimizda faktorialni hisoblash dasturi keltirilgan (4.5.3-misol).

4.4.3-misol. Faktorialni hisoblash

```
<!DOCTYPE HTML>
<html>
<head> <title> Faktorialni hisoblash </title>
<meta http-equiv"Content-Type" contentq"text/html; charsetqutf-8">
<script typeq"text/javascript">
```

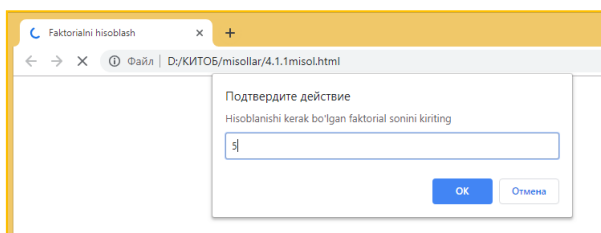
```
function f_Factorial(x) {
if (x qq 0 || x qq 1) return 1;
else return (x * f_Factorial(x - 1));
}
```

```
</script>
</head>
<body>
<script typeq"text/javascript">
```

```

var z;
z q window.prompt("Hisoblanishi kerak bo'lgan faktorial sonini kiriting", "");
if (zqnull) {
document-write("Otmena berildi");
}
else {
document.write(z Q" sonning faktoriali q ");
document.write(f_Factorial(parseInt(z)));
}
</script>
</body>
</html>

```



4.4.3-rasm



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. *Funksiya nima?*
2. *JavaScriptda funksiya qanday e'lon qilinadi?*
3. *JavaScriptda funksiya parametrlari qanday beriladi?*
4. *JavaScriptda rekursiv funksiyadan foydalanish qanday bajariladi?*

4.5-§. JavaScriptda OYD asoslari (standart, brauzer va dasturchi tomonidan yaratilgan ob'ektlar).



O`quv modullari

*sinf, hususiyat, metod, global sinf, number sinfi, string sinfi, date sinfi, array sinfi, **assosiativ massiv**, math sinfi*

JavaScriptda sinflar. Asosiy tushunchalar.

Sinf — bu o'zgaruvchi va funksiyalarni o'z ichiga qamrab oluvchi bu o'zgaruvchilarni boshqarish uchun obyekt tipidir. O'zgaruvchi *hususiyat*, funksiya esa *metod* deb nomlanadi.

Sinflarning metod va hususiyatlaridan foydalanishda sinf nusxasini yaratish zarur. Buning uchun new operatoridan foydalaniladi, keyin nusxa ma'lumotlariga

tegishli uning sinf nomi ko'rsatiladi. Sinf nomidan so'ng aylanma qavs ichida bir necha parametrlarni berish mumkin. Umumiy holda quyidagicha yoziladi:

```
<Sinf nusxasi> q new <Sinf nomi> ([<parametrlar>]);
```

Sinf nusxasini yaratishda havola (ko'rsatkich) o'zgaruvchida saqlanadi. Foydalanilayotgan havola yaratilgan sinf nusxasi hususiyat va metodlariga murojat qilishi mumkin.

Hususiyatga murojaat quyidagicha amalga oshiriladi:

```
<Sinf nusxasi>.<Hususiyat nomi>;
```

Uslubga murojaat ham shunga o'xshash amalga oshiriladi, faqat uslub nomidan keyin aylanma qavs ko'rsatiladi:

```
<Sinf nusxasi>.<Uslub nomi>();
```

Qavs ichida ko'pincha uslub parametric ko'rsatiladi.

Global sinfi. Global sinfi hususiyat va uslubidan foydalanish sinf nusxasini yaratishni talab qilmaydi. Berilgan sinfning hususiyat va uslublari JavaScript qurilgan fuksiyalari bo'ladi.

Hususiyat:

- ✓ NaN qiymati NaN dan iborat (Not a Number, son emas):

```
var x q NaN;
```

- ✓ Infinity "plyus cheksizlik" qiymatini qaytaradi:

```
var x q Infinity;
```

Uslublar:

- ✓ parseInt(<Qator>, [<Asos>]) qatorni asosda ko'rsatilgan sanoq sistemasidagi songa o'zgartiradi. Agar asos berilmasa o'nlik sanoq sistemasidagi songa o'tkaziladi. Agar qatorni songa o'zgartira olmasa NaN qiymatiga ega bo'ladi. Masalan:

```
var Number1 q 15; var Str q "5"; var Str5 q "FF";
```

```
var Str2 q Number1 - parseInt(Str); // O'zgaruvchi qiymati 10 ga teng
```

```
var Str3 q Number1 - parseInt(Str5,16); // O'zgaruvchi qiymati -240 ga teng
```

```
var Str4 q Number1 Q parseInt(Str); // O'zgaruvchi qiymati 20 soniga teng
```

- ✓ parseFloat(<qator>) qatorni suzuvchi vergulli songa o'zgartiradi:

```
var Str q "5.2";
```

```
var Str2 q parseFloat(Str); // O'zgaruvchi qiymati 5.2 soniga teng
```

- ✓ eval(<qator>) qatorli ifodani JavaScriptdagi oddiy ifoda kabi hisoblaydi:

```
var Str q "3 Q 5";
```

```
var Str2 q eval(Str); // O'zgaruvchi 8 sonini o'ziga oladi
```

- ✓ isNaN(<Ifoda>) ifodani songa tekshiradi. Agar ifoda NaN ga teng bo'lsa true, ifoda qiymati son bo'lsa false qiymatni qaytaradi;

✓ `isFinite(<ifoda>)` yakunlovchi sonni tekshiradi. `True` yoki `false` qiymat qaytaradi;

✓ `escape(<qator>)` qatorni o'n oltilik sistema kodida kodlaydi:

```
var Str q escape(" Salom");
```

```
// Str q %20%u0421%u0430%u043B%u043E%u043C
```

✓ `unescape(<qator>)` `escape()` ga teskari uslub:

```
var Str q unescape("%20%u0421%u0430%u043B%u043E%u043C"); // Str q Salom
```

IZOX. `escape()` va `unescape()` funksiyalari eskirgan. Ularning o'rniga `encodeURIComponent()` va `decodeURI()` yoki `encodeURIComponent()` va `decodeURIComponent()` dan foydalaniladi.

✓ `encodeURIComponent(URL-manzil)` butun URL-manzil kodlanadi:

```
var Str q "test.php?idq5&nqAlisher";
```

```
window.alert(encodeURIComponent(Str));
```

```
//test.php?idq5&nq%D0%90%D0%BB%D0%B8%D1%88%D0%B5%D1%80
```

✓ `decodeURI(<Qator>)` qator koddan ochiladi, `encodeURIComponent()` uslubi bilan kod;

✓ `encodeURIComponent(<qator>)` qatorni URL-kodlash bajariladi:

```
var Str q encodeURIComponent("Qator");
```

```
//Str q %D0%A1%D1%82%D1%80%D0%BE%D0%BA%D0%B0
```

`encodeURIComponent()` funksiyasidan farqi o'n oltilikdagi barcha maxsus belgilarni o'zgartiradi:

```
var Str q "test.php?nameqAliser"; window.alert(encodeURIComponent(Str));
```

```
//test.php%3Fid%3D5%26n%3D%D0%90%D0%BB%D0%B8%D1%88%D0%B5%D1%80
```

Number sinfi. Sonlar bilan ishlash. `Number` sinfi sonli o'zgaruvchi va o'zgarmaslarni ifodalashda ishlatiladi. Sinf nusxasi quyidagi ko'rinishda yaratiladi:

`<Sinf nusxasi> q new Number (<Boshlang'ich qiymat>);`

`Number` sinfi xususiyati `sinf nusxasini` yaratmay foydalanish mumkin:

➤ `MAX_VALUE` — JavaScript da ruxsat etilgan maksimal son:

```
var x q Number.MAX_VALUE; // 1.7976931348623157eQ308
```

➤ `MIN_VALUE` — JavaScript da ruxsat etilgan minimal son:

```
var x q Number.MIN_VALUE; // 5e-324
```

➤ `NaN` — `NaN` qiymati:

```
var x q Number.NaN; // NaN
```

➤ `NEGATIVE_INFINITY` — "minus cheksizlik":

```
var x q Number.NEGATIVE_INFINITY; // -Infinity
```

➤ `POSITIVE_INFINITY` — "plyus cheksizlik":

```
var x q Number.POSITIVE_INFINITY; //Infinity
```

Metodlar:

➤ valueOf() sinf nusxasining sonli qiymatini qaytaradi:

```
var x q new Number(15);
```

```
var y q x.valueOf(); // 15
```

```
document.write(typeof y); // number
```

➤ toString() sonni qatorli ko'rinishda ifodalaydi:

```
var x q new Number (15);
```

```
var Str q x.toString(); // "15"
```

```
document.write(typeof Str); // string
```

String sinfi. Qatorlarni qayta ishlash. String sinfi qatorlarni qayta ishlash uslublariga yo'l ochib beruvchi sinf. Sinf nusxasi quyidagi ko'rinishda yaratiladi:

<Sinf nusxasi> q new String (<Qator>);

Siz qatorlar qo'shtirnoq yoki apostrof orqali ifoda qilinishini yaxshi bilasiz:

```
var Str1 q "Qator 1";
```

```
var Str2 q 'Qator 2';
```

Bu usul bilan xosil qilingan qatorlar string ma'lumotlar tipiga tegishli bo'ladi, string sinfi nusxasini yaratishda esa ma'lumotlar tipi object tipiga tegishli bo'ladi:

```
var Str1 q "Qator 1";
```

```
var Str2 q 'Qator 2';
```

```
var Str3 q new String ("Qator 3");
```

```
document.write(typeof Str1); // string
```

```
document.write(typeof Str2); // string
```

```
document.write(typeof Str3); // object !
```

Shunga qaramasdan oddiy qatorlarga string sinfi uslublarini qo'llash mumkin:

```
var Str q "Qator".toUpperCase(); // belgilarni yuqori registrga o'tkazish
```

```
document.write(Str); // "QATOR"
```

```
document.write(typeof Str); // string
```

toUpperCase() uslubidan foydalanishda qator string ma'lumotlar tipiga ega bo'ladi. Qator toUpperCase()uslubidan foydalanishda joriy string tipi avtomatik ravishda String sinfi nusxasiga aylantiriladi.

length hususiyati qator uzunligini aniqlaydi:

```
var Strqnew String ("Hello, world");
```

```
document.write(Str.length); // 12
```

String sinfi ob'ektlarining uslublari quyidagilar:

✓ toString() va valueOf() qatorli qiymat qaytaradi:

```
var Str q new String ("Hello, world");
```

```
var Str2 q Str.toString();
```

```
document.write(Str2); // "Hello, world"
```

- document.write(typeof Str); //object document.write(typeof Str2); // string*
- ✓ *charAt(<belgi raqami>)* ko'rsatilgan raqamdagi belgini chiqaradi. Qatorda belgilarni raqamlash noldan boshlanadi:
var Str q "Hello, world";
document.write(Str.charAt(0)); // "N"
 - ✓ *charCodeAt(<belgi raqami>)* qatordagi ko'rsatilgan parametrdagi belgi kodini chiqaradi. Qatorda belgilarni raqamlash noldan boshlanadi:
var Str q "Hello, world"; window.alert(Str.charCodeAt(0)); // 72
 - ✓ *fromCharCode(<kod1>, ..., <kodN>)* ko'rsatilgan kodlardan qator yasaydi:
var S q String.fromCharCode(1055, 1088, 1080, 1074, 1077, 1090);
window.alert(S); // "Privet"
 - ✓ *toLowerCase()* qator belgilarini quyi regist belgilariga o'zgartiradi:
var Str q "SALOM, DUNYO";
Str q Str.toLowerCase();
document.write(Str); // "salom, dunyo"
 - ✓ *toUpperCase()* qatordagi regist belgilarini yuqori registrli belgilarga o'zgartiradi:
var Str q "salom, dunyo";
Str q Str.toUpperCase();
document.write(Str); // "SALOM, DUNYO"
 - ✓ *substr(<boshlang'ich qism>, [<qism uzunligi>])* qatordan berilgan uzunlikdagi qismni chiqarib oladi. Agar ikkichi parametr berilmasa, qator oxirigacha barcha belgilar olinadi:
var Str q "Salom, dunyo";
document.write(Str.substr(0, 5)); // "Salom" document.write(Str.substr(7)); // "dunyo"
 - ✓ *substring(<Boshlang'ich qism>, <Oxirgi qism>)* satrdan ko'rsatilgan pozitsiyadagi satr ostini ajratib olish uchun ishlatiladi. Bu holda satrning boshlang'ich va oxirgi pozitsiyasini ko'rsatiladi. Qismdagi oxirgi belgi kirmaydi:
var Str q "Salom, dunyo";
document.write(Str.substring(6, 12)); // "dunyo"
 - ✓ *indexOf(<Satr osti>, [<Qidirishning boshlang'ich pozitsiyasi>])* satrdan satr ostining birinchi uchraydigan pozitsiyasi, agar ikkinchi parametr berilmasa, qidirish satr boshidan boshlanadi. Agar satr osti umuman uchramasa, -1 qiymatni qaytaradi:
var Str q "Salom, dunyo";
document.write(Str.indexOf("lo")); // 2 document.write(Str.indexOf("lo", 5)); // -1
 - ✓ *lastIndexOf(<Satr osti>, [<Qidirishning boshlang'ich pozitsiyasi>])* joriy satrdan berilgan satr ostining oxirgi joylashgan pozitsiyasini aniqlaydi. Agar

ikkinchi parametr berilmasa, qidirish satr boshidan boshlanadi. Agar satr osti topilmasa -1 qiymatni qaytaradi:

```
var Str q "Salom, dunyo";
```

```
document.write(Str.lastIndexOf("o")); // 11
```

- ✓ split(<bo'lgich>, [<Limit>]) bo'lgich simvol yordamida sartni satrli massivlarga ajratadi. Agar ikkinchi parametr ishtirok etmasa, natijaviy massiv maksimal elementlar soniga ega bo'ladi:

```
var Str q "Salom, dunyo";
```

```
var Mass q Str.split(",");
```

```
document.write(Mass[0]); // "Salom" — massivning birinchi elementi
```

```
document.write(Mass[1]); // "dunyo" — massivning ikkinchi elementi
```

- ✓ search(<Doimiy ifoda>) doimiy ifoda bilan mos keluvchi satr osti kirish pozitsiyasi tartibini aniqlaydi.

Array sinfi. Massivlar bilan ishlash va ularni saralash.

Massiv bu – o'zgaruvchilarning raqamlangan(tartiblangan) to'plamidir. Massivdagi o'zgaruvchilar uning elementi sifatida, ularni o'rni esa indekslar bilan belgilanadi. Massiv elementlarning tartibi 0 dan boshlanadi. Buni yodda saqlash kerak. Umumi elementlar miqdori massivning **o'lchami** deb nomlanadi.

Massiv elementlari kvadrat qavs ichiga o'zaro vergul bilan ajratib yoziladi:
mass1 q [1, 2, 3, 4];

Massiv elementining qiymatini biror o'zgaruvchiga o'zlashtirishda massiv nomidan keyin indeksi kvadrat qavs ichida yoziladi:

```
Str q mass1[0]; // Str o'zgaruvchisiga massivning 0 chi elementi qiymati – 1 o'zlashtiradi;
```

4.4.1-misolda massiv tuzish va uning elementlarini web-brauzer oynasida chop etish ko'rsatilgan.



4.5.1-misol. Massivlar

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title>Massivlar</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
</head>
```

```
<body>
```

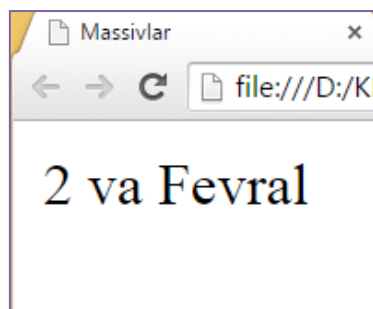
```
<script type="text/javascript">
```

```
var Mass1, Mass2;
```

```
Mass1 q [1, 2, 3, 4];
```

```
Mass2 q ["", "Yanvar", "Fevral", "Mart", "Aprel", "May", "Iyun", "Iyul",  
"Avgus", "Sentabr", "Oktabr", "Nayabr", "Dekabr"];
```

```
document.write(Mass1[1] Q " va " Q Mass2[2]);
</script>
</body>
</html>
```



4.5.1-rasm

Massivga yangi elementlar qo'shish yoki mavjudlarini o'zgartirish mumkin:
 Mass1[5] q 6; Mass1[0] q 0;

Bunda massivning ikkita elementi yaratildi va mavjud bir elementinig qiymati o'zgartirildi. **Nimaga ikkita element yaratildi?** Misolimizda 4 elementdan iborat, oxirgi elementning tartibi (indeksi) 3 bo'lgan massiv mavjud edi. Biz 5-tartibdagi massivni yaratganimizda 4-element avtomatik yaratiladi va unga undefined (aniqlanmagan) qiymat o'zlashtiriladi.

Massivning ixtiyoriy elementiga boshqa massivni o'zlashtirish mumkin:

```
Mass1[0] q [1, 2, 3, 4];
```

Bunda holda massivning indeksi ikkita sondan iborat bo'ladi:

```
Str q Mass1[0][2]; // Str o'zgaruvchisiga 3 o'zlashtiriladi
```

O'zlashtirish amalida massivlarga bog'lanish saqlanishini xisobga olish kerak. Misol orqali ko'rib chiqalik.

```
var Mass1, Mass2;
```

```
Mass1 q [1, 2, 3, 4];
```

```
Mass2 q Mass1; // massivga bo'glanish!!! (Prisvaivaetsya sso`lka na massiv!!!)
```

```
Mass2[0] q "Yangi qiymat";
```

```
document.write(Mass1.join(", ") Q "<br>");
```

```
document.write(Mass2.join(", "));
```

Array sinfi massiv yaratishga obyekt sifatida murojaat qilish va uning metodlariga murojaat qilish imkoniyatini yaratadi.

Sinf nusxasi quyidagi usulda yaratilishi mumkin:

<Sinf nusxasi> q new Array (<Massiv elementlari soni>);

<Sinf nusxasi> q new Array (<Vergul bilan ajratilgan massiv elementlari>);

Agar aylanma qavs ichida hech qanday parameter bo'lmasa, u holda massiv yaratilguncha nol uzunlikda elementsiz bo'ladi. Agar bitta son ko'rsatilsa, bu son massiv elementlari sonini bildiradi. Agar bir nechta elementlar vergul bilan ko'rsatilsa ular massivning elementlari hisoblanadi.

Massiv elementlariga murojaat kvadrat qavslar ichida berilgan indekslar orqalai amalga oshiriladi. Massiv elementlarini tartiblash noldan boshlanadi:

```
var Massqnew Array("Bir", "Ikki", "Uch");
document.write(Mass[0]); // "Bir"
Mass[3] q 4; // Massivning yangi elementini yaratish
document.write(Mass.join(", ")); // "Bir, Ikki, Uch, 4"
length massiv elementlari sonini qaytaradi:
var Mass q ["Bir", "Ikki", "Uch"];
document.write(Mass.length Q "<br>"); // 3
for (var iq0, cqMass.length; i<c; iQQ){
document.write(Mass[i] Q "<br>"); // Bir, Ikki, Uch
}
```

Massivlar bilan ishlashning qulay metodlari:

- **push(<elementlar ro'yxati>)** massiv elementlariga ko'rsatilgan elementlar ro'yxatini qo'shadi. Elementlar massiv oxiriga qo'shiladi. Metod massivning yangi uzunligini qaytaradi:

```
var Mass q [ "Bir", "Ikki", "Uch"];
document.write(Mass.push("To'rtinchi", "Beshinchi")); //5
document.write(Mass.join(", ")); // "Bir, Ikki, Uch, To'rtinchi, Beshinchi"
```
- **unshift(<elementlar ro'yxati>)** massiv elementlariga ko'rsatilgan ro'yxat elementlarini qo'shish. Elementlar massivning boshidan qo'shiladi:

```
var Mass q["Bir", "Ikki", "Uch"];
Mass.unshift("To'rtinchi", "Beshinchi");
document.write(Mass.join(", ")); // "To'rtinchi, Beshinchi, Bir, Ikki, Uch"
```
- **concat(<elementlar ro'yxati>)** joriy massiv va elementlar ro'yxatini birlashtirib natijani boshqa massiv sifatida qaytaradi. Bunda joriy massivga elementlar ro'yxati qo'shilmaydi:

```
var Mass q [ "Bir", "Ikki", "Uch" ];
var Mass2 q []; // Bo'sh massiv
Mass2 q Mass.concat("To'rtinchi", "Beshinchi");
document.write(Mass.join(", ")); // "Bir, Ikki, Uch"
document.write(Mass2.join(", ")); // "Bir, Ikki, Uch, To'rtinchi, Beshinchi"
```
- **join(<bo'luvchi>)** massivning barcha elementlarini bo'luvchi orqali ajratilgan qator sifatida qaytaradi:

```
var Massq["Bir", "Ikki", "Uch"];
var StrqMass.join("-");
document.write(Str); // "Bir-Ikki-Uch"
```
- **shift()** massivning birinchi elementini o'chiradi va uni qaytaradi:

```
var Mass q ["Bir", "ikki", "Uch"];
document.write(Mass.shift()); // "Bir"
document.write(Mass.join(", ")); // "Ikki, Uch"
```

- `pop()` massivning oxirgi elementini o'chiradi va uni qaytaradi:
`var Mass q ["Bir", "Ikki", "Uch" ];`
`document.write(Mass.pop()); //"Uch"`
`document.write(Mass.join(", ")); //"Bir, Ikki"`
- `sort()` (saralash funksiyasi) massivni saralaydi. Agar fuksiya ko'rsatilmasa, odatiy saralash (sonlar o'sish, belgilar alfavit tartibida) amalga oshiriladi:
`var Mass q ["Bir", "Ikki", "Uch"];`
`Mass.sort();`
`document.write(Mass.join(", ")); //"Bir, Ikki, Uch" (alfavit tartibida, ya'ni b,i,u)`

Agar standart saralanish tartibini o'zgartirish kerak bo'lsa, buni funksiya yordamida amalga oshirish mumkin. Funksiya ikkita parametрни qabul qiladi va qaytarishi kerak:

- 1 — agar birinchi kata ikkinchidan bo'lsa;
- -1 — agar ikkinchi kata birinchidan bo'lsa;
- 0 — agar elementlar teng bo'lsa.

Masalan, standart saralash simvol registriga bog'liq:

```
var Mass q ["alisher", "Erkin", "Akmal", "elbek"];
Mass.sort();
document.write(Mass.join(", ")); //"Akmal, Erkin, alisher, elbek"
```

Natijada biz xato saralshga erishdik. Endi funksiya yordamida registrlarga bog'liq bo'lmagan saralashni ko'rib chiqamiz (4.6.1-misol).



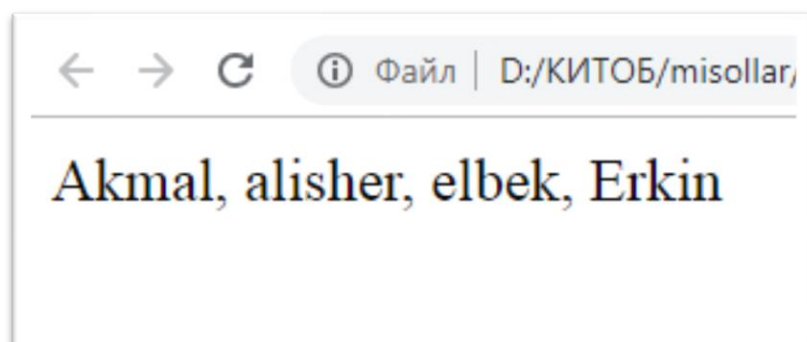
4.5.2-misol. Registrga bog'liq bo'lmagan saralash

```
function f_sort(Str1, Str2) {
var Str1_1qStr1.toLowerCase();//Quyida registrga o'tkazish
var Str2_1qStr2.toLowerCase();//Quyida registrga o'tkazish
if (Str1_1>Str2_1) return 1;
if (Str1_1<Str2_1) return -1;
return 0;
}

var Massq["alisher", "Erkin", "Akmal", "elbek"];

Mass.sort(f_sort);//Funksiya nomi qavssiz ko'rsatiladi

document.write(Mass.join(", ")); // "Akmal, alisher, elbek, Erkin"
```



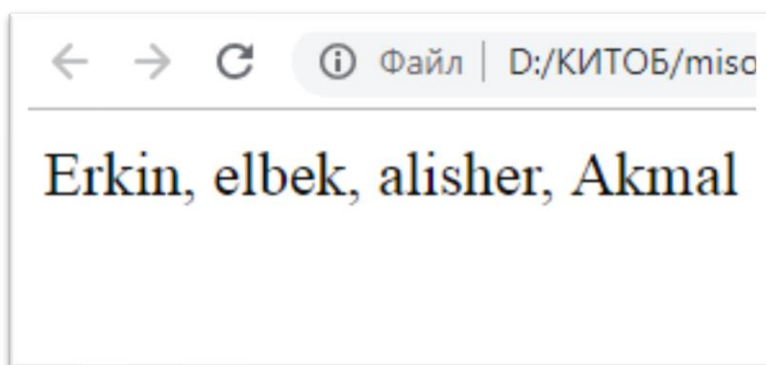
4.5.2-rasm

Ikkita ozgaruvchi bitta registra o'tkaziladi, keyin standart taqqoslash bajariladi. Teskari tartibda tartiblash quyidagicha bo'ladi:



4.5.3-misol. Registrga bo'g'liq bo'lmagan teskari tartiblash

```
function f_sort(Str1, Str2) { // Registrga bo'g'liq bo'lmagan teskari tartiblash
  var Str1_1 q Str1.toLowerCase(); // Quyi registrga o'tkazish
  var Str2_1 q Str2.toLowerCase(); // Quyi registrga o'tkazish
  if (Str1_1 > Str2_1) return -1; if (Str1_1 < Str2_1) return 1; return 0; }
var Mass q ["alisher", "Erkin", "Akmal", "elbek"];
Mass.sort(f_sort);
document.write(Mass.join(", ")); // "Erkin, elbek, alisher, Akmal"
```



4.5.3-rasm

- reverse() massivni teskarilash. Berilgan massiv elementlari o'rinlari teskarisiga almashadi:


```
var Mass q ["Bir", "Ikki", "Uch"]; Mass.reverse();
document.write(Mass.join(", ")); // "Uch, Ikki, Bir"
```
- slice(<boshlanish>, [<Ohir>]) massivning <boshlanish>dan boshlab <Ohir> gacha bo'lgan indeksdagi elementlarini qaytaradi va bunda <Ohir>gi element kirmaydi. Agar ikkinchi parameter ko'rsatilmasa, massivning ohirgi elementigacha olinadi:


```
var Mass1q[1, 2, 3, 4, 5 ];
var Mass2qMass1.slice(1, 4);
```

```

window.alert(Mass2.join(",")); //"2, 3, 4"
var Mass3 q Mass1.slice(2);
window.alert(Mass3.join(",")); //"3, 4, 5"

```

- splice(<boshlanish>, <Miqdor>, [<Qiymatlar ro'yxati>]) massiv elementlarini o'chirish, almashtirish yoki joylashtirishga imkon beradi. Qaytariladigan massiv o'chirilgan elementlardan iborat bo'ladi:

```

var Mass1 q [1, 2, 3, 4, 5];
var Mass2 q Mass1.splice(2, 2);
window.alert(Mass1.join(",")); //"1, 2, 5"
window.alert(Mass2.join(",")); // "3, 4"
var Mass3 q Mass1.splice(1, 1, 7, 8, 9);
window.alert(Mass1.join(",")); //"1,7,8,9,5" window.alert(Mass3.join(","));
// "2"
var Mass4 q Mass1.splice(1, 0, 2, 3, 4);
window.alert(Mass1.join(",")); //"1,2,3,4,7,8, 9, 5"
window.alert(Mass4.join(",")); //Bo'sh massiv

```

- toString() va valueOf() massivni qatorga o'tkazish. Massiv elementlari probelsiz vergul bilan ajratiladi:

```

var Mass q ["Bir", "Ikki", "Uch"];
document.write(Mass.toString()); //"Bir,Ikki,Uch"

```

Ko'p o'lchamli massiv. Ko'p o'lchamli massivni yaratish usullarini sanab chiqamiz:

```

var Mass q new Array(new Array("Bir", "Ikki", "Uch"), new Array ("To'rt",
"Besh", "Olti"));
document.write(Mass[0][1]); // "Ikki"
var Mass2q[["Bir", "Ikki", "Uch"],["To'rt", "Besh", "Olti"]];
document.write(Mass2[1][1]); // "Besh"
yoki element bo'yicha:
var Mass q new Array();
Mass[0] q new Array();
Mass[1] q new Array();
Mass[0][0] q "Bir";
Mass[0][1] q "Ikki";
Mass[0][2] q "Uch";
Mass[1][0] q "To'rt";
Mass[1][1] q "Besh";
Mass[1][2] q "Olti";
document.write(Mass[1][2]); // "Olti"
var Mass2q[];
Mass2[0] q [];
Mass2[1] q [];
Mass2[0][0] q "Bir";

```

```

Mass2[0][1] q "Ikki";
Mass2[0][2] q "Uch";
Mass2[1][0] q "To'rt";
Mass2[1][1] q "Besh";
Mass2[1][2] q "Olti";
document.write(Mass2[0][0]); // "Bir"

```

Ko'p o'lchamli massiv elementlariga murojaat ikki indeks orqali amalga oshiriladi:

```
var Str q Mass[1][2];
```

Assosiativ massiv. Assosiativ massivlarni saralash

Oddiy va assosiativ massiv asosiy farqi uning elementlariga murojaat qilishdadir. Assosiativ massivda elementlarga murojaat sonlar orqali emas qatorlar orqali amalga oshiriladi.

```

var Mass q new Array();
Mass["Bir"] q 1;
var Mass q new Array();
Mass["Bir"] q 1;
Mass["Ikki"] q 2;
Mass["Uch"] q 3;
document.write(Mass["Bir"]); // 1

```

Massivning barcha elementlarini qanday chop etiladi? Array sinfidagi hech bir metod birdaniga chop etish imkoniyatini bermaydi. O'z vaqtida, length xususiyati ham ishlamaydi. Buni standart sikl operatori for bilan ham amalga oshirib bo'lmaydi.

Bu maqsad uchun maxsus sikl for...in ishlatiladi. U quyidagi formatda beriladi:

```
for (<O'zgaruvchi> in <Sinf nusxasi>) { <Sikl tanasi> }
```

for...in siklida har bir iteratsiyada <O'zgaruvchi> ga assosiativ massiv elementlari indekslarini navbat bilan o'zlashtirirladi:

```

var Mass q new Array();
Mass["Bir"] q 1;
Mass["Ikki"] q 2;
Mass["Uch"] q 3;
for (var Name in Mass) { // Name o'zgaruvchisi har bir iteratsiyada
// assosiativ massiv qator indeksini o'zlashtiradi
document.write(NameQ"q "QMass[Name]Q "<br>");
}

```

Natijada biz quyidagi natijani olamiz:

Bir q 1

Ikki q 2

Uch q 3

Math sinfi. Matematik funksiyalardan foydalanish.

Math sinfi o'z ichiga matematik o'zgarmaslar va funksiyalarni oladi. Uning foydalanilishida sinf nusxasini yaratsihni talab qilinmaydi.

Hususiyat:

- E — e, natural logarifm asosi;
 - LN2 — natural logarifm 2;
 - LN10 — natural logarifm 10;
 - LOG2E — 2 asosga ko'ra logarifm e;
 - LOG10E — 10 asosga ko'ra logarifm e;
 - PI — Pi:
 - document.write(Math.PI); //3.141592653589793
 - SQRT1_2 — 0,5 kvadrat ildiz;
 - SQRT2 — 2 kvadrat ildiz.
- Uslublar:
- abs() — absolyut qiymat;
 - sin(), cos(), tan() — standart trigonometrik funksiyalar (sinus, kosinus, tangens). Qiymatlar radianda ko'rsatiladi;
 - asin(), acos(), atan() — teskari trigonometric funksiyalar (arksinus, arkkosinus, arktangens). Qiymatlar radianda qaytariladi;
 - exp() — eksponenta;
 - log() — natural logarifm;
 - pow(<Son>, <Daraja>) — <Son> <Daraja> ga ko'tariladi:
var x q 5;
document.write(Math.pow(x, 2)); // 25 (5 ning kvadrat darajasi)
 - sqrt() — kvadrat ildiz:
var x q 25;
document.write(Math.sqrt(x)); // 5 (25 dan kvadrat ildiz)
 - round() — sonni eng yaqinb butun songacha yaxlitlash. Agar verguldan keying birinchi son 0 va 4 oralig'ida bo'lsa, kami bilan yaxlitlanadi aksincha ortig'i bilan:
var x q 2.499; var y q 2.5;
document.write(Math.round(x)); // 2
document.write(Math.round(y)); // 3
 - ceil() — sonni eng yaqin katta butun songa yaxlirlash:
var x q 2.499; var y q 2.5;
document.write(Math.ceil(x)); // 3
document.write(Math.ceil(y)); // 3
 - floor() — eng yaqin kichik butun songa yaxlitlash:
var x q 2.499; var y q 2.6;
document.write(Math.floor(x)); //2

```
document.write(Math.floor(y)); // 2
```

- max(<vergul bilan ajratilgan sonlar ro'yxati>)- ro'yxatdagi qiymatlarning eng kattasi:

```
document.write(Math.max(3, 10, 6)); // 10
```

- min (<vergul bilan ajratilgan sonlar ro'yxati>) – ro'yxatdagi qiymatlarning eng kichkinasi:

```
document.write(Math.min(3, 10, 6)); // 3
```

- random() —0 dan 1 gacha tasodifiy son:

```
document.write(Math.random()); // masalan, 0.9778613566886634
```

Agar 0 dan 9 gacha tasodifiy butun sonni olish kerak bo'lsa, random() metodidan olingan natijani 9.9999 ga ko'paytirish, keyin floor() metodi yordamida yaxlitlash mumkin bo'ladi:

```
var x q Math.floor(Math.random()*9.9999);
```

```
document.write(x);
```

Web sahifani bir necha bor qayta yuklab tekshirib ko'rish mumkin. Tasodifiy sonlar 0 va 9 oralig'ida berilmoqda. Bu nima uchun ishlatiladi? Masalan, bizda to'rtta 468x60 o'lchamdagi bannerni tasodifiy ko'rsatish talab etilsin.

```
var x q Math.floor(Math.random()*3.9999);
```

```
document.write('<img srcq"banner' Q x Q '.gif" widthq"480" heightq"60">');
```

banner0.gif, banner1.gif, banner2.gif va banner3.gif nomli to'rt banner bir papkada scripda foydalanish uchun joylashgan bo'lishi kerak.

Turli nomdagi bannerli fayllarni massivga olish va ulardan massivnicng tasodifiy elementlari sifatida foydalanish mumkin:

```
var Mass q ["banner-red.gif", "banner-blue.jpeg", "banner-gray.gif", "banner-white.png" ];
```

```
var x q Math.floor(Math.random()*3.9999);
```

```
document.write('<img srcq"' Q Mass[x]);
```

```
document.write('widthq"480" heightq"60">');
```

Date sinfi. Joriy sana va vaqt. Web brauzer oynasida sava va vaqtni chop etish.

Date sinfi vaqt va sana bilan ishlash imkonini beradi. Sing nusxasi quyidagicha yaratiladi:

```
<Sinf nusxasi> q new Date();
```

```
<Sinf nusxasi> q new Date(<Milisekund miqdori>);
```

```
<Sinf nusxasi> q new Date(<Yil>, <Oy>, <Kun>, <Soat>, <Minutlar>, <Sekundlar>, <Milisekundlar>);
```

Sinf quyidagi metodlarni qo'llaydi:

- toString() sanani qatorga o'tkazadi va uni qaytaradi:

- ```

var d q new Date();
document.write(d.toString());
// Opera da: Fri,30 Oct 2009 01:07:17 GMTQ0300
// Firefox da: Fri Oct 30 2009 01:07:17 GMTQ0300
// IE da: Fri Oct 30 01:07:17 UTCQ0300 2009

```
- toLocaleString() tizimda o'rnatilgan sanani qatorga o'tkazadi va uni qaytaradi:

```

var d q new Date();
document.write(d.toLocaleString());
// Opera da: 30.10.2009 1:11:27
// Firefox da: 30 Oktyabr 2009 g. 1:11:27
// IE da: 30 oktyabrya 2009 g. 1:11:27

```
  - valueOf()– 01.01.1970 00:00:00 dan boshlab o'tgan vaqtdagi milisekundlar miqdorini aniqlash imkonini beradi:

```

var d q new Date();
document.write(d.valueOf()); // 1256854444062

```
  - getDate() oy kuni tartibini qaytaradi (1 dan 31 gacha):

```

var d q new Date(); document.write(d.getDate()); // 30

```
  - getDay() hafta kunini tartibini aniqlatydi (yakshanba- 0 dan 6 -shanba gacha):

```

var Mass q["yakshanba", "dushanba", "seshanba",
"chorshanba", "payshanba", "juma", "shanba"];
var d q new Date();
document.write(Mass[d.getDay()]); // juma

```
  - getMonth() oy tartibini qaytaradi (0- yanvar dan 11-dekabr gacha):

```

var Mass q["yanvar", "fevral", "mart", "aprel", "may",
"iyun", "iyul", "avgust", "sentabr", "oktabr", "nayabr", "dekabr"];
var d q new Date();
document.write(Mass[d.getMonth()]); // oktabr

```

Joriy oy tartibini aniqlash uchun funksiya qaytargan qiymatni 1 ga oshirish kerak:

```

var d q new Date();
var Month q d.getMonth() Q 1;
document.write(Month); // 10

```
  - getFullYear() yilni aniqlash imkonini beradi:

```

var d q new Date();
document.write(d.getFullYear()); // 2009

```
  - getHours() soatni aniqlash imkonini beradi (0 dan 23 gacha):

```

var d q new Date(); document.write(d.getHours()); // 1

```
  - getMinutes() minutni aniqlash imkonini beradi (0 dan 59 gacha):

```

var d q new Date();
document.write(d.getMinutes()); // 23

```
  - getSeconds() sekundni aniqlash imkonini beradi (0 dan 59 gacha):

- ```
var d q new Date();
document.write(d.getSeconds()); // 20
```
- getMilliseconds() milisekundni qaytaradi (0 dan 999 gacha):
- ```
var d q new Date();
document.write(d.getMilliseconds()); // 156
```
- getTime()– 01.01.1970 00:00:00 dan boshlab o'tgan milisekundlar miqdorini aniqlash imkonini beradi:
- ```
var d q new Date();
document.write(d.getTime()); // 1256855182843
```



4.5.4-misol. Joriy sana va vaqt

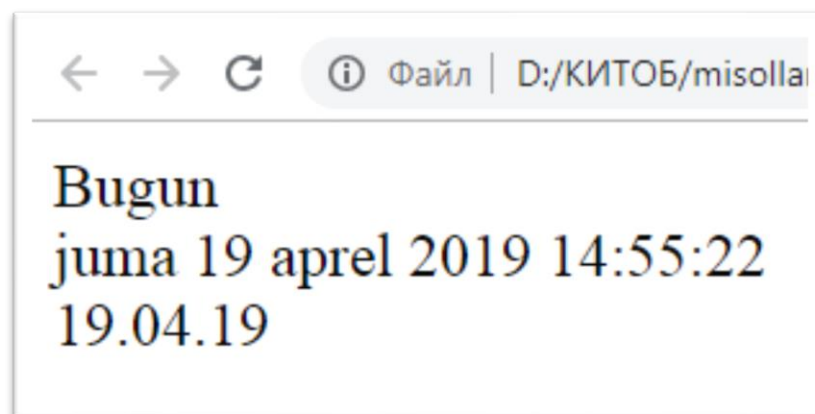
```
<html> <head>
<title>Joriy vaqt va sana</title>
<meta http-equiv="Content-Type" content="text/html;
charset q windows-1251">
<script type="text/javascript">
function fDate(Str) {
Str Qq ""; // sanani qatorga o'tkazish
if (Str.lengthq1) return ("0" Q Str);
else return Str;}
function fYear(Year) {
Year Qq ""; // sananani qatorga o'tkazish
return Year.substr(2);
}
</script>
</head>
<body>
<script type="text/javascript">
var d q new Date();
var msg;
var Dayq["yakshanba", "dushanba", "seshanba", "chorshanba",
"payshanba", "juma", "shanba"];
var Monthq["yanvar", "fevral", "mart", "aprel", "may", "iyun", "iyul", "avgust",
"sentyabr", "oktyabr", "noyabr", "dekabr"];
msg q "Bugun <br>" Q Day[d.getDay()] Q " ";
msg Qq d.getDate() Q " ";
msg Qq Month[d.getMonth()] Q " ";
msg Qq d.getFullYear() Q " ";
msg Qq fDate(d.getHours()) Q ":";
msg Qq fDate(d.getMinutes()) Q ":";
msg Qq fDate(d.getSeconds()) Q "<br>";
msg Qq fDate(d.getDate()) Q ".";
msg Qq fDate(d.getMonth()Q1)Q ".";
```

```

msg Qq fYear(d.getFullYear());
document.write(msg);
</script>
</body>
</html>

```

Web brauzer oynasida quyidagi paydo bo'ladi



4.5.4-rasm

Misolda biz ikkita funksiya yartib undan foydalandik:

- fDate(str) — agar parameter bitta raqamdan iborat bo'lsa, funksiya uning oldiga 0 qo'yib qatorni qaytaradi;
- f_Year(Year)— funksiya yilning ohirgi ikki raqamini qaytaradi.

Function sinfi

Function sinfi funksiya dan sinf nusxasi sifatida foydalanish imkonini beradi. Bu quyidagicha amalga oshiriladi:

<Funksiya nomi> q new Function(<Parametr1>, ... , <Parametrn>, <Funksiya tanasi>);

Masalan, ikki son yig'indisini xisoblasin

```
function f_Sum(x, y) {return x Q y;}
```

quyidagicha yozish mumkin:

```
var f_Sumqnew Function ("x", "y", "return xQy");
```

Funksiya tanasini qator sifatida ko'rsatish juda noqulay. Bu holdan hech kim foydalanmaydi.

Bu o'rinda anonim funksiya qo'llaniladi:

```
var f_Sum q function(x, y){ return x Q y; };
```

Funksiyani chaqirish oldingi kabi quyidagicha:

```
document.write(f_Sum(5, 6)); // 11
```

Anonim funktsiyalardan foydalanganda, global o'zgaruvchan funktsiyadagi funktsiyani belgilaganingizda, bu o'zgaruvchiga havola saqlanadi, uning qiymati emas:

```
var x q 5;  
var f_Sum q function() {  
  return x; // havola saqlanadi, x qiymati emas };  
document.write(f_Sum()); // 5  
x q 10; // Qiymatni o'zgartirdi  
document.write(f_Sum()); // 10 ga teng, 5 emas
```



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. JavaScriptda sinf qanday yaratiladi?
2. Metod nima?
3. JavaScriptda Global sinf haqida nimalarni bilasiz?
4. String sinfi haqida nimalarni bilasiz?
5. Array sinfi qanday maqsadlarda ishlatiladi?
6. **Number sinfi haqida nimalarni bilasiz?**
7. **Date sinfidan foydalanish qanday?**

4.6-§.JavaScriptda xodisalar. Interaktiv formalar ishlab chiqish.



O`quv modullari

xodisa, sichqoncha xodisasi, klaviatura xodisasi, forma xodisasi, hujjatlar bilan ishlash xodisasi, event obyeki

Foydalanuvchilarning web saxifadagi harakatlarida hodisalar sodir bo`ladi. **Hodisa** –bu tizim yoki foydalanuvchi tomonidan qandaydir harakatlarning bajarilishida paydo bo`ladigan jarayondir.

Asosiy tushunchalar.

Hodisa elementga tekkandan so`ng, sichqonchaning harakatida, klaviaturaning tugmalarini bosilganda, oynaning o`lchamlari o`zgarganda web saxifadagi amallar bajarilganda paydo bo`ladi.

Web-sahifaning yoki uning qaysidir elementiga tegishli hodisalarni bilish uchun ushbu hodisani boshqarish funktsiyasini yozishingiz mumkin bo`ladi. Qanday hodisa web saxifada qanaqa o`zgarishlar bajaradishini bilishimiz mumkin. Masalan, ayni bir formadan ma`lumot jo`natganda onsubmit hodisasi paydo bo`ladi.

Bu hodisa bajarilishi boshlanishida malumotlarni tekshirish, foydalanuvchilar kirishi va agar ular muvofiq bo'lmasa malumotlarni jo'natilishi to'xtatilishini nazorat qilish mumkin.

Hamma hodisalar **on** old qo'shimchasi bilan boshlanadi

Sichqoncha xodisasi

Sichqoncha xodisalarining asosiylarini ko'rib chiqmiz:

- ✓ onmousedown — Sichqoncha tugmasini web saxifaning yoki saxifaning elementi ustida bosilganda
- ✓ onmouseup — Sichqoncha tugmasini bosib, qo'yib yuborilishida;
- ✓ onclick — Sichqoncha tugmasini web saxifaning elementi yoki web saxifaning o'zida bosilganda;
- ✓ ondblclick — Sichqoncha tugmasini ikki marta bosilganda;
- ✓ onmousemove — Sichqoncha tugmasining har qanday harakatida;
- ✓ onmouseover — Sichqoncha ko'rsatkichining web saxifaning elementidan chiqib ketishida;
- ✓ onmouseout — Web saxifaning elementiga sichqoncha ko'rsatkichi borganda;
- ✓ onselectstart — Matnning belgilanisho boshida;
- ✓ onselect — Elementning elgilanishida;
- ✓ oncontextmenu — Sichqonchaning o'ng tugmasini bosib kontekst menyuni chiqarilishida uchun.

Klaviatura xodisasi

Klaviatura hodisasining asosiylarinin ko'rib chiqamiz:

- ✓ onkeydown — Klaviatura tugmalari bosilganda;
- ✓ onkeypress — onkeydown hodisasiga o'xshash, faqat Unicode kodirovkasi belgisi qiymatini qaytaradi. Har doim foydalanuvchi klavishni qo'yib yubormasdan oldin chiqadi;
- ✓ onkeyup — Klaviatura tugmalarini erta qo'yib yuborilganda;
- ✓ onhelp — <F1> tugmasini bosilganda;

Hujjatlar bilan ishlash hodisasi

Hujjatlar ning asosiy hodisasini sanab o'tamiz:

- ✓ onload — Web sahifa yuklangach;
- ✓ onscroll — hujjat, oyna yoki freymda o'tkazilganda (varaqlanganda);
- ✓ onresize — Oynaning o'lchamlari o'zgarganda;
- ✓ onbeforeunload — Hujjatning yuklanishidan keyin;

- ✓ onunload — bevosita hujjatning yulklangandan keyin chiqadi. onbeforeunload hodisasidan so`ng sodir bo`ladi;
- ✓ onbeforeprint — Hujjatni birinchi marta ochganda yoki har safar ochganda;
- ✓ onafterprint — Hujjatni birinchi ocilishida yoki hujjatni ochgandan so`ng

Forma hodisasi

Forma hodisasining asosiylarini ko`ramiz:

- ✓ onsubmit — Ayni bir formani jo`natganda;
- ✓ onreset — Formani tozalanganda;
- ✓ onblur — Formaning focus elementini yo`qotganda;
- ✓ onchange — matnli maydon ma`lumotlarini o`zgarishida va formaning boshqa elementiga fokusni ko`chirishda va forma ma`lumotlarini yuborishda; (onblur hodisasidan oldin boshlanadi);
- ✓ onfocus — Formaning focus elementini qabul qilganda;

Xodisalar bajarilish ketma-ketligi

Hodisa ketma ketlikda paydo boladi, masalan, web saxifa elementida sichqoncha bosilganda hodisalar ketma-ketligi quyidagicha bo`ladi:

onmousedown

onmouseup

onclick

Ketma-ketlik sichqoncha ikki mara bosilganda quyidagicha ;

onmousedown

onmouseup

onclick

ondblclick

Bu shuni bildiradiki ondblclick hodisasi onclick hodisasidan keyin vujudga keladi.

Klaviatura tugmalari bosilganda hodisalar ketma-ketligi quyidagicha bo`ladi:

onkeydown

onkeypress

onkeyup

Quyidagi misolni ko`rib chiqamiz (Misol 3.34).



4.6.1. Hodisa ketma ketligi

```
<!DOCTYPE HTML>
```

```
<html> <head>
```

```
<title> Hodisa ketma keyligi</title>
```

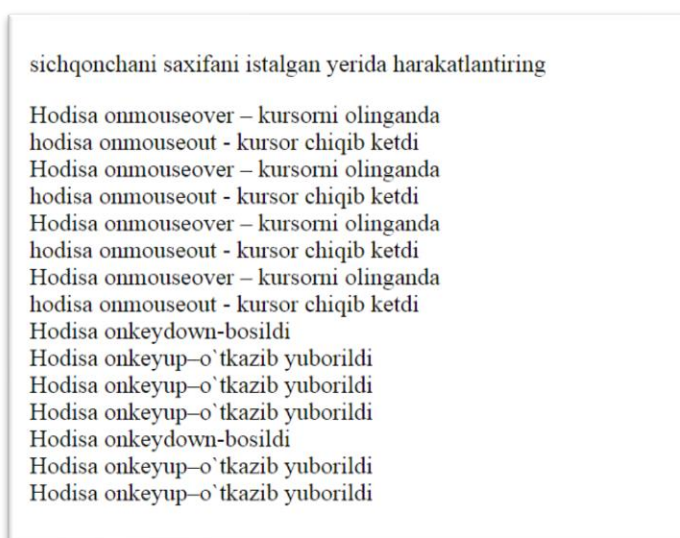
```
<meta http-equiv q "Content-Type" content q "text/html; charsetutf-8">
```

```
<script typeq"text/javascript">
```

```

function f_print(Str)
{
var div1 q document.getElementById("div1");
div1.innerHTML Qq Str Q "<br>";
}
</script> </head>
<body onload q"f_print('Hodisa onload-saxifa yuklandi');"
onmousedownq"f_print('hodisa onmousedown –.bosildi');"
onmouseupq"f_print('hodisa onmouseup–qo`yib yuborildi');"
onclickq"f_print('Hodisa onclick - bosildi');"
onkeydownq"f_print('Hodisa onkeydown-bosildi');"
onkeypressq"f_print('Hodisa onkeypress - tugma bosildi');"
onkeyupq"f_print('Hodisa onkeyup–o`tkazib yuborildi');">
<p onmouseoverq"f_print('Hodisa onmouseover – kursorni olinganda');"
onmouseoutq"f_print('hodisa onmouseout - kursor chiqib ketdi');"> sichqonchani
saxifani istalgan yerida harakatlantiring </p><p></p>
<div idq"div1"> </div>
</body>
</html>

```



4.6.1-rasm

Yuqoridagi misol hodisalar ketma-ketligii yaqqol ko`rish imkonini beradi. **onload** hodisasi yuklanishdan keyin yuzaga keladi. Sichqonchani tugmachasini qo`yib yubormasdan, oynaning istalgan qismida harakatlantirsak onmousedown

hodisasi yuzaga keladi. Agar tugmachani qo'yib yuborsak darhol ikki hodisa onmouseover va onclick paydo bo'ladi. Agar klaviaturaning istalgan tugmalarini bosib tursak ikki hodisa onkeydown va onkeypress hodisalari nomoyon bo'ladi. Bundan tashqari, agar klavishni bosib turishni davom ettirsak onkeypress hodisasi qayta takrorlanadi. Agar qo'yib yuborsak onkeyup hodisasi yuzaga keladi. Sichqoncha kursor yo'nalishini "Sichqonchani saxifani istalgan yerida harakatlantiring" yozuvida harakatlantirilsa onmouseover hodisasi paydo bo'ladi. Agar kursorni yozuvdan olsak onmouseout hodisasi paydo bo'ladi.

***ESLATMA!** Eslatib o'tamiz hodisalarning ushbu ketma –ketligi Internet Explorer Web brauzerida ishlaydi. Boshqa web brauzerlarda boshqacha kuzatilishi mumkin. Hamma misollar Microsoft Internet Explorer uchun yozilgan va bundan keyin ayni ushbu web brauzer uchun o'rganamiz.*

Xodisalarning paydo bo'lishi

“Paydo bo'luvchi” hodisasi nima o'zi. Keling buni misolda korib chiqamiz



4.62-misol. “Paydo bo'luvchi” hodisa

```
<!DOCTYPE HTML>
<html> <head>
<title> “Paydo bo'luvchi” hodisa </title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script type="text/javascript">
function f_print(Str) {
var div1 = document.getElementById("div1");
div1.innerHTML += Str + "<br>";
}
</script>
</head>
<body onclick="f_print('Hodisa onclick - Hujjat');">
<p onclick="f_print('Hodisa onclick – Abzas');"> Sichqonchani harakatlantiring
<span style="color:red" onclick="f_print('Hodisa onclick - SPAN');"> bu
yerda</span> </p>
<div id="div1"></div>
</body>
</html>
```

Sichqonchani harakatlantiring **bu yerda**

Hodisa onclick - SPAN
Hodisa onclick – Abzas
Hodisa onclick - Hujjat
Hodisa onclick – Abzas
Hodisa onclick - Hujjat
Hodisa onclick – Abzas
Hodisa onclick - Hujjat
Hodisa onclick – Abzas
Hodisa onclick - Hujjat
Hodisa onclick – Abzas
Hodisa onclick - Hujjat
Hodisa onclick – Abzas
Hodisa onclick - Hujjat
Hodisa onclick – Abzas
Hodisa onclick - SPAN
Hodisa onclick – Abzas
Hodisa onclick - Hujjat

4.6.2-rasm

Bu misolda biz onclick hodisasini saxifaning uch elementi—hujjat tanasi, tegi va satr boshida ishlashini ko'rdik. Sichqoncani chap tugmachasini “shu yer” yozuvi ustida bosib ko'ramiz. Natijada, bitta onclick hodisasi o'rniga, hodisalar ketma-ketligini ko'rishimiz mumkin.

Hodisa onclick - SPAN

Hodisa onclick -Abzas

Hodisa onclick - Hujjat

Boshqacha aytganda onclick hodisasi navbatdagi elementga o'tkaziladi. tegi uchun ajdod element bu abzas hisoblanadi. Absaz uchun esa ajdod element hujjat tanasi hisoblanadi. Bunday hodisalar paydo bo'luvchi hodisalar deb ataladi.

Ba'zida paydo bo'luvchi hodisani to'xtatish kerak bo'ladi. Buning uchun event obyektining cancelBubble hususiyatini true ga o'rnatish kerak. Bundan tashqari bazi bir web brauzerlarda paydo bo'luvchi hodisalarni toxtatish uchun event obyektining stopPropagation() metodidan foydalanish mumkin. stopPropagation() metodi Internet Explorer brauzeri uchun ishlamasligini unutmang. Paydo bo'luvchi hodisasining to'xtatilishini (3.36- Misol) dagi misolda ko'rib chiqamiz.



4.6.3-misol. “Paydo bo'luvchi” hodisani to'xtatilishi

```
<!DOCTYPE HTML>
<html> <head>
<title> Paydo bo'luvchi hodisaning to'xtatilishi</title>
<meta http-equivq"Content-Type" contentq"text/html; charsetutf-8">
<script typeq"text/javascript">
function f_print(Str, e) {
var div1 q document.getElementById("div1");
div1.innerHTML Qq Str Q "<br>";
e q e||window.event;
if(e.stopPropagation) e.stopPropagation();
else e.cancelBubbleqtrue;
}
```

```

</script>
</head>
<body onclickq"f_print('Hodisa onclick-Hujjat', event);">
<p onclickq"f_print('Hodisa onclick –Absaz' , event);"> Sichqonchani
harakatlantiring
<span styleq"color: red" onclickq"f_print('Hodisa onclick - SPAN', event);">
Bu yerda</span>
</p>
<div idq"div1"></div>
</body>
</html>

```

“Bu yerda” yozuvi ustida sichqonchani harakatlantirish tugmasini bosib ko`ramiz. Bu yerda biz uchta hodisani o`rniga faqat bittani ko`ramiz:

Sichqonchani harakatlantirish **Bu yerda**

Hodisa onclick - SPAN
 Hodisa onclick –Absaz
 Hodisa onclick –Absaz
 Hodisa onclick - SPAN
 Hodisa onclick –Absaz
 Hodisa onclick - SPAN
 Hodisa onclick –Absaz
 Hodisa onclick –Absaz

4.6.3-rasm

Odatiy harakat va ularni bekor qilish

Ko'pgina hodisalar uchun odatiy holatlar ya'ni web-brauzer hujjatda sodir bo'lgan voqealarga javob beradigan harakatlar belgilanadi.

Ko`plab hodisalar uchun odatiy holatda harakatlar belgilanadi, web brauzer hujjatdagi belgilangan hodisaga javob beradi. Masalan, gippersilkaga bosilganda ko`rsatilgan URL manziliga ishora qilingan bo`ladi, **JO`NATISH** ni bosganda belgilangan forma manzilga jo`natiladi.

Ba`zan odatiy harakatni to`xtatish kerak bo`ladi. Masalan, ayni bir hujjat jo`natishda kutuvchilarga tog`ri kelishini tekshirish kerak. Agar to`g`ri kelmasa jo`natishni to`xtatish kerak. Buning uchun false qiymatga mos keladi. False ga qaytganda harakatlarni bekor qilganda ishora harakati preventDefault metodidan, event obykti yoki returnValue xususiyatidan foydalanish mumkin. E-mail kiritilishini to`g`riligini tekshirish va gipersilka bo`yicha o`tish misoli berilgan.



4.6.4-misol. Gipperishorat harakatini to`xtatilishi

```

<!DOCTYPE HTML>
<html> <head>
<title> Odatiy holatni to`xtatish </title>
<meta http-equivq"Content-Type" contentq"text/html;
charsetqwindows-1251">

```

```

<script type="text/javascript">
function f_test() {
var p q /^[a-z0-9_g`.g`-]Q@([a-z0-9g`.]Qg`.)Q[a-z]{2,6}$/i;
// Email maydoni qiymatini olamiz
var email q document.forms[0].email.value;
if (p.test(email)) {
if (window.confirm("Ushbu formani jo`natasizmi?")){ return true;
// Jo`natamiz
}
else return false; // To`xtatamiz
}
else {
window.alert("E-mail tog`ri ko`rsatilmagan"); return false;
// To`xtatamiz
}
}
function f_event(e) { e q e || window.event;
if (e.preventDefault) e.preventDefault();
else e.returnValue q false;
window.alert("Ishoratga o`tilmaydi!");
}
</script>
</head>
<body>
<form action="file.php" method="GET" onsubmit="return f_test();" > <div>
E-mail:<br> <input type="text" name="email"><br>
<input type="submit" value="Jonatamiz"> </div> </form>
<p>
<a href="file.html" onclick="window.alert('Ishoratga o`tilmaydi!');
return false;"> Ishoratga o`tish uchun bosing</a><br><br> <a href="file.html"
onclick="f_event(event);"> Ishoratga o`tish uchun bosing </a>
</p>
</body>
</html>

```



E-mail:

[Ishoratga o`tish uchun bosing](#)
[Ishoratga o`tish uchun bosing](#)

4.6.4-rasm

Xodisalar qayta ishlanmasining yozilishi

Avvalgi misollardan ko`rinib turibdiki, hodisa ishlanmalaridan teglar sifatida foydalanish mumkin:

```
<span styleq"color: red" onclickq"f_print('Hodisa onclick - SPAN', event);"> bu yerda </span>
```

Lekin bu qayta ishlanmalarning yozishning yagona yo`li emas. Qayta ishlanmalarni for va event <script> tegi parametrlari yordamida ham yozish mumkin (Misol 3.38). Buning uchun web saxifaning elementi id parametriga ega bo`lishi kerak. id paremetri ko`pchilik teglarda uchraydi.

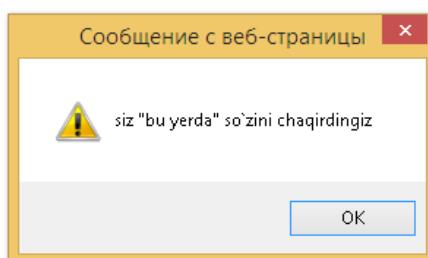


4.6.5-misol. Qayta ishlangan hodisaning yozilishi

```
<!--Faqat Internat exploderda ishlaydi !!! -->
```

```
<html> <head>
<title> Hodisani qayta ishlash</title>
<meta http-equivq"Content-Type" contentq"text/html; charsetqwindows-1251">
<script typeq"text/javascript" forq"txt" eventq"onclick">
window.alert('siz "bu yerda" so`zini chaqirdingiz'); </script>
</head>
<body>
Sichqonchoni harakatlantiring
<span styleq"color: red" idq"txt">bu yerda</span>
</p>
</body>
</html>
```

Sichqonchoni harakatlantiring **bu yerda**



4.6.5-rasm

Qayta ishlanmani funksiya ko`rsatkichi yordamida belgilash mumkin. Bekor qilish uchun funksiyaning nomi albatta qavslarsiz yozilishi kerak hamda atributlati to`liq yozilishi kerak



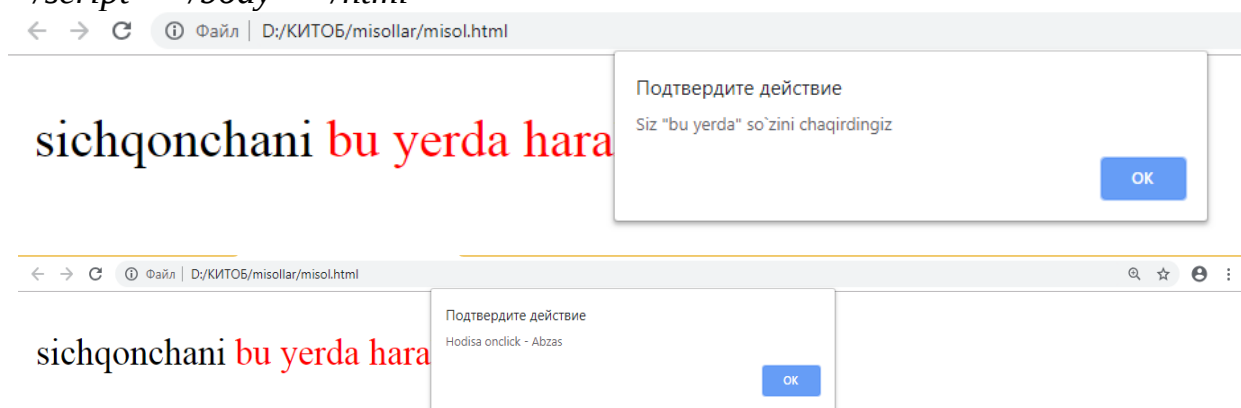
4.6.6-misol. Qayta ishlanma funksiya ko`rsatkichi yordamida yozilishi

```
<!DOCTYPE HTML >
<html> <head>
<title> Hodisa qayta ishlanmasi </title>
```

```

<meta charsetq"utf-8">
<script typeq"text/javascript">
function f_click(e) {
e q e || window.event;
window.alert('Siz "bu yerda" so`zini chaqirdingiz');
// this –bu elementga sisilka, chaqiruvchi hodisa this.innerHTML q "yangi matn";
// qalqigan hodisa to`xtatildi
if (e.stopPropagation) e.stopPropagation(); else e.cancelBubble q true; // IE
uchun
}
</script> </head> <body>
<p onclickq"window.alert('Hodisa onclick - Abzas');"> sichqonchani
<span styleq"color: red" idq"txt">bu yerda harakatlantiring</span>
</p>
<script typeq"text/javascript">
// Etibor qiling: Funksiya nomi qavslarsiz ko`rsatiladi !!!
document.getElementById("txt").onclick q f_click;
</script> </body> </html>

```



4.6.6-rasm

Shuning uchun qayta ishlanmani yashiringan (anonim) funksiya yordamida yozish mumkin (4.6.7-misol).



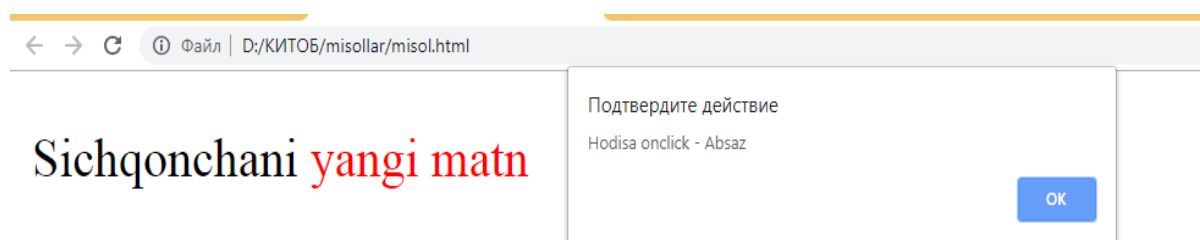
4.6.7-misol. Qayta ishlanma yashiringan funksiya yordamida yozilishi

```

<!DOCTYPE HTML>
<html> <head>
<title>hodisa qayta ishlanmasi</title>
<meta http-equivq"Content-Type" contentq"text/html; charsetqutf-8">
</head> <body>
<p onclickq"window.alert('Hodisa onclick - Absaz');"> Sichqonchani
<span styleq"color: red" idq"txt">Shu yerda harakatlantiring</span>
</p>
<script typeq"text/javascript">

```

```
document.getElementById("txt").onclick q function(e) { eqe||window.event;
window.alert(' Siz"bu yerda " so`zini chaqirdingiz');
// this - Bu sisilka chaqiruvchi hodisaning elementi
this.innerHTML q "yangi matn";
// qalquvchi hodisaning to`xtatilishi
if (e.stopPropagation) e.stopPropagation();
else e.cancelBubble q true;
}
</script>
</body>
</html>
```



4.6.7-rasm

Event obykti. Kursor koordinatasi va bosilgan tugma (klavish) kodini chiqarish. Klavishlar kombinasiyasi bosilishidagi habarini chiqarish.

Event obykti sodir bo`layotgan hodisalar haqida aniq malumot olishni va kutilmagan harakatlarni bajarish imkonini beradi. Event obykti faqat hodisalarni qayta ishlash uchun ruxsat etiladi. Keyingi hodisa paydo bo`lishi bilan undan oldingi hamma hususiyatlari yo`qoladi.

Event obykti quyidagi hususiyatlarga ega:

- ✓ srcElement — hodisaning manbasi hisoblangan elementga sisilka. DOM modeli Level 2 da target hususiyatidan foydalaniladi;
- ✓ currentTarget — DOM modeli Level 2 da qayta ishlanayotgan hodisa uchun elementga murojaatni qaytarad. Silkanayotgan elementga this kalit so`z currentTarget so`zining funksiyasi, target hususiyatiga tog`ri kelmasligi mumkin;
- ✓ type — hodisa tipdan iborat qator. Quyi registrda va on qo`shimchasisiz ishlaydi. Masalan, onclick hodisasida type hususiyati clickga teng;
- ✓ clientX i clientY — kliyent kordinata o`qlari bo`yicha hodisa kordinatasi (X va Y o`qi bo`yicha);
- ✓ screenX i screenY — nisbiy oyna bo`yicha hodisa kordinatsi (X va Y o`qi bo`yicha);

- ✓ offsetX i offsetY — nisbiy konteyner bo'yicha hodisa kordinatsi (X va Y o'qi bo'yicha);
- ✓ x va y — X va Y o'qi bo'yicha hodisaning kordinatsi. DOM Level 2 modelida bunday hususiyat yo'q;
- ✓ button — sichqoncha tugmasini bosilishini ko'rsatuvchi son. Son quyidagi qiymatlarga ega bo'lishi mumkin:
 - 0 — tugma bosilmagan;
 - 1 — sichqonchani chap tugmasi bosilsa;
 - 2 — sichqonchani o'ng tugmasi bosilsa;
 - 3 — sichqonchani chap va o'ng tugmasi bir vaqtda bosilsa;
 - 4 — o'rtadagi tugma bosilsa;
 DOM Level 2 modelida boshqa belgilar:
 - 0 — Sichqonchani chap` tugmasi bosilsa;
 - 1 — o'rtadagi tugma bosilsa;
 - 2 — sichqoncha chap tugmasi bosilsa;
- ✓ keyCode — klaviatura tugmasining bosilishi. Firefox web brauzerida onkeypress xodisasi qayta ishlanmasida odatiy tugma bosilishida keyCode hususiyati 0 ga ega bo'ladi, belgi kodi esa charCode hususiyati orqali aniqlanadi. Agar faqat funksional tugmalar bosilsa, bu holat almashadi charCode hususiyati 0 ga, belgi kodi esa keyCode orqali aniqlanadi;
- ✓ altKey — true, agar hodisa vaqtida <Alt> tugmasi bosilgan bo'lsa;
- ✓ altLeft — true, agar chap tugma <Alt> bosilgan bo'lsa va false, agar o'ng tugma bosilgan bo'lsa. DOM Level 2 modelida esa bu hususiyat yo'q;
- ✓ ctrlKey — true, <Ctrl> tugmasi bosilgan bolsa ;
- ✓ ctrlLeft — true, agar <Ctrl>ni chap tugmasi bosilgan bo'lsa va false, agar ongi bosilsa. DOM Level 2 modelida bu funksiya yo'q;
- ✓ shiftKey — true, agar <Shift> tugmasi bosilsa;
- ✓ shiftLeft — true, agar <Ctrl>ni chap tugmasi bosilgan bo'lsa va false, agar ongi bosilsa. DOM Level 2 modelida bu funksiya yo'q;
- ✓ cancelBubble hodisa obyekti ierarhiya bo'yicha nomoyon bo'lishini aniqlaydi;
- ✓ returnValue Sahifa elementi uchun ko'rsatuv amal bajarilishini bildiradi. Standart ishni to'xtatish uchun ushbu xususiyatni false deb belgilash kerak (4.6.8-misol) DOM 2-darajali modeli preventDefault() usulidan foydalanadi.



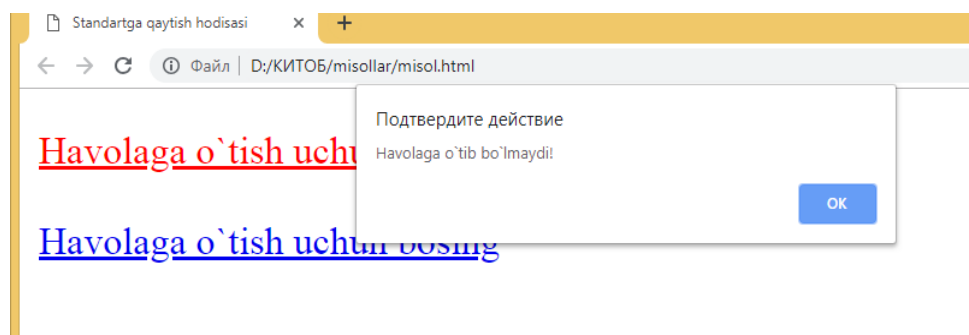
4.6.8-misol. Odatiy harakatni to'xtatish

<!DOCTYPE HTML>

```

<html> <head>
<title> Standartga qaytish hodisasi</title>
<meta charsetq"utf-8">
<script typeq"text/javascript">
function f_print(Str, e) {
window.alert(Str);
e q e || window.event;
if (e.preventDefault) e.preventDefault();
else e.returnValue q false;
}
</script>
</head>
<body>
<p>
<a hrefq"file.html"
onclickq"f_print('Havolaga o`tib bo`lmaydi!', event);">
Havolaga o`tish uchun bosing</a><br><br>
<a hrefq"file.html" onclickq"window.alert('Havolaga o`tib bo`lmaydi!'); return
false;"> Havolaga o`tish uchun bosing</a></p>
</body>
</html>

```



4.6.8-rasm

Bu misolda standartga qaytish hodisasiga ikkita misol ko`rsatilgan. Birinchi sisilkada returnValue hususiyatidan event obyektiga standartlashtirish ko`rsatilgan. Ikkinchi sisilkada false ga qaytish ko`rsatilgan

- fromElement — elementga sisilka, sichqonchani kursori harakatlantirishi uchun. DOM Level 2 da relatedTarget hususiyatidan foydalaniladi;
- toElement — sisilkaga element, sichqonchani kursorini harakatlantiruvchiga. DOM Level 2 modelida relatedTarget hususiyatidan foydalaniladi
- repeat — true, agar onkeypress hodisasi tugmachani bosib turish davomida qayta natija paydo bo`lishi. DOM Level 2 modelida bu hususiyat yo`q.
- propertyName — teg atributining nomi, saxifaning ko`rinish yoki hususiyati, o`zgargandagi. V DOM Level 2 modelida bu hususiyat yo`q.

event obyektining hususiyatlaridan foydalanishni 3.42. Misolda ko`ramiz



4.6.9-misol. Bosilgan tugma kodi va kursor kordinatasini aniqlash

```
<html>
<head>
<title>Bosilgan tugma kodi va kursor kordinatasini aniqlash</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script type="text/javascript">
function f_unload() {
event.returnValue = "Xujjatdan chiqasizmi?";
}
function f_body() {
switch (event.type) { case "mousemove":
var m1, m2;
m1 = event.clientX;
m2 = event.clientY;
var div1 = document.getElementById("div1");
div1.innerHTML = "clientX, clientY: " + m1 + " y - " + m2; m1 =
event.screenX; m2 = event.screenY;
var div2 = document.getElementById("div2");
div2.innerHTML = "screenX, screenY: " + m1 + " y - " + m2; m1 =
event.offsetX; m2 = event.offsetY;
var div3 = document.getElementById("div3");
var div4 = document.getElementById("div4");
var div5 = document.getElementById("div5");
```

```

div3.innerHTML q "offsetX, offsetY: x - " Q m1 Q " y - " Q m2; div4.innerHTML q
"x, y: x - " Q event.x Q " y - " Q event.y;
div5.innerHTMLq"Teg:"Qevent.srcElement.tagName; break; case "keypress":
var div6 q document.getElementById("div6");
div6.innerHTML q "kod tugma bilan - " Q event.keyCode;
if (event.ctrlLeft && event.keyCodeqq10) {
window.alert("Chap tugma hamda Ctrl Q Enter ni bosing");
}
if (!event.ctrlLeft && event.keyCodeqq10) {
window.alert("O`ng tugma hamda Ctrl Q Enter bosing");
}break; case "contextmenu":
event.returnValue q false;
break; case "selectstart":
event.returnValue q false;
break;
}
}
</script> </head>
<body onkeypressq"f_body();" onmousemoveq"f_body();"
onbeforeunloadq"f_unload();">
<p oncontextmenuq"f_body();">
Bu abzasda kontekst menyuni chiqarish mumkin emas</p>
<div idq"div5"></div>
<div idq"div1"></div>
<div idq"div2"></div>
<div idq"div3"></div>
<div idq"div4"></div>
<div idq"div6"> Klaviaturani kodini ko`rish uchun uni bosing</div>
<p>Habarni ko`rish uchun o`ng yoki chap tugmalarni Ctrl hamda Enter tugmalari
bilan birga bosing</p>
<p onselectstartq"f_body();">Bu abzasni boshqa abzaslardan bo`lib bolmaydi
</p>

```

```
<div><a href="file.html">
```

```
Silka paydo bo`lishi uchun bosing </a></div>
```

```
</body>
```

```
</html>
```

← → ↻ ⓘ 0айн | D:/KITO5/misollar/misol.html 🔍 ☆ ☰ :

Bu abzasda kontekst menyuni chiqarish mumkin emas

Teg:BODY

screenX, screenY: x - 245 y - 344

offsetX, offsetY: x - 115 y - 129

x, y: x - 122 y - 136

Klaviaturani kodini ko`rish uchun uni bosing

Habarni ko`rish uchun o`ng yoki chap tugmalarni Ctrl hamda Enter tugmalari bilan birga bosing

Bu abzasni boshqa abzaslardan bo`lib bolmaydi

[Silka paydo bo`lishi uchun bosing](#)

4.6.9-rasm

Shunday qilib, deyarli hamma hodisalar birgina `f_body()` funksiyasi bilan ishlaydi. `type` hususiyati yordamida event obykti va `switch` tanlash operatori qanday hodisa sodir bo`lganini aniqlash mumkin va unda ishlash mumkin. Sichqonchani o`ng tugmasini birinchi abzasga qo`yilsa kontekst menyu paydo bo`ladi. Sichqonchani kursorini joyini o`zgartirilsa kordinatni o`zgarishini kuzatamiz, klaviatura tugmalarini bossak uning kodi paydo bo`ladi. Masalan, bir vaqtning o`zida `<Ctrl> Q Left` va `<Enter>` tugmalarini bir bossak habar `<Ctrl> Q Right` va `<Enter>` tugmalarini bossak yana boshqa habar paydo bo`ladi.

Endi sisilkaga bosing yoki web – brayzerdan chiqisha harakat qilib ko`ring. Savolli oyna paydo bo`ladi. `onbeforeunload` hodisasi hujjat yuklanishidan oldin paydo bo`ladi standart dialogli ikki tugmali oyna paydo bo`lishini taminlaydi. Oyna ozida matnni saqlashi sifatini `returnValue` so`zi funksiyasi taminlaydi. `returnValue` sozi nafaqat `true` va `false` so`zlarini balki dialogli oynado joylashgan qator so`zlarni joylasishini ham taminlaydi. Biroq faqat bu `onbeforeunload` hodisai uchun to`g`ri.

```
event.returnValue="hujjatdan chiqishni hohlaysizmi?";
```

Bular faqatgina Microsoft Internet Explorer uchunligini eslatib o`tamiz. Masalan, Opera Web- brauzerida hususiyatlarni yarmi ishlamaydi. Boshqa Web-brauzerlar uchun o`zgartirishlar kiritish kerak bo`ladi..



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. Hodisa nima?
2. JavaScriptda sichqoncha hodisalaridan qanday foydalaniladi?
3. JavaScriptda klaviatura hodisalarini ayting.
4. JavaScriptda hujjat bilan ishlashning qanday hodislarini bilasiz?

5. Foma hodisalalari qanday qo'llaniladi?

4.7-§. Forma elementlari bilan ishlash



O`quv modullari

boshqaruv elementlari, forms to'plami, matnli maydon, qiymatlar ro'yxati, bayroqcha ,tugmalar

HTML ni o'rganishda biz forma elementlarini yaratishni o'rgangan edik. Bu mavzuda JavaScript yordamida forma elementlariga foydalanuvchi tomonidan kiritilgan ma'lumotlarni qayta ishlashni o'rganamiz. Kliyent tomonida ishlov berish noto'g'ri kiritilgan qiymatlarda forma ma'lumotlari yuborilishini bekor qilish orqali web-serverdagi yuklanishni kamaytiradi.

Boshqaruv elementlari

Boshlanishida yana bir marta formaning barcha elementlarini ko'rib chiqamiz:

- `<input type="text">` — matn kiritish maydoni;
- `<input type="password">` — parol kiritish maydoni;
- `<input type="file">` — Web-serverga fayl yuborish imkonini beradi;
- `<input type="checkbox">` — bayroqcha o'rnatish maydoni;
- `<input type="radio">` — yoqib o'chirish elementi;
- `<input type="reset">` — barcha forma maydonlarini tozalash tugmasi;
- `<input type="submit">` — web serverga barcha ma'lumotlarni yuborish tugmasi;
- `<input type="button">` — oddiy buyruq tugmasi;
- `<input type="hidden">` — formaning yashirin elementi;
- `<textarea>tekst</textarea>` — ko'p qatorli matn kiritish maydoni;
- `<select><option>Element</option></select>` — tanlash ro'yxatini xosil qilish.

Forms to'plami. Skripdan forma elementlariga ruxsat

Hujjatning barcha formalarga forms to'plami orqali murojaat qilinadi. Masalan, `text1` nomli matnli maydon qiymatini olish lozim bo'lsa, quyidagicha usuldan foydalaniladi:

```
document.forms["form1"].text1.value
```

Formaga murojaat qilish mumkin. Uning ixtiyoriy elementiga murojaat quyidagicha:

```
document.form1.text1.value
```

Alohida formalarga murojaat indeks orqali ham amalga oshirilishi mumkin:

`document.forms[0].text1.value`

Agar element `<form>` tegi ichida joylashgan bo'lsa, murojlatda albatta har bir formani alohida ko'rsatish kerak bo'ladi. Aks holda web-brouzer hujjatning tanasi bo'ylab qidiradi va null qiymatni qaytaradi.

Elementga murojlat qilishga forma ichida yoki ichida emasligidan qat'iy nazar `document` obyektining `getElementById()` metodi imkon beradi:
`document.getElementById("text1").value`

Formaning barcha elementlariga ruxsat `elements` to'plami orqali amalga oshirilishi mumkin:

`document.forms["form1"].elements["text1"].value`

`document.forms["form1"].elements[0].value`

`document.forms[0].elements[0].value`

`document.form1.elements[0].value`

Forma obyekti xususiyatlari

Forma obyekti quyidagi xususiyatlarga ega:

- `length` — forma elementlari soni;
- `action` — formani qayta ishlash dasturining URL-manzili;
- `elements` — `elements` to'plamiga murojaat;
- `encoding` — MIME –tur ma'lumotlarini o'tkazish;
- `method` — forma ma'lumotlarini web serverga qayta uzatish rejimi;
- `enctype` — forma ma'lumotlarini kodirovkalash metodi;
- `name` — forma nomi;

Forma obyekti metodlari

Forma obyekti quyidagi metodlarni qo'llaydi:

- `submit()` forma ma'lumotlarini server dasturiga uzatadi. **Submit** tugmasi bosilishining analogi;
- `reset()` formani tozalash, bunda formaning barcha elementlari boshlang'ich holga o'tadi. **Reset** tugmasi bosilishining analogi.

Forma obyekti xodisalari

Forma obyekti quyidagi xodisalarni qo'llaydi:

- `onsubmit` forma ma'lumotlarini uzatish boshlanishi;
- `onreset` forma tozalanishida paydo bo'ladi.

Boshqaruv elementlari o'zining xususiyati, metodlari va hodisalariga ega. Har bir forma elementini alohida ko'rib chiqamiz.

Matnli maydon va parol kiritish maydoni. E-mail va parol kiritilishini tekshirish.

Forma elementlaridan ma'lumot olish

Matnli maydon va parol kiritish maydoni xususiyatlari:

- value — forma elementi qiymatlari;
- defaultvalue — value parametrda berilgan boshlang'ich qiymat;
- disabled — forma elementlariga taqiq qo'yish: agar berilgan qiymat true bo'lsa forma passiv bo'ladi (och rangda ko'rinadi);
- form — element joylashgan formaga murojaat;
- maxLength — maydonga kiritish mumkin bo'lgan belgilarning maksimal soni;
- name — element nomi;
- type — forma elementi turi;
- readonly — taxrirlashga taqiq: agar berilgan qiymat true, bo'lsa matnli maydonni formatlab bo'lmaydi, false bo'lsa mumkin.

Boshqa metodlar ham shu kabi:

- blur() formaning joriy element fokusini olib tashlash;
- focus() formaning joriy elementiga focus joylahstirish;
- select() maydond matnini belgilash.

Barcha Quyidagi hodisalarni ko'rib chiqamiz:

Fon elementlari quyidagi hodisalarni qo'llab-quvvatlaydi::

- onblur forma elementi fokusi yo'qolishida sodir bo'ladi;
- onchange maydonda ma'lumotlarni o'zgartirishdan keyin, focus olish yoki forma ma'lumotlarini yuborishda yuz beradi. Onblur hodisasidan oldin keladi;
- onfocus formaning elementlariga focus olganda yuzaga keladi.



4.7.1-misol. Formada E-mail va parol kiritilishi to'g'riligini tekshirish

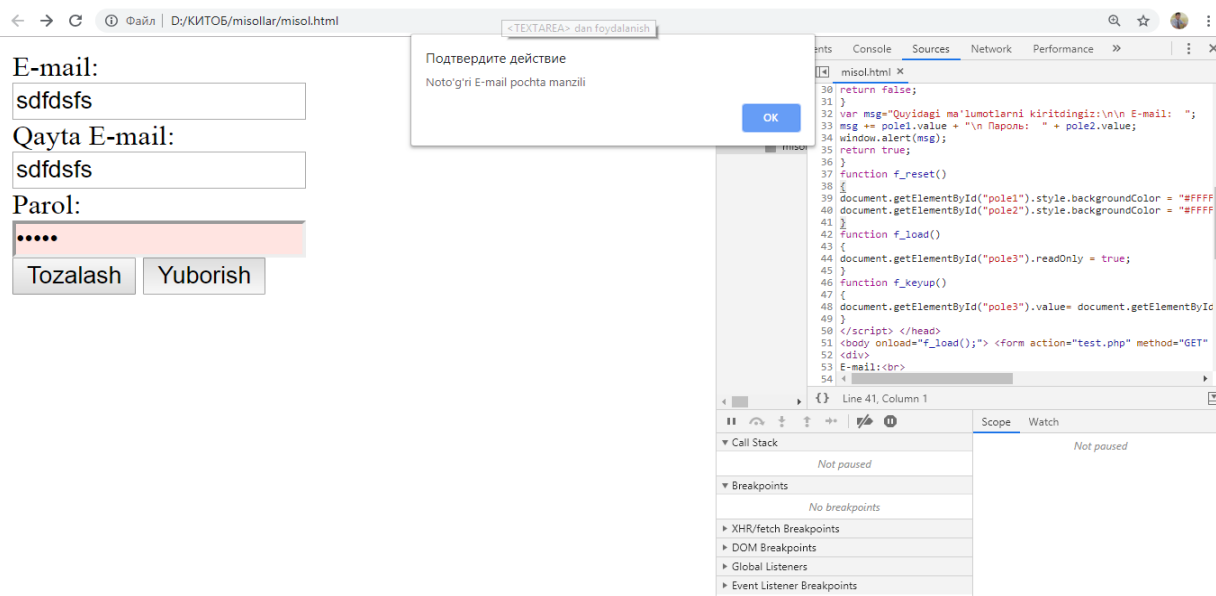
```
<!DOCTYPE HTML>
<html> <head>
<title> Matnli maydondan foydalanish misoli</title>
<meta http-equiv"Content-Type" contentq"text/html; charsetq utf-8">
<script typeq"text/javascript">
function f_submit()
{
var pole1 q document.getElementById("pole1");
var pole2 q document.getElementById("pole2");
pole1.style.backgroundColor q "#FFFFFF";
pole2.style.backgroundColor q "#FFFFFF";
var pq /^[a-z0-9_g`.g`-]Q@([a-z0-9g`-]Qg`)Q[a-z]{2,6}$ /i;
var Str q pole1.value;
if (!p.test(Str))
{
window.alert("Noto'g'ri E-mail pochta manzili");
pole1.style.backgroundColor q "#FFE4E1";

```

```

pole1.select();
return false;
}
p q /^[a-z0-9_g`.g`-]{6,16}$/i;
Str q pole2.value;
if (!p.test(Str)) {
window.alert("Noto'g'ri parol");
pole2.style.backgroundColor q "#FFE4E1";
pole2.select();
return false;
}
var msgq"Quyidagi ma'lumotlarni kiritdingiz:g`ng`n E-mail: ";
msg Qq pole1.value Q "g`n Parol: " Q pole2.value;
window.alert(msg);
return true;
}
function f_reset()
{
document.getElementById("pole1").style.backgroundColor q "#FFFFFF";
document.getElementById("pole2").style.backgroundColor q "#FFFFFF";
}
function f_load()
{
document.getElementById("pole3").readOnly q true;
}
function f_keyup()
{
document.getElementById("pole3").valueq
document.getElementById("pole1").value;
}
</script> </head>
<body onloadq"f_load();" > <form actionq"test.php" methodq"GET" nameq"frm"
idq"frm" onsubmitq"return f_submit();" onresetq"f_reset();" >
<div>
E-mail:<br>
<input typeq"text" nameq"pole1" idq"pole1"
styleq"background-color: #FFFFFF" onkeyupq"f_keyup();" ><br> Qayta E-
mail:<br>
<input typeq"text" nameq"pole3" idq"pole3" styleq"background-color:
#FFFFFF"><br>Parol:<br>
<input typeq"password" nameq"pole2" idq"pole2"
styleq"background-color: #FFFFFF"><br>
<input typeq"reset" valueq"Tozalash">
<input typeq"submit" valueq"Yuborish">
</div> </form> </body> </html>

```



4.7.1-rasm

Ko'p qatorli matn kiritish maydoni. <textarea> maydonidagi matnli maydondan so'z qo'shish

Ko'p qatorli matnlarni kiritish uchun <textarea> juft tegidan foydalaniladi. Quyidagi xususiyatlari bor:

- `maxLength` — maydonga kiritish mumkin bo'lgan belgilarning maksimal soni;
- `wrap` — so'zni ko'chirish rejimi. Quyidagi qiymatlarni qabul qilishi mumkin:
 - `off` — so'z ko'chirilmaydi;
 - `physical` — so'zlar ekranda ham va ma'lumotlarni serverga uzatishda ko'chiriladi;
 - `virtual` — so'zlar faqat ekranda ko'chiriladi, lekin serverga ma'lumotlarni uzatishda emas.

Misol orqali ko'p qatorli matn kiritish maydonini misol orqali yaxshiroq ko'rib chiqamiz. So'z qo'shish **So'z qo'shish** yoki <Enter> tugmasi orqali amalga oshiriladi.



4.7.2-misol. Matn maydonidan <textarea> maydoniga so'z qo'shish

<!DOCTYPE HTML>

<html> <head>

<title> <TEXTAREA> dan foydalanish</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<script type="text/javascript">

function f_submit(){

```

var value1 q document.getElementById("pole1").value;
window.alert("Joriy qiymat:g`n" Q value1);
return false;}
function f_click(){
var pole2 q document.getElementById("pole2");
var text1 q pole2.value;
if (text1!q"") {
document.getElementById("pole1").value q
document.getElementById("pole1").valueQtext1 Q "g`n";
pole2.value q "";
pole2.focus();
}
else {
window.alert("Maydon to'ldirilmagan!");
pole2.focus(); } }
function f_press(e) {
e q e|| window.event;
if(e.keyCodeqq13) {
f_click();
if (e.preventDefault) e.preventDefault();
else e.returnValue q false;
} }
</script>
</head>
<body>
<form actionq"test.php" methodq"GET" nameq"frm" idq"frm"
onsubmitq"return f_submit();">
<div> So'z:<br>
<input typeq"text" nameq"pole2" idq"pole2"
onkeypressq"f_press(event);">
<br> <textarea nameq"pole1" idq"pole1" colsq"15"
rowsq"10"></textarea>
<br><input typeq"button" valueq"So'z qo'shish" onclickq"return
f_click();">
<br> <input typeq"submit" valueq"Maydon qiymati"> </div> </form>
</body> </html>

```

So'z:

Andijon

Namangan
Toshkent
Buxoro

So'z qo'shish
Maydon qiymati

4.7.2-rasm

Qiymatlar ro'yxati. Yangi bo'lim qo'shish imkoniyati. Ishorat o'rnida ro'yxatning qo'llanilishi

Ro'yxat obyekt **hususiyatlari**:

- disabled — ruxsatni taqiqlash: agar true bo'lsa, ro'yxat faol bo'lmaydi(och rangda ko'rinadi);
- form — Element joylashgan formaga silka;
- length — ro'yxat elementlari soni (ruxsat etilgan va yozish uchun);
- multiple — ko'p belgilashga ruxsat: true bo'lsa ro'yxatdan bir vaqtda bir nechta elementlarni belgilash mumkin;
- name — element nomi;
- options — ro'yxat bandlariga silka;
- selectedIndex — belgilangan element tartib raqami (tartib raqam noldan boshlanadi);
- size — ro'yxatda bir vaqtda ko'rinadigan elementlar soni;
- type — forma elementi turi (select-multiple yoki select-one);
- value — ro'yxatdan belgilangan band qiymati.

Ro'yxat bandi hususiyatlari:

- defaultselected — dastlab tanlangan ro'yxatning bandi;
- index — ro'yxatdagi element raqami;
- selected — tanlash bayrog'i: Ro'yxatda ob'ekt tanlangan bo'lsa true;
- disabled — agar true deb belgilansa, ro'yxat elementi faol emas (u och rangda aks etadi). Ushbu hususiyat Internet Explorer web-brauzerining 8.0 versiyadan boshlab qo'llab-quvvatlanadi;
- text — ro'yxat bandi matni;
- value — ro'yxatdan belgilangan band qiymati.

Metodlar:

- blur()joriy kiritish elementidan fokusni olib tashlaydi;
- focus() formaning joriy elementiga fokus qo'yadi.

Xodisalar:

- onblur forma elementidan fokus yo'qola boshlaganda sodir bo'ladi;
- onchange ro'yxatning yangi elementini tanlagandan so'ng sodir bo'ladi;
- onfocus formaning kiritish elementi fokus olganingizda sodir bo'ladi.

Ushbu hodisalardan tashqari siz sichqoncha va klaviaturaning standart hodisalarini ham qo'llashingiz mumkin.

Bayroqcha (Flajok) va o'chirib-yoqqichlar (pereklyuchatel). Belgilangan periklyuchatel qiymatini sikl yordamida olish va bayroqcha o'rnatilishini tekshirish

Bayroqchalar (Flajki) va o'chirib-yoqqichlar(pereklyuchateli) quyidagi hususiyatlarga ega:

- value — formaning joriy elementi qiymati;
- checked — qayd etish belgisi: true agar bayroqcha yoki o'chirib-yoqqich yoniq holatda bo'lsa true bo'ladi;
- defaultchecked — bayroqcha yoki o'chirib-yoqqich sukut rejimida (po umolchaniyu). true yoki false qiymat qaytaradi;
- disabled — belgi taqiqlangan: agar true bo'lsa, element passiv bo'ladi (och rangda ko'rinadi);
- indeterminate — bayroq aniqlanmagan holda joylashgan (kul rangli). true yoki false qaytariladi;
- form — element joylashgan formaga yo'l;
- name — element nomi;
- type — forma elementi tipi.

Uslublar:

- blur()joriy kiritish elementidan fokusni olib tashlaydi;
- focus() formaning joriy elementiga fokus qo'yadi.

Xodisalar:

- onblur forma elementidan fokus yo'qola boshlaganda sodir bo'ladi;
- onfocus formaning kiritish elementi fokus olganingizda sodir bo'ladi.
- onclick element tanlanganda sodir bo'ladi.

Agar o'chirib-yoqqichlar guruxidan tanlangan elementni aniqlash lozim bo'lsa, barcha o'chirib yoqqichlarni sikl ichida olishga to'g'ri keladi. Tanglangan o'chirib-yoqqich qiymatini guruxdagi indeksini ko'rsatgan holda olish uchun item() metodidan foydalanish mumkin. Buni misolda ko'rib chiqamiz (Misol 3.70).



4.7.3-misol. Bayroqcha va ochirib-yoqqich qayta ishlanmasi

```
<!DOCTYPE HTML>
```

```
<html>
```

```

<head>
<title> Bayroqcha va ochirib-yoqqich qayta ishlanmasi </title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script type="text/javascript">
function f_click() {
var msg q "";
if (document.getElementById("check1").checked)
{
msg Qq "Bayroqcha o'rnatilgang`n";
msg Qq "Qiymati: ";
msg Qq document.getElementById("check1").value;}
else { msg q "Bayroqcha o'chirilgang`n" ;}
var value1 q "";
var count q document.frm.radio1.length;
for (iq0; i<count;iQQ)
{
    if (document.frm.radio1.item(i).checked)
    {
        value1 q document.frm.radio1.item(i).value;
        break;
    }
}
if (value1 qq "erkak") {
msg Qq "g`nJins: ERKAKg`n";
}
else {
msg Qq "Jins: AYOLg`n";
}
window.alert(msg);
}
</script>
</head>
<body>
<form action="test.php" method="GET" name="frm" id="frm">
<div>
<input type="checkbox" name="check1" id="check1" value="Ha"
checked>

```

Matn

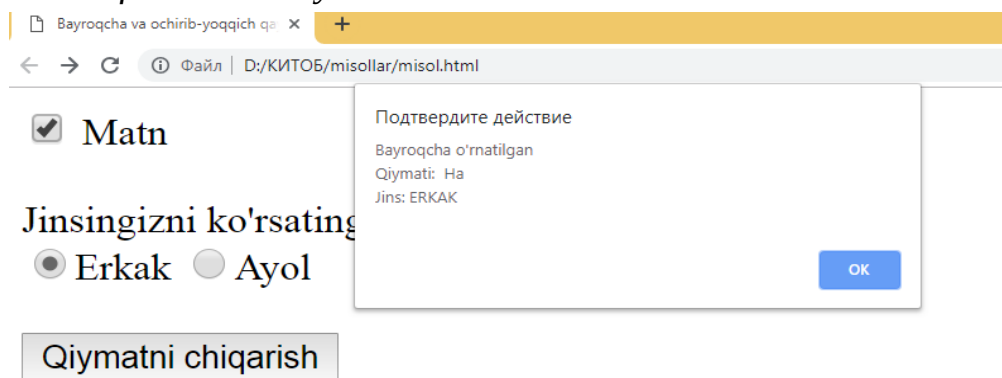
Jinsingizni ko'rsating:

<input type="radio" name="radio1" id="radio1" value="erkak" checked>Erkak

<input type="radio" name="radio1" id="radio2" value="ayol">Ayol

<input type="button" value="Qiymatni chiqarish" onclick="f_click();">

</div> </form> </body> </html>



4.7.3-rasm

Tugmalar. Tugma bosilishini qayta ishlash. Tugmalar ro'yxati.

Tugmalar quyidagi hususiyatlarga ega:

- value — tugmada ko'rindigan matn;
- disabled — taqiqlash belgisi: agar true bo'lsa, tugma passiv bo'ladi (kulrang ko'rinadi);
- form — element joylashgan formaga silka;
- name — element nomi;
- type — forma elementi turi.

Asosiy uslublar:

- blur() joriy kiritish elementidan fokusni olib tashlaydi;
- focus() formaning joriy elementiga fokus qo'yadi.

Xodisalar:

- onblur forma elementidan fokus yo'qola boshlaganda sodir bo'ladi;
- onfocus formaning kiritish elementi fokus olganingizda sodir bo'ladi.
- onclick element tanlanganda sodir bo'ladi.

Quyida keltirilgan misolda (4.7.4-misol) tugma boshlang'ich holda faol bo'lmaydi. Matnli maydonga ma'lumot kiritish bilan tugma faollashadi.



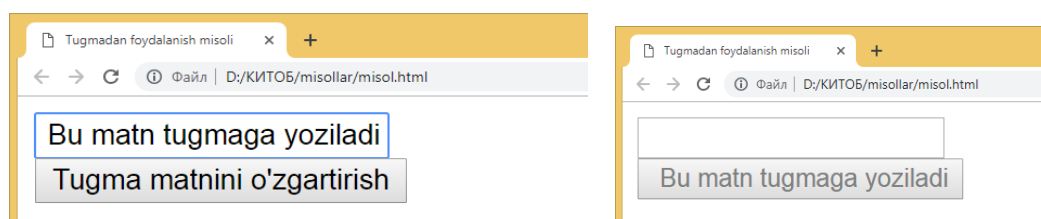
4.7.4-misol. Tugma bosilishini qayta ishlash

<!DOCTYPE HTML>

```

<html> <head>
<title>Tugmadan foydalanish misoli</title>
<meta http-equiv q "Content-Type" contentq"text/html; charset q utf-8">
<script typeq"text/javascript">
function f_up()
{
if (document.getElementById("text1").value qq"")
{document.getElementById("button1").disabledqtrue;}
else {document.getElementById("button1").disabled q false;}
}
function f_click() {
document.getElementById("button1").valueq
document.getElementById("text1").value;
document.getElementById("text1").value q" ";
document.getElementById("button1").disabledqtrue;
}
</script>
</head>
<body>
<form actionq"test.php" methodq"GET" onsubmit q "return false;">
<div>
<input typeq"text" nameq"text1" idq"text1" onkeyupq"f_up();"><br>
<input typeq"button" valueq"Tugma matnini o'zgartirish"
onclickq"f_click();" idq"button1" disabled> </div>
</form> </body> </html>

```



4.7.4-rasm



MUHOKAMA UCHUN SAVOLLAR VA TOPSHIRIQLAR!

1. *JavaScriptdan forma elementlariga ruxsat qanday amalga oshiriladi?*
2. *Forma obyeki xususiyatlarini sanang*
3. *Forma obyeki metodlarini tushuntiring*
4. *Forma obyeki xodisalari qnday*
5. *Matnli maydon va parol kiritish maydoni qiymatlari JavaScriptda qanday qayta ishlanadi?*
6. *Bayroqcha (Flajok) va o'chirib-yoqqichlar (pereklyuchatel) elementi xususiyatlari qanday?*
7. *Tugma qanday xususiyatlarga ega?*

Glossariy

1. **Internet** –dunyodagi turli xil kompyuter tarmoqlari bilan aloqa bog'lab turishni ta'minlovchi texnik vositalar, programma ta'minoti, standart va kelishuvlar yig'indisi.
2. **HTTP (Hyper Text Transfer Protocol)**–bu internet protokoli hisoblanib, uning yordamida bir formatdagi ikki kompyuter o'zaro bog'lanib, muloqot olib borish imkoniyatiga ega bo'ladi.
3. **PPP (Post office protocol)**–oddiy modem liniyalarini internetga kirishda ishlatiladigan kanal darajasidagi protokol,(Analog Slip).
4. **Telnet**–uzoqda turib tarmoqdagi istalgan kompyuterni boshqarish rejimi.
5. **Usenet (Usenet Wewsq roupe)**-tarmoq yangiliklari va tarmoqdagi elektron elonlar doskasini olish.
6. **SLIP (Serial Line Internet Protocol)**–oddiy modem liniyalarini internetga kirishida ishlatiladigan jahon darajasidagi protokol.
7. **Faks-servis**-tarmoq faks serveridan foydalanib, foydalanuvchiga faksimal aloqa orqali xabarlar jo'natish imkonini beradi.
8. **Shlyuzlar**-abonentga TCP/IP protokollar bilan ishlamaydigan tarmoqda xabarlarni jo'natish imkonini beradi.
9. **Elektron tarjimon**–o'ziga yuborilgan matnni bir tildan ikkinchi tilga tarjima qilib beradi.
10. **UUCP**–bir Unix-xoctdan boshqasiga axborotlarni nusxalash protokoli. Ko'plab pochta almashuv sistemalari shu protokolga asoslanib tuzilgan.
11. **PAP (Password authentication protocol)**–serverga ulovchi parollar sistemasi.
12. **NNTP (Net News Transfor Protocol)**–tarmoq yangiliklarini uzatuvchi protokol.
13. **Servis markazi**–internetga ulangan ko'plab kompyuter sistemalarini quvvatlovchi markaz.
14. **Clarinet**–foydalanish uchun ko'pchilik servis markazlari bilan imzolanadigan katta yangiliklar xizmati.
15. **FTP (Fili Transfer Protocol)**–fayllarni uzatuv protokoli; kompyuterlararo axborot almashuvining standart usuli.
16. **Veronica (Very Easy Rodent–Oriented Vetwide Index to Computer Archives)**–kalit so'zlar bo'yicha internet tarmog'ining ommaviy arxivida axborotlarni qidirish sistemasi.
17. **WWW (World Wide Web)**–hujjatlararo gipermatn aloqa bog'lash qobiliyatiga ega bo'lgan tarqoq ma'lumotlar bazasi sistemasi.

18. **Netscape Communication**—bu dunyodagi eng ommabop va eng ko'p ishlatiladigan brauzer hisoblanadi.
19. **Unlinked files**—saytning asosiy sahifasi bilan to'g'ridan-to'g'ri bog'lanmagan fayllar miqdori.
20. **Linked files**—sayt asosiy sahifasi bilan bog'langan fayllar miqdori.
21. **All files**—sayt barcha fayllarining miqdori va umumiy o'lchami.
22. **Pictures**—sayt grafik fayllarining miqdori va umumiy hajmi.
23. **Slow pages**—30 sekundan ortiq yuklanadigan HTML—fayllar miqdori («sust sahifalar).
24. **Whois**—Internet tarmog'ining adres kitobi.
25. **WAIS (Wide Arle Information Service)**—kalit so'zlar bo'yicha internet tarmog'ining ma'lumotlar bazasida kuchli axborotlar qidiruv sistemasi.
26. **Gopher**—Internet zaxira va imkoniyatlarni qidirish, ularga bog'lanish va ulardan foydalanish uchun mo'ljallangan interaktiv obolochka (qobig') foydalanuvchi bilan interfeys menyu sistemasi orqali olib boriladi.
27. **Telnet**—uzoqdan kirish. abonentga Internet tarmog'idagi istalgan EHMda ishlash imkonini beradi.
28. **LAN (local area NetWork)**—geografik bir joydagi lokal tarmoq.
29. **WAN (wide Area NetWork)**—katta hududda joylashgan global tarmoq.
30. **NSFNET—IP**—texnologiyasida tashkil qilingan milliy — ilmiy fondning xususiy tarmog'i.
31. **NOC**—Internet tarmoqlari orasida paydo bo'ladigan turli xil muammolarni xal qiluvchi Internet har bir tarmog'ini xususiy ekspluatsion markazi.
32. **IP (Internet Protocol)**—tarmoqdagi paketlarni marshrutlashni ta'minlovchi tarmoqlararo o'zaro harakat protokoli.
33. **TCP (Transmission Control Protocol)**—tarmoqdagi axborot uzatuvini nazorat qilib turuvchi protokol; katta hajmdagi axborotlarning jo'natish muammolarini xal qiladi.
34. **DOMEN (DNS—DOMAIN NAME SYSTEM)**—normalarning domen sistemasi; internet tarmog'idagi kompyuter nomlarini IP-adreslariga o'tkazib beruvchi ma'lumotlar bazasining tarmoq sistemasi.
35. **HTML (Hypertext Markyp Languge)**—gippermatn hujjatlarini yozish uchun mo'ljallangan til.
36. **<HEAD>** — HTML—hujjatning kirish va bosh qismini belgilaydi.
37. **<BODY>**—asosiy matn va axborotni belgilaydi.
38. **<ADRESS>** — bu Web-sahifa to'g'risida ko'proq to'la-to'kis axborot olish uchun kerak bo'ladigan elektron pochta adressiga ega.
39. **** (Unordered List)—tartibga solinmagan ro'yxat tegi.
40. **<LH>** (List Header)—ro'yxat sarlavhasini aniqlovchi teg.

41. (– Ordered List)-nomerlangan–tartibga keltirilgan ro'yxat).
42. <DT> (Definition List)-aniqlash ro'yxatlari tegi.
43. <TABLE>–jadval hosil qilish tegi.
44. **CELLPADDING**–yacheyka ichidagi narsalar va chegarasi orasidagi masofa.
45. **ALIGN**–atributi tasvirning nisbiy egallab turgan joyini boshqarish imkoniyatini yaratib beradi.
46. **ALT**–atributi «alternativ matn», deb ataladigan grafik obraz paydo bo'lishi lozim bo'lgan matn satrini beradi.
47. **CELLSPACING**–yacheyka orasidagi masofa.
48. **BGCOLOR**–jadval foni rangi.
49. **HEIGHT**–jadval balandligi.
50. **BORDERCOLOR**–ramka rangi.
51. **FRAME**–jadval ramkasining qaysidir tashqi qismi chizmasini aniqlaydi.
52. **CELLPADDING**–yacheyka ichidagi narsalar bilan ramka orasidagi bo'sh oraliq razmerini piksellarda beradi.
53. **CELLSPACING**–jadval yacheykalari orasidagi bo'sh oraliqni piksellarda ko'rsatadi.
54. **Void**–tashqi ramkani butunlay yo'qotadi.
55. **Above**–tashqi ramkaning faqat yuqori chizig'i chiziladi.
56. **Below**–tashqi ramkaning faqat pastki chizig'i chiziladi.
57. **Hsides**–tashqi ramkasining faqat gorizontal chizig'i chiziladi, ya'ni yuqori va quyi chiziqlari
58. **RULES**–jadval ramkasini ichki qismining qanday chizilishini ko'rsatadi.
59. **None**–hech qanday ichki ramkalar bo'lmaydi.
60. **Rows**–faqat gorizontal chiziqlar chiziladi, (satrlar orasidagi).
61. **Cols**–faqat vertikal chiziqlar chiziladi, (ustunlar orasidagi).
62. **Ale**–barcha ichki ramkalar chiziladi.
63. **TEXT**–matnli maydon
64. **RADIO**–selektor tugmasi
65. **CHECKBOX**–nazorat indikator
66. **SUBMIT**–anketani jo'natish tugmasi
67. **RESET**–anketani tozalash tugmasi.
68. **Microsoft Front Page**–Web-uzelni dasturlashsiz tezda yetkazish va unda professional ravishda tayyorlangan hujjatlarni nashr etish imoniyatini beradigan maxsus vosita hisoblanadi.
69. **Status satr**–foydalanuvchiga dastur tomonidan berilishi lozim bo'lgan barcha xabarlar bu satrda xabar ko'rinishida paydo bo'ladi.
70. **Page**–loyihalash rejimi.
71. **Folders**–sayt strukturasini aks ettirish rejimi.

72. **Reports**—sayt to'g'risidagi zaruriy axborotlarni aks ettirish rejimi.
73. **Navigation**—sayt navigasiyasi rejimi.
74. **Hyperlinks**—ichki va tashqi aloqalar strukturasi rejimi.
75. **Task**—topshiriq va masalalarni boshqarish rejimi.
76. **NAME**—qatorida fayl yoki papka nomi ko'rsatiladi.
77. **Title**—sahifa sarlavhasi yoki sayt qolgan elementlari nomini ko'rsatadi.
78. **Modified Date**—soni va qaysidir saytning oxirgi o'zgarish vaqtini ko'rsatadi.
79. **Modified** – Bu oxirgi fayl nomi.
80. **Cowwents**—sharhlarni joylashtirish uchun.
81. **All files**—sayt barcha fayllarining miqdori va umumiy o'lchami.
82. **Pictures**—sayt grafik fayllarining miqdori va umumiy hajmi.
83. **Unlinked files**—saytning asosiy sahifasi bilan to'g'ridan-to'g'ri bog'lanmagan fayllar miqdori.
84. **Linked files**—sayt asosiy sahifasi bilan bog'langan fayllar miqdori.
85. **Slow pages**—30 sekundan ortiq yuklanadigan HTML—fayllar miqdori («sust sahifalar»).
86. **Gippermatn hujjat**—bu boshqa hujjatlarga o'tish uchun aloqa bog'lovchini o'zida saqlaydigan hujjat.
87. **Marshrutizator** —(roater)—tarmoq paketlarini marshrutlash bilan shug'ullanadigan kompyuter tarmog'i, ya'ni paketlarning tarmoq bo'ylab eng qisqa harakat marshrutlari tanlab beriladi.
88. **Protokol**—ikki va undan ortiq mustaqil qurilma yoki prosessorlar o'rtasida forma va proseduralarga reklama qiluvchi qoida va kelishuvlar yig'indisi.
89. **Resurs**—foydalanuvchi ixtiyoriga berilish imkoniyati bor bo'lgan sistemaning mantiqiy yoki fizikaviy qismi.
90. **Server**—kompyuter—boshqalarga o'z xizmatini tavsiya qiluvchi tarmoq kompyuteri, ya'ni foydalanuvchilarning talablari, (savollari) bilan shug'ullanadi.
91. **Server**—programma—bitta kompyuter xizmatini boshqa kompyuterga taqdim etish imkonini yaratuvchi tarmoq kompyuter dasturi.
92. **Uzel**—tarmoqning asosiy vazifalarini bajaruvchi tarmoq kompyuteri.
93. **Xost**—tarmoq vazifalaridan tashqari foydalanuvchilarning topshiriqlarini, (programmalar, hisoblash ishlari va b.q.) bajaruvchi tarmoqning ishchi mashinasi ya'ni, bosh EHM.
94. **Shlyuz**—tarmoqni har xal kompyuter sistemalari bilan bog'lab turuvchi o'zaro harakatdagi tarmoqlararo vosita.
95. **PPP (Post office protocol)**—oddiy modem liniyalarini internetga kirishda ishlatiladigan kanal darajasidagi protokol, (Analog Slip).

96. **SLIP (Serial Line Internet Protocol)**—oddiy modem liniyalarini internetga kirishda ishlatiladigan jaxon darajasidagi protokol.
97. **UUCP**—bir Unix-xoctdan boshqasiga axborotlarni nusxalash protokoli. Ko'plab pochta almashuv sistemalari shu protokolga asoslanib tuzilgan.
98. **PAP (Password authentication protocol)**—serverga ulovchi parollar sistemasi.
99. **Clarinet**—foydalanish uchun ko'pchilik servis markazlari bilan imzolanadigan katta yangiliklar xizmati.
100. **FTP (Fili Transfer Protocol)**—foydali uzatuvlar protokoli; kompyuterlararo axborot almashuvining standart usuli.
101. **Veronika (Very Easy Rodent-Oriented Vetwide Index to Computer Archives)**—kalit so'zlar bo'yicha internet tarmog'ining ommaviy arxivida axborotlarni qidirish sistemasi.
102. **E-mail**—Internet ning istagan abonenti bilan pochta xabarlarini almashtirish va xabarlarni uzatish servisi.
103. **Fayl-servis**—boshqa kompyuterga o'z fayliga kirish imkonini beruvchi kompyuter.
104. **Klient**—server zaxiralaridan foydalanuvchi kompyuter yoki programma.
105. **Programma**—server—o'z mijozidan buyurtma qabul qiladi, unga ishlov beradi va mijozga kerakli axborotni qaytaradi.
106. **Portlar**—har xil ilova va qo'shimchalar bilan aloqani qilovchi server programma nomer, (yoki port nomeri).
107. **POP (Post Office Protocol)**—protokol «pochtali ofis». Xost va abonent o'rtasida pochta almashuvi uchun ishlatiladi. Abonent talabi bo'yicha ham almashuv ishlari bajariladi.
108. **Xost—kompyuter** —internetga mustaqil ravishda ulanish xuquqiga ega bo'lgan kompyuterlar.
109. **SMTP (Simple Mail Transfer Protocol)**—xabarlarni jo'natish uchun ishlatiladigan oddiy pochta uzatuv protokoli.
110. **ASCII (American Standart Cade for Information inferchange)**—matnli axborotlarni almashtirish uchun ishlatiladigan amerika standart kodi.
111. **MIME (Multipurpose Internet Mail Extension)**—grafikani, audio va vidiofayllarni (matndan tashqari) uzatuvchi elektron pochta.
112. **Gipersso'liki (gipersvyazi)**—xohlagan serverda saqlanadigan hujjatlarga tayanish, (xavola).
113. **Gipertekst**—ajratib ko'rsatilgan so'z sistemasi orqali qilaoladigan hujjat.
114. **Mazkup togs**—tanishib chikish programmasi uchun HTML standarti tomonidan ta'riflangan simvollar tartibi, (yo'l-yo'rig'i).
115. **HTTP (Hyper Text Transfer Protocol)**—gipermatnni uzatuv protokoli.

116. **Brauzer (Browse)**–matn va grafik interfeysli tanishib chiqish programmasi.
117. **Front Page Express**–HTML va Web sahifani yaratish va jihozlash uchun Web saxifa muharriri.
118. **Gipermediya**–foto audio-fayili gipermatn.
119. **IJK (Internet Jumpstart Kit)**–Internetga tez kirish uchun vositalar to'plami.
120. **ISW (Internet Setup Wizard)**–Internetning ishga tushirish, (ustanovka) masteri.
121. **Dial-upip**–kommutasion liniya bo'yicha seansli ulanish.
122. **Border atributi**–tasvir atrofidagi (qalinligini beradigan piksellarda).
123. **ALIGN atributi**–tasvirning nisbiy egallab turgan joyini boshqarish imkoniyatini yaratib beradi. U quyidagi ahamiyatga ega:
124. **left**–tasvir chapga suriladi, matn esa uning o'ng tomonidan aylanib o'tadi.
125. **Right**–tasvir o'ngga suriladi, matn esa uning chap tomonidan o'tadi.
126. **Top**–tasvir joriy satrning yuqorisiga to'g'rilanadi.
127. **Texttop**–tasvir joriy satrning eng baland simvol cho'qqisiga to'g'rilanadi.
128. **Middle**–tasvir markazi joriy satrning bazaviy chizig'iga to'g'rilanadi.
129. **Avsmiddle**–tasvir markazi joriy satr markaziga aniq to'g'rilanadi.
130. **Baseline**–tasvirning pastki qirrasi joriy satr bazaviy chizig'iga to'g'rilanadi.
131. **Botton**–tasvirning pastki qirrasi joriy satr quyi qismiga to'g'riladi.
132. **Absbotton**–tasvirning quyi qirrasi joriy satr eng quyi qismidagi simvolning quyi qismiga to'g'rilanadi.
133. **COMPACT** parametri ma'nosiz yoziladi va mazkur ro'yxatni kompakt ko'rinishida chiqarish zarurligini brauzerga biriktirib qo'yish uchun ishlatiladi.
134. **TYPE** parametri ro'yxat satri belgilanadigan znachokni berish (yuklash) imkonini yaratadi.
135. **Java Script**–ssenariylarini tuzish tili.
136. **<Script>**–tegi tuzilgan ssenariyni JavaScript tilidagi dasturiy kodga ega bo'lgan tashqi fayl bilan bog'lash imkonini beruvchi SRC parametriga egadir.
137. **Typeof** – inkrement, dekrement, arifmetik va mantiqiy inversiya, tip belgisi.
138. **CGI (Common Gateway Interface)**–standart interfeys bo'lib, u Web-server bilan berilgan ma'lumotlar va mahsuslashgan internet-ilovalari o'rtasida axborot almashinuvini amalga oshirishga imkon yaratadi.

Adabiyotlar ro'yxati

1. M. Aripov, S.Dottoyev, M.Fayziyeva “Web texnologiyalari”. Oq’uv qo’llanma. «Faylasuflar» nashriyoti.Toshkent–2013 yil
2. M.A.Ryabsev, U.B.Xakimov “Web dizayn san’ati”. Oq’uv qo’llanma. «Faylasuflar» nashriyoti. Toshkent – 2014
3. T.A.Zakirova, E. U. Ibragimov, A.K.Masharipov, M.A.Musayeva “Web–dasturlash”. Darslik. Toshkent–2010
4. Nikolay Proxorenok. “HTML, JavaScript, PHP i MySQL”. Sank Peterburg-2010 g.
5. F.M.Mulaydinov Web dasturlash fanidan MA`RUZALAR MATNI Farg’ona - 2014 y.

Internet saytlari

1. www.ziyonet.uz
2. www.google.com
3. www.lesson.itportal.uz
4. <https://www.texnoman.uz>
5. www.php.net
6. www.w3schools.com

Kirish	3
1-BOB. "WEB DASTURLASH" FANIGA KIRISH.	4
1.1-§. <i>Web dizaynning asosiy tushunchalari. Web texnologiya: HTML; Web brauzerlar; Web-saxifa, Web-sayt. Web-server; razmetkali va senariyli dasturlash tillari; "klient-server" texnologiyasi. Server strukturasi. Web xosting.</i>	4
1.3-§. <i>Web dizayn texnologiyalari: saytning mantiqiy va fizik strukturasi</i>	19
1.4-§. <i>Saytning statik va dinamik komponentlari</i>	22
1.5-§. <i>Web-saxifa elementlari</i>	24
2-BOB. HTML. ASOSIY TUSHUNCHALAR.	26
2.1-§. <i>Hujjat tuzilishi. HTMLning asosiy teglari.</i>	26
2.2-§. <i>HEAD bo'limi. Hujjatning texnik tavsifi</i>	30
2.3-§. <i>Matnlarni bezash.</i>	33
2.4-§. <i>Ro'yxatlar.</i>	37
2.5-§. <i>Jadvallar.</i>	41
2.6-§. <i>Freymlar.</i>	46
2.7-§. <i>Grafika va multimedia. Giperishorat teglari.</i>	50
2.8-§. <i>Formalar</i>	57
3-BOB. CSS ASOSLARI	64
3.1-§. <i>CSS (Cascading Style Sheets) asoslari. Selektor, klass, identifikatorlar va ularning CSSda ishlash prinsiplari. Selektor guruxlari.</i>	64
3.2-§. <i>Selektorlar va ulardan foydalanish. Havola selektorlari.</i>	72
3.3-§. <i>CSSda shriftlar bilan ishlash. CSS da matn va ro'yxatlar bilan ishlash. Blok elementlari umumiy konsepsiyalari.</i>	81
3.4-§. <i>Blok va kontenlarni joylashtirish: clip, visibility, overflow, vertical-align. Sloylar bilan ishlash.</i>	107
4-BOB. JAVASCRIPTDA DASTURLASH.	115
4.1-§. <i>JavaScriptning asosiy tushunchalari: xossalalar, metodlar, xodisalar.</i>	115
4.2-§. <i>JavaScriptda ma'lumot tipi va o'zgaruvchilar, ob'ektlar, ifodalar, arifmetik amallar.</i>	120
4.3-§. <i>JavaScriptda jarayonlarini boshqarish elementlari (shartli va sikl operatorlari).</i>	131
4.4-§. <i>Funksiyalar. Dasturning qismlarga bo'linishi.</i>	141
4.5-§. <i>JavaScriptda OYD asoslari (standart, brauzer va dasturchi tomonidan yaratilgan ob'ektlar).</i>	146
4.6-§. <i>JavaScriptda xodisalar. Interaktiv formalar ishlab chiqish.</i>	163
4.7-§. <i>Forma elementlari bilan ishlash</i>	179
Glossariy	191
Adabiyotlar ro'yxati	197

Jakbarov Odiljon Otamirzaevich



Namangan muhandislik - qurilish institutining Informatika va AT kafedrası mudiri, texnika fanlari nomzodi, dotsent. Matematik modellashtirish, Axborot texnologiyalari, Dasturlash hamda Web dasturlash bo'yicha mutaxassis. 1 ta o'quv qo'llanma, 60 dan ortiq ilmiy – uslubiy ishlar muallifi, dasturiy mahsulotlar yaratish bo'yicha 6 dan ortiq guvohnomalari mavjud. 10 dan ortiq Kasb ta'limi (Informatika va AT) mutaxassislik bo'yicha magistrlik dissertasiyalariga rahbarlik qilgan.

Goyipov Umidjon Gulomjonovich



Namangan muhandislik - qurilish institutining Informatika va AT kafedrası o'qituvchisi. Axborotlarga ishlov berishni algoritmlash hamda Web dasturlash bo'yicha mutaxassis. 30 dan ortiq ilmiy teziz va maqolalar, 10 dan ortiq uslubiy ishlar muallifi. Dasturiy mahsulotlar yaratish bo'yicha 4 dan ortiq guvohnomalari mavjud.

Terishga __.__.2019 yilda berildi. Bosishga __.__.2019 yilda ruxsat etildi. Shrift garniturası Times New Roman Uzb. Oq qog'ozga ofset usulida bosildi. Bichimi 60x84 1/16. Shartli bosma tabog'i 8.5. Nashriyot bosma tabog'i 8.5 Adadi 200 nusxa.

Buyurtma №__. Bahosi kelishilgan narxda. «Namangan» nashriyoti. (Namangan shahri, Navoiy ko'chasi,36.) Nashriyot lisenziya raqami AN-156 2009 yil 14 avgustda berilgan.