

**WEB DASTURLASH**  
fanidan  
**MA'RUZALAR MATNI**

# 1-ma'ruza. Web-dasturlash faniga kirish.

## Reja:

- I. Kirish.
- II. Asosiy qism:
  - 1. Web-sahifa, Web-sayt, Web-server;
  - 2. Internet texnologiyasi haqida
  - 3. Web-texnologiya klassifikatsiyasi;
  - 4. Razmetkali tillar: HTML, XML, XHTML, WML;
  - 5. Ssenariyli tillar. "Klient-server" texnologiyasi;
- III. Xulosa.

**Kalit so'zlar:** Web-sahifa, Web-sayt, Web-server, HTML, XML, XHTML, WML, klient-server texnologiyasi.

**Ishdan maqsad:** Web-dasturlash fani bo'yicha ishlash jarayonida talabalarda bilim, ishlash ko'nikmasi va malakalarini shakllantirish va bu jarayonni amalga oshirishda talabalarga ma'lumot berishni tashkil qilish.

## 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 web-texnologiyalarning inson hayotining ta'lim, kommersiya, siyosat, ko'ngil ochar , ... bo'laklariga kirib borganligini tasavvur eta oladi va uning guvohi va foydalanuvchisiga aylanmoqda.

Internet turli xil insonlarni yagona maqsad bilan birlashishiga sabab bo'lmoqda. Hamma Internet tarmog'idan biror turdagi axborot olishga harakat qiladi. Shunday vaqtlar keladiki, hujjatni Internetda chop etish malakasi yozuv mashinasidan foydalanish kabi har bir, hatto o'rta ma'lumotga ega bo'lgan insonning qo'lidan keladi.

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.

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 engillashtirish, 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. Ular serverga kelgan so'rovlarni qayta ishlaydi va qayta ishlash natijasida yangi Web sahifa hosil bo'ladi.

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. Shu bois matematika va informatika yo'nalishida tahsil oluvchi talabalarga Web texnologiyalarni alohida kurs sifatida o'qitila boshlandi.

Ushbu qo'llanma kamchiliklardan xoli emas. Bundan tashqari turli o'quv muassasalarida o'quv ishlab chiqarish amaliyotini o'tkazish bo'yicha turli xil tajribalar to'plangan. Biz quyida web-texnologiyaning asosiy tushunchalari bilan tanishib chiqamiz.

## 1. Web-sahifa, Web-sayt, Web-server

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 esa o'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.

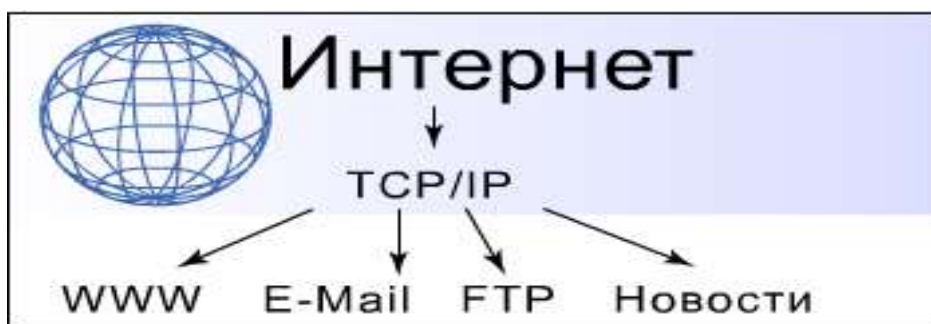
**Web-server** – tarmoqqa ulangan kompyuter yoki undagi dastur hisoblanib, umumiy resurslarni klientga taqdim etish yoki ularni boshqarish vazifalarini bajaradi. Internet tarmog'ini foydalanuvchilarga tarmoq resurslaridan erkin foydalanish imkoniyatini beradigan web-serverlarsiz tasavvur etib bo'lmaydi.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

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

Web-serverda mijoz kompyuteri tizimini tashkil qilishning umumiy tamoyillari nuqtai nazaridan mijoz-server texnologiyalari ishlatiladi.

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

## 2. Internet texnologiyasi haqida



Internet tarmog'ining ishlash prinsipi TCP/IP (Transmission Control Protocol/Internet Protocol - ma'lumotlarni uzatish qaydnomasi/ Internet qaydnomasi) kompyuter tarmog'ida ma'lumotlarni uzatish qaydnomalari majmuining nomidir. TCP/IP jumlasini o'z ichiga Transmission Control Protocol (TCP) va Internet Protocol (IP) qaydnomalar nomlarini birlashtirib olgan qaydnoma bo'lib, u shunday qoidalar majmuiki, TCP/IP barcha kompyuter ishlab chiqaruvchi kompaniyalarning moslamaviy va dasturiy ta'minot xamkorligini ta'minlaydi. Bu qoida jumladan, TCP/IP paketi bilan ishlovchi Digital Equipment firmasi kompyuterlaridan PC kompyuterlariga murojat qilishni kafolatlaydi. TCP/IP ochiq qaydnoma, bu shuni bildiradiki, qaydnoma xaqidagi barcha ma'lumotlar chop etilgan va undan barcha ochiq foydalanadi.

Ko'pchilik foydalanuvchilar TCP/IP ni bitta dastur deb o'ylashadi. Aksincha, u tarmoqning bir vaqtning o'zida ma'lumot uzatish uchun ishlab chiqilgan, o'zaro bog'langan qaydnomalarning butun bir dasturlar oilasidir. TCP/IP tarmoqning dasturlar qismi bo'lib, u TCP/IP oilasidagi har bitta qism ma'lum bir aniq maqsadga qaratilgan: elektron pochta yuborish, sistemaga olis masofalardan kirishni ta'minlash,

fayllarni manzillarga jo'natish, xabarlariga yo'l ko'rsatish yoki tarmoqlardagi buzilishlarni talqin qilish. TCP/IP Internet global tarmog'ida keng foydalanuvchi qaydnomalardir. U xam yirik korporativ tarmoqlarda, shuningdek, kompyuterlar soni oz bo'lgan lokal tarmoqlarda xam qo'llaniladi.

**TCP** (Transmission Control Protocol). Qabul qiluvchi va uzatuvchi kompyuterlarning mantiqiy bog'lanishga asoslangan ma'lumotlar uzatishini qo'llab - quvvatlovchi qaydnoma.

**IP** (Internet Protocol)- Ma'lumotlar uzatishni ta'minlaydi

Internetning paydo bo'lishi tarixi 60-yillarning oxirida Amerika hukumati tomonidan asos solingan ARPANet (Advanced Research Projects Agency tashkiloti) hisoblash tarmog'iga borib taqaladi. Tarmoq harbiy tashkilotlarga xizmat qilgan.

1980 yillar boshlarida ma'lumotlarni uzatishni boshqarish protokoli TCP/IP (Transmission Control Protocol / Internet Protocol) ga asos solindi. Taxminan shu vaqtda ma'lum bo'ldiki, TCP/IP dan turli milliy va xalqaro darajadagi kompyuter tarmoqlarini bog'lashda foydalanish mumkin.

1989 yilning oxirida ARPANet mukammal holga etib keldi, lekin bu vaqtga kelib ko'pgina univetsitetlar va ilmiy muassasalar Internetga ulangan edilar. 1990 yillar boshlarida korporatsiyalar ham Internetdan elektron pochta orqali ma'lumotlar almashishda aktiv ishtirok etardilar. U vaqtlarda Milliy Ilmiy fond tijorat maqsadida Internetdan foydalanishni ta'qiqlagan edi. 1991 yilda bu cheklash bekor qilinadi va Internetdan tashkilot, muassasa, nohukumat tashkilotlarining foydalanish darajasi ortdi, shuningdek, tijorat maqsadida Internetdan keng foydalanila boshlandi (Internet magazinlar, Internet reklamalar va h.k.).

1993 yilda birinchi web-brauzer Mosaic paydo bo'ldi.

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

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

**WWW (World Wide Web)** – bu qanaqadir Internetdan ajratilgan ma'lum bir joy emas, kompyuter aloqa o'rnatadigan biror nima ham emas. Butunjahon o'rgimchak to'rini Internet doirasidagi xizmat deyish to'g'riroq. Web-serverlar deb ataluvchi ma'lum protokollardan, kompyuterlardan foydalanish orqali (chunki ular tarmoqqa ulangan va server dasturiy ta'minotiga ega) Internet xizmati yo'lga qo'yiladi.

Kompyuter web-server bo'lishi uchun Internetga ulangan va server dasturiy ta'minoti (DT) ga ega bo'lishi etarli. Bu DT bilan Windows, Mac OS, Unix kabi operatsion sistemalar ta'minlay oladi. Web-server har doim Internetda "o'tiradi" va talab qilingan tomonga kerakli informatsiyani jo'natadi.

Webda hujjatlar bilan ishlash mumkin. Quyida ulardan ba'zi birlari bilan tanishib chiqamiz.

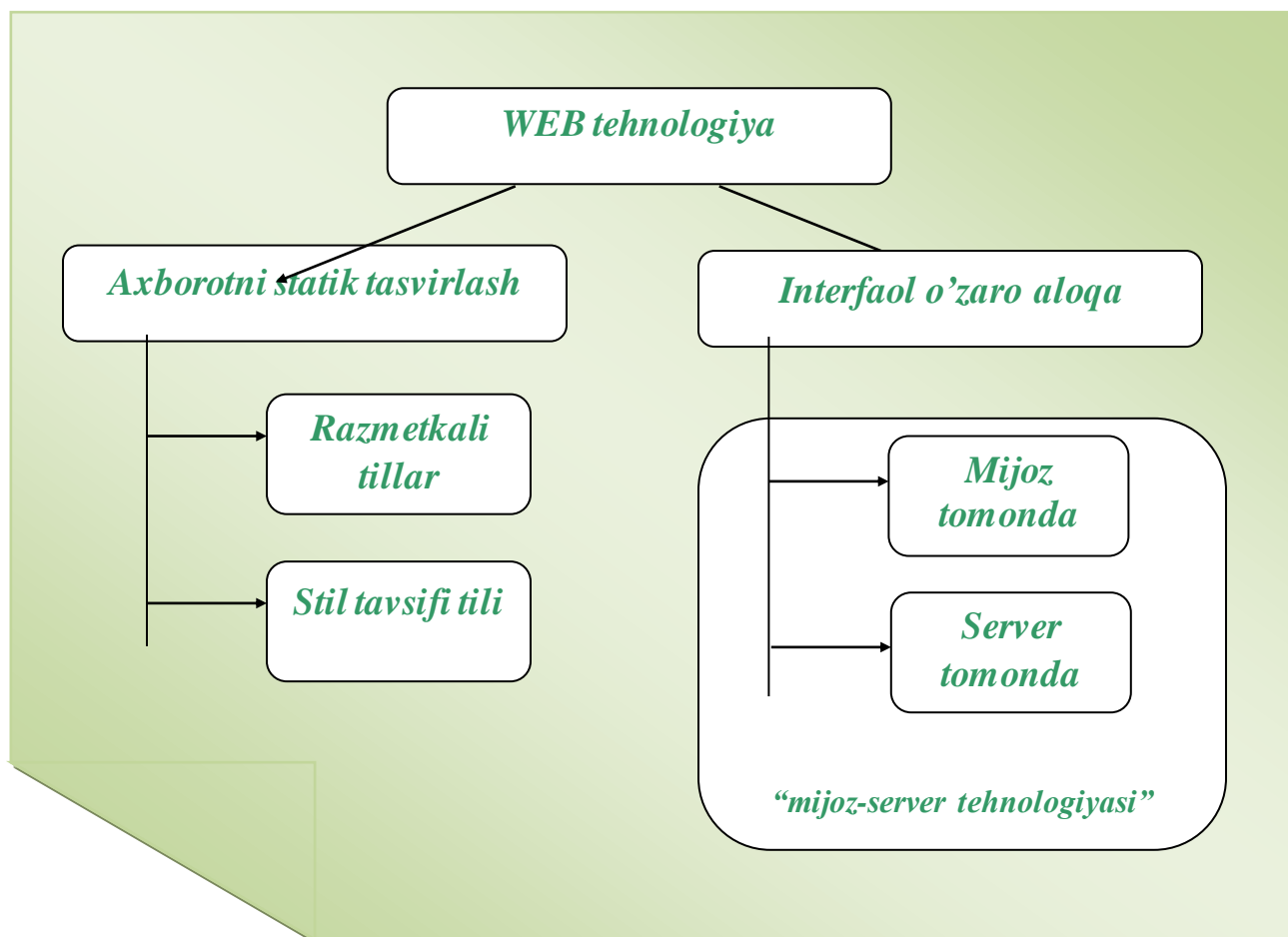
### **Hujjatlar bilan ishlashni tezlashtirish**

Hujjatlar bilan ishlashni tezlashtirish maqsadida Webda ma'lum buyruqlar mavjud. WWW buyruqlar ro'yxati quyidagilardir:

<b>b</b>	avvalgi hujjatga qaytish;
<b>o</b>	hujjatning keyingi sahifasiga o'tish;
<b>g</b> <b>h</b>	Ko'rsatilgan axborot manbaiga bevosita o'tish; saxifani chiqarish (yordam berish yo'li bilan);
<b>Ho</b>	Boshlang'ich hujjatga qaytish;
<b>I</b>	Joriy hujjatda boshqa hujjatlarni murojaatlarini ko'rsatish;
<b>m</b>	Programmadan foydalanish xaqidagi ma'lumotlarni ekranga chiqarish
<b>n</b>	Avvalgi hujjatdan keyingi murojaatga o'tish;
<b>con</b>	Hujjatga murojaat bilan o'tish;
<b>quit</b>	WWWdan chiqish;
<b>V</b>	Ko'rib chiqilgan hujjatlarning ro'yxatini chiqarish;
<b>vcon</b>	Ko'rsatilgan hujjatga qaytish;

<b>return</b>	Bir sahifa pastga o'tish;
<b>t</b>	Hujjatning keyingi sahifasiga qaytish;
<b>u</b>	Hujjat ichida bir sahifaga yuqori chiqish.

### 3. Web-tehnologiya klassifikatsiyasi.



Web-tehnologiyani asosini quyidagi ikkita tushuncha tashkil qiladi:

1. Axborotni statik tasvirlash;
2. Interfaol o'zaro aloqa.

**Axborotni statik tasvirlash.** Ma'lumotlar segmentida joylashgan ma'lumotlar *statik ma'lumotlar* deb ataladi. Bunga asosiy sabab, ular uchun xotira ishlash jarayoni davomida ajratib qo'yiladi. Ishlash davomida esa bu xotira o'zgarmay qoladi. To'plamdagi xotira esa ishlash davomida to'ldirib boriladi va zarur bo'lgan paytda bu xotira bo'shatib qo'yilishi mumkin.

**Razmetkali tillar.** 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.

Web-dasturlash texnologiyalarini, dasturlarini, interfaol o'zaro aloqa qismini ham asosan ikkita qismga ajratish mumkin:

1. klient tomonidagi dasturlarlash (client-side);
2. server tomonidagi (server-side).

**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.

**Server tomonida** bajarilishi kerak bo'lgan ssenariylar odatda sayt papkasining ichidagi maxsus papkaga joylashtiriladi.

#### 4. Razmetkali tillar: HTML, XML, XHTML, WML.

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

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.

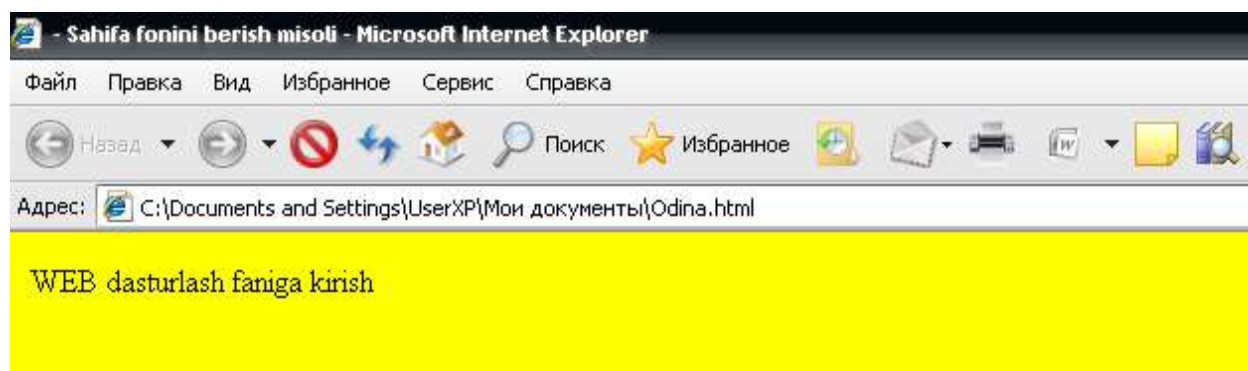
Hujjatlar orasidagi aloqani ta'minlash va ma'lumotlarni formatlash vositalari teg (tag) deb ataluvchi vosita orqali amalga oshiriladi.

Web-sahifaning matn va teglari aralash ravishda HTML-hujjat deb ataluvchi faylining ichiga joylashtiriladi. Qanday tegni qo'llaganingizga qarab brauzer 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'ringining asosiy va HTML ning tarkibiy qismini gipermatnlar va gipermurojaatlar tashkil etadi. Maxsus komandalar yordamida matnning ma'lum qismi shunday ajratiladiki, natijada o'sha matn ustiga sichqon tugmasi bosilsa boshqa matn yoki saxifa 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:

```
<HTML>
  <HEAD>
    <TITLE> - Sahifa fonini berish misoli </TITLE></HEAD>
  <BODY BGCOLOR = YELLOW TEXT = BLACK LINK = RED
    VLINK = PURPLE ALINK = GREEN>
    WEB dasturlash faniga kirish
  </BODY>
```

Bu dasturni ishga tushirish natijasida quyidagi oyna ochiladi:



Gipermatn yoki gipermurojaat biror bir tasvirga ham qo'yilishi mumkinki uning ustiga bosilganda ham yuqorida aytilgan holatlar ro'y berishi mumkin.

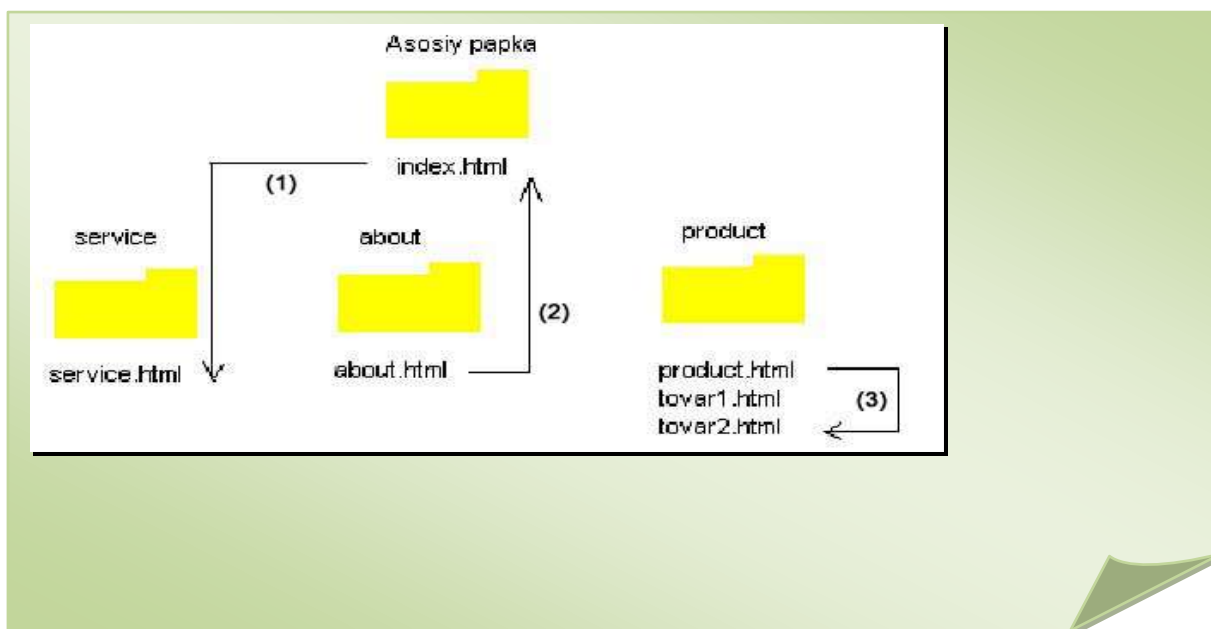
Har bir web-saxifa o'zida bir nechta gipermatn yoki gipermurojaatlarni mujassam etishi mumkin.

Gipermurojaatlar web-saytlar bo'ylab xarakatning asosi xisoblanadi. Murojaatni tanlaganda foydalanuvchi brauzer oynasiga yuklanuvchi yoki yordamchi programmani 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:



(1) ning adresi : "Service/service.html"

Asosiy papkadan ixtiyoriy ichki papkadagi veb-saxifaga murojaat : "Papka\_nomi/fayl\_nomi.html" ko'rinishda beriladi.

(2) ning adresi : "../index.html"

Ixtiyoriy ichki papkadan asosiy papkadagi asosiy web-saxifaga murojaat : "../asosiy\_fayl.html" ko'rinishda beriladi.

(3)ning adresi : "Tovar2.html"

Bir papkadagi veb-saxifalardan bir-biriga murojaat : "fayl\_nomi.html" ko'rinishda beriladi

Internet xizmatlarining ko'pchiligiga dostup (ruxsat, yo'l) adresatsiya sxemasi (URL) yordamida qiziqtirilgan ixtiyoriy hujjatni topish imkoniyatini beradi. Har bir tur boshqasidan farq qiluvchi o'zining format adresiga ega.

URL dan foydalanib, web-brauzerlar yordamida ixtiyoriy hujjat va xizmatlarga dostup olish mumkin. URL quyidagi tartibda yoziladi:

**Protokol://internet\_adres /yo'l /fayl\_nomi.kengaytma**  
**Protokol://internet\_adres**

yoki

URL ga misol:

**http://www.microsoft.com/windows/index.html**

Bu erda:

<b>http://</b>	–	<b>protokol;</b>
<b><u>www.microsoft.com</u></b>	–	<b>internet_adres</b> (Microsoft kompaniyasinnig web- serverining nomi)
<b>/windows/</b>	–	<b>yo'l</b>
<b>index</b>	–	<b>fayl_nomi</b>
<b>html</b>	–	<b>kengaytma</b>

URL da qo'llaniladigan protokollar ro'yxati:

Protokol nomi	Protokol nimaga dostup berishi mumkinligi
http://	HTTP (web) serverlariga
https://	Shifrlangan ba'zi bir HTTP (web) serverlarga
file://	Foydalanuvchi qattiq diskidagi fayllarga
ftp://	FTP server fayllariga
gopher://	Gopher menyu va fayllariga
<u>news://</u>	Usenet yangiliklar serverlari gruppasiga
news:	Aniq Usenet yangiliklar gruppasiga
mailto:	Aniq elektron pochta adresiga
telnet:	Telnet udalen serveriga

**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:

```
<?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:



Buni quyidagicha qisqartiriladi:

## XHTML

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

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

## 5.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 orttirish maqsadida qo'llaniladi.

JavaScript yordamida odatda ma'lumotli va muloqot oynalarini chiqarish, animatsiyalarni ko'rsatish kabi vazifalarni bajarish mumkin. Bundan tashqari, JavaScript-ssenariy ba'zan o'zi ishlab turgan brauzer va platforma 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 ssenariylar.*

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

## *Perl*

Perl tili Web-ilovalar 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 **Chillsoft Лидер среди независимых производителей ASP-средств – компания Chillsoft** 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.

Quyida PHP tilida yozilgan dastur kodini ko'rib chiqamiz:

```
<php
    echo("<p>Hello</p>");// izox
    echo("<p>Hello</p>");// # izox
    /*
        bu ham izox
    */
```

## **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-ilovalar yaratish uchun etarli 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.

## **Xulosa**

Bugungi kunda hayotimizning har bir sohasida internet texnologiyalaridan foydalanmoqdamiz. Internet turli xil insonlarni yagona maqsad bilan birlashishiga sabab bo'lmoqda. Internet texnologiyalarining muhim elementlaridan biri bo'lgan web- texnologiyalar ham taraqqiy etib boryapdi. Hozirda ixtiyoriy inson web-texnologiyalarning inson hayotining ta'lim, kommersiya, siyosat, ko'ngil ochar , ... bo'laklariga kirib borganligini tasavvur eta oladi va uning guvohi va foydalanuvchisiga aylanmoqda. Web-dasturlash fani asosida turli dasturlash tillari yordamida web-saytlar yaratish mumkin. Har bir dasturlash tilining o'ziga hos afzallik va kamchiliklari bor. Ushbu qo'llanmada ularni farqli tomonlarini ko'rib o'tdik.

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. Shu bois matematika va informatika yo'nalishida tahsil oluvchi talabalarga Web texnologiyalarni alohida kurs sifatida o'qitila boshlandi.

Bu fanni o'zlashtirishga bo'lgan chuqur intilish kelajakda yaratilajak zamonaviy ahborot texnologiyalarini saviyasini ko'tarishi shubhasiz.

## **Nazorat savollari:**

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?

## **2-ma'ruza. HTML.razmetkali tiliga kirish.**

**Reja:**

1. HTML kodlarni yozishda qo'llaniladigan matn muharrirlari haqida ma'lumot;
2. HTML tili haqida qisqacha ma'lumot;
3. Hujjat tuzilishi;
4. Hujjatning HEAD bo'limi;
5. Hujjatning BODY bo'limi;
6. HTML shablonni yaratish;
7. HTML shablonlarni saqlash va sinab ko'rish.

**Kalit so'zlar:** Matn muxarrirlari, kod muxarrirlari, WYSIWYG texnologiyasi, HTML, HEAD, BODY

**Ishdan maqsad:** HTML asosiy tushunchalarini o'rganish, uning sarlavha va tana qismiga tegishli bo'lgan asosiy teglar bilan tanishish, talabalarga gipermatnli hujjat tashkil etish asoslari, manbalari, HTML-hujjat haqida tushunchalar berish va HTML-hujjat yarata olish ko'nikmalarini shakllantirishni tashkil etish;

### **1. HTML kodlarni yozishda qo'llaniladigan matn muharrirlari haqida ma'lumot**

Windows muxitidagi matn muxarrirlari: Notepad, TextPad, UltraEdit, EdutPlus. Ko'rsatilgan xamma matn muxarrirlari yordamida HTML kodlarni yozishda qo'llash mumkin.

Ko'pgina matn muxarrirlarida HTML kodlarni xatto programmashtirish tillarida kodlarni yozish juda kulay. Ba'zilar avtomatik ravishda operatorlar, funksiyalarni tanish va ularni xar-xil ranglarda tasvirlash imkoniyatiga ega. Ba'zi matn muxarrirlarida HTML xujjatni web brauzerda sinab ko'rish tugmasi mavjud.

HTML xujjatni yaratishga mo'ljallangan maxsus programmalar (HTML muxarrirlar) ham mavjud: FrontPage, Adobe GoLive, Macromedia Dreamweaver, Netscape Composer. Muxarrirlar 2 turga bo'linadi:

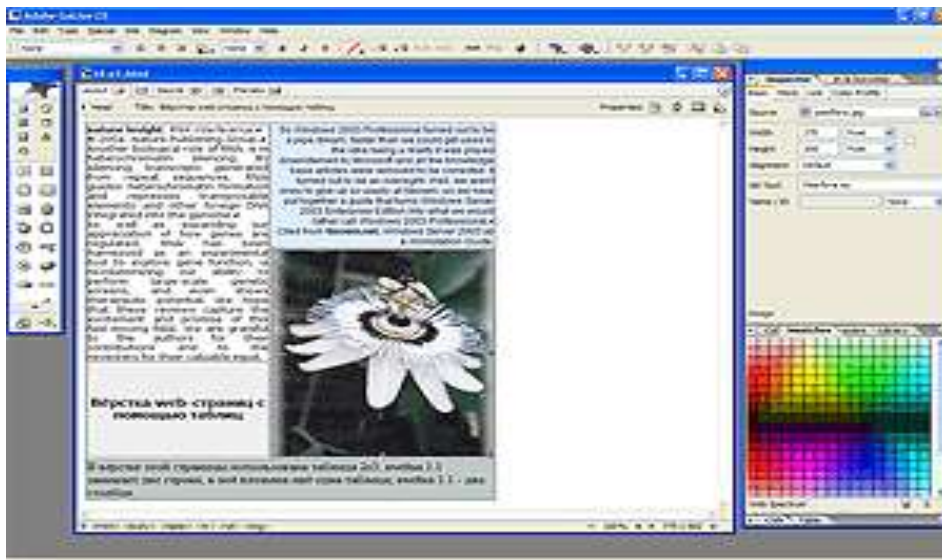
- ✓ kod muxarrirlari;
- ✓ WYSIWYG texnologiyasi (What You See Is What You Get – nimani ko'rsang o'shani olasan) asosida ishlaydigan muxarrirlari. Bu muxarrirlar yordamida foydalanuvchi HTML komandasi va elementlarini yozmaydi, oddiy matn muxarrirlaridek matn yozadi, tasvirlarni kerakli joyga joylashtiradi va x.k. xolos.

### **WYSIWYG-muharriri**

Vizual HTML muharrirlarni 2 asos bo'yicha etirof qilinadi:

1. Adobe tarkibidagi GoLive;
2. Macromedia tarkibidagi Dreamweaver;

Adobe GoLive CS oynasi ko'rinishi quyidagicha:



Macromedia Dreamweaver MX 2004 oynasi ko'rinishi quyidagicha:



## 2. HTML tili haqida qisqacha ma'lumot

**HTML** (ing. Hypertext Markup Language — gipermatnli belgilash tili) - bu **SGML**ga (Standard Generalized Markup Language — standart umumlashtirilgan belgilash tili) asoslangan va xalqaro ISO 8879 standartiga mos keluvchi til, xalqaro to'rda ishlatiladi.

**HTML** tili taxminan 1991-1992 yillarda Yevropa Yadroviy Tadqiqotlar Markazida ishlovchi britaniyalik mutaxassis Tim Bernars Li tomonidan ishlab chiqilgan. Dastlab bu til mutaxassislar uchun xujjat tayyorlash vositasi sifatida yaratilgan. **HTML** tilining soddaligi (SGMLga nisbatan) va yuqori formatlash imkoniyatlarining mavjudligi uni foydalanuvchilar orasida tez tarqalishiga sabab bo'ldi. Bundan tashqari unda gipermatnlardan foydalanish mumkin edi. Tilning rivojlanishi bilan unga qo'shimcha multimedia (rasm, tovush, animatsiya va boshqalar) imkoniyatlari qo'shildi.

Brauserlar - maxsus **HTML** tilida yaratilgan xujjatlarni o'quvchi kompyuter dasturi. Aynan browserlar **HTML** tilida yaratilgan xujjatlarni formatlangan xolda ko'rish imkoniyatini beradi. Xozirda eng mashxur browserlar bu Internet Explorer, Firefox, Opera va xokazolardir.

## Imkoniyatlari

HTML quyidagi formatlash imkoniyatlariga ega:

- Matn qismining mantiqiy rolini belgilash (matn sarlavhasi, paragraf, ro'yxat va hokazo).
- Gipermatnlar yaratish. Bu ayniqsa juda qulay bo'lib o'zaro bog'langan hujjat sahifalari orasida navigatsiya qilishni yengillashtiradi.
- Matnning rangi, qalinligi va boshqa shrift ko'rsatkichlarini belgilash.
- Maxsus belgilar qo'yish. ASCII kodirovkasida ko'rsatilmagan belgilar **HTML** vositalari bilan qo'yish mumkin. Masalan grek alfaviti belgilari  $\alpha, \psi, \zeta$ , matematik belgilar  $\int, \infty, \sqrt{\phantom{x}}, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}$ , moliya belgilari €, £, ¥, ©, ®, ™ va hokazolar.
- Foydalanuvchi kiritishi uchun maydonlar yaratish.
- Multimedia fayllarini ochish.
- Boshqa imkoniyatlar.

### 3. HTML hujjat tuzilishi

**HTML** (Hyper Text Markup Language) – belgili til bo'lib, ya'ni bu tilda yozilgan kod o'z ichiga mahsus ramzlarni mujassamlashtiradi. Bunday ramzlar hujjat ko'rinishini faqatgina boshqarib, o'zi esa ko'rinmaydi. HTMLda bu ramzlarni teg (teg – yorlik, belgi) deb ataladi. HTMLda hamma teglar ramz-chegaralovchilar (< , >) bilan belgilanadi. Ular orasiga teg identifikatori (nomi, masalan V) yoki uning atributlari yoziladi. Yagona istisno bu murakkab chegaralovchilar (<!--va -->) yordamida belgilanuvchi sharxlovchi teglardir.

Aksariyat teglar jufti bilan ishlatiladi. Ochuvchi tegning jufti yopuvchi teg. Ikkala juft teg faqatgina yopuvchi teg oldidan «slesh» («/») belgisi qo'yilishini hisobga olmaganda, deyarli bir xil yoziladi. Juft teglarning asosiy farqi shundaki, yopuvchi teg parametrlardan foydalanmaydi. Juft teg yana konteyner deb ham ataladi. Juft teglar orasiga kiruvchi barcha elementlar teg konteyneri tarkibi deyiladi. Yopuvchi tegda zarur bulmagan bir qator teglar mavjud. Ba'zida yopuvchi teglar tushirib qoldirilsa ham zamonaviy brauzerlar aksariyat hollarda hujjatni to'g'ri formatlaydi, biroq buni amalda qo'llash tavsiya etilmaydi. Masalan, rasm qo'yish tegi <IMG>, keyingi qatorga o'tish <BR>, baza shriftini ko'rsatish <BASEFONT> va boshqalar o'zining </IMG>, </BR> va hokazo yopuvchi juftlarisiz yozilishi mumkin. Noto'g'ri yozilgan tegni yoki uning parametri brauzer tomonidan rad qilinadi. (bu brauzer tanimaydigan teglarga ham taalluqli). Masalan, <NOFRAME> teg-konteyneri faqatgina freymnlarni taniydigan brauzer tomonidan hisobga olinadi. Uni tanimaydigan brauzer <NOFRAME> tegini tushunmaydi. Teglar parametr va atributlarga ega bo'lishi mumkin. Parametrlar yig'indisi har-bir tegda individualdir. Parametrlar quyidagi qoida asosida yoziladi:- Teg nomidan so'ng probellar bilan ajratilgan parametrlar kelishi mumkin;- Parametrlar ixtiyoriy tartibda keladi;- Parametrlar o'zining nomidan keyin keluvchi «=» belgisi orqali beriluvchi qiymatlarga ega bo'lishi mumkin. - Odatda parametrlar qiymati « » - «qo'shtirnoq» ichida beriladi. - Parametr qiymatida ba'zan yozuv registri muxim. Shuni esda tutish lozimki, hamma teglar o'zining individual parametriga ega bo'lishiga qaramay, shunday bir qator parametrlar mavjudki, ularni <BODY> bo'limining barcha teglarida ishlatish mumkin. Bu parametrlar CLASS, ID, LANG, LANGUAGE, STYLE va TITLElardir. HTML-hujjatini yozishni boshlashda ishlatiladigan birinchi teg bu <HTML> tegidir. U har doim hujjat yozuvining boshida bo'lishi lozim. Yakunlovchi teg esa </HTML> shakliga ega bo'lishi kerak. Bu teglar, ular orasida joylashgan yozuvning hammasi butun bir HTML-hujjatini anglatishi bildiradi. Aslida esa hujjat oddiy matnli ASCII-faylidir. Bu teglarsiz brauzer hujjati formatini aniqlab, tarjima qila olmaydi. Ko'pincha bu teg parametrga ega emas. HTML 4.0 versiyasiga qadar VERSION parametri mavjud edi. HTML 4.0da esa VERSION o'rniga <!DOCTYPE> parametri paydo bo'ldi.

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

### 4. Hujjatning HEAD bo'limi



**<HTML>**

**<HEAD>**

\*

**сарлавҳа қисми**

\*

**</HEAD>**

**<BODY>**

\*

**тана қисми**

\*

**</BODY>**

**</HTML>**

HEAD bo'lini sarlavha hisoblanadi va u majburiy teg emas, biroq mukammal tuzilgan sarlavha juda ham foydali bo'lishi mumkin. Sarlavha qismining maqsadi hujjatni tarjima qilayotgan

dastur uchun mos axborotni etkazib berishdan iborat. Hujjat nomini ko'rsatuvchi <TITLE> tegidan tashhari bu bo'limning qolgan barcha teglari ekranda aks ettirilmaydi. Odatda <HEAD> tegi darhol <HTML> tegidan keyin keladi. <TITLE> tegi sarlavhaning tegidir, va hujjatga nom berish uchun xizmat qiladi. hujjat nomi <TITLE> va </TITLE> teglar orasidagi matn qatoridan iborat. Bu nom brauzer oynasining sarlavhasida paydo bo'ladi (bunda sarlavha nomi 60 belgidan ko'p bo'lmasligi lozim). O'zgartirilmagan qolda bu matn hujjatga «zakladka» (bookmark) berilganda ishlatiladi. hujjat nomi uning tarkibini qisqacha ta'riflashi lozim. Bunda umumiy ma'noga ega bo'lgan nomlar (masalan, Homepage,

Index va boshqalar)ni ishlatmaslik lozim. Hujjat ochilayotganda birinchi bo'lib uning nomi aks ettirilishi, so'ngra esa hujjat asosiy tarkibi ko'p vaqt olib, kengayib ketishi mumkin bo'lgan formatlash bilan birga yuklanishini hisobga olgan qolda, foydalanuvchi xech bo'lmaganda ushbu axborot qatorini o'qiy olishi uchun hujjatning nomi berilishi lozim.

## 5. Hujjatning BODY bo'limi

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

<BODY> tegi parametrlari:

ALINK – faol murojaat (ssilka)ning rangini belgilaydi.

BACKGROUND – fondagi tasvir sifatida foydalaniluvchi tasvirning

URL-manzilini belgilaydi.

BOTTOMMARGIN – hujjatning quyi chegaralarini piksellarda belgilaydi.

BGCOLOR – hujjat fonining ranglarini belgilaydi.

BGPROPERTIES – agar FIXED qiymati o'rnatilmagan bo'lsa, fon tasviri aylantirilmaydi.

LEFTMARGIN – chap chegaralarni piksellarda belgilaydi.

LINK – xali ko'rib chiqilmagan ssilkaning rangini belgilaydi.

RIGHTMARGIN – hujjat o'ng chegarasini piksellarda o'rnatadi.

SCROOL – brauzer darchalari xarakatlantirish (prokrutka) yo'laklarini o'rnatadi.

TEXT – matn rangini aniqlaydi.

TOPMARGIN – yuqori chegarasini piksellarda o'rnatadi.

VLINK – ishlatilgan murojaat rangini belgilaydi.

BOTTOMMARGIN, LEFTMARGIN, RIGHTMARGIN va TOPMARGIN parametrlari matn chegarasi va darcha chetlari orasidagi masofani piksellarda belgilaydi. (Faqat HTML 4.0 versiyasidan boshlab IE brauzerlari bu parametrlarni taniy oladi)BGPROPERTIES parametri faqatgina bitta FIXED qiymatiga ega. HTML dagi ranglar o'n oltilik sanoq tizimida (RGB), yoki ranglar nomi yordamida berilishi mumkin. Ranglar bazasi 3 ta rangga – qizil (R) , yashil (G) va ko'k (B) ranglarga asoslangan bo'lib, u RGB deb belgilanadi. har-bir rang uchun 00 dan FF gacha bo'lgan o'n oltilik sanoq tizimidagi qiymat beriladi, bu esa 0 dan 255 gacha bo'lgan diapazonga to'g'ri keladi. So'ngra bu qiymatlar bir songa birlashtiriladi va ularning oldiga “#” belgisi qo'yiladi. Masalan, #800080 siyohrangni bildiradi.

Misollar:

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

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

<BODY LINK q "#ff0000"> yoki <BODY LINK = RED>  
 <BODY LINK = "#00FFFF" ALINK = "#800080"> yoki <BODY VLINK = Aqua ALINK = PURPLE>

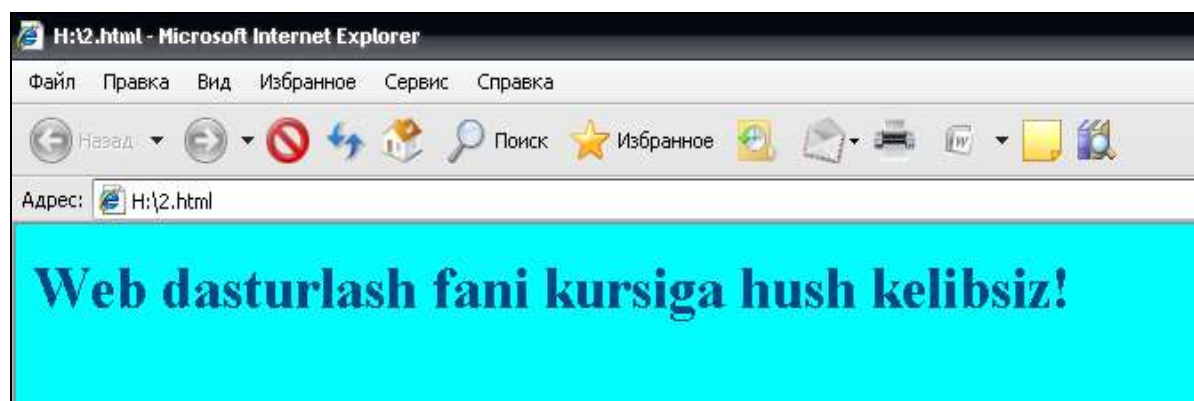
hamma barauzerlar o'n oltilik sanoq tizimidagi standart ranlarni taniydi. Bular quyidagilardir:

Black	#000000	Maroon	#800000
Silver	#C0C0C0	Red	#FF0000
Grey	#808080	Purple	#800080
White	#FFFFFF	Green	#008000
Fuchsia	#FF00FF	Navy	#000080
Lime	#00FF00	Blue	#0000FF
Olive	#808000	Teal	#008080
Yellow	#FFFF00	Aqua	#00FFFF

Quyida ko'rastilgan parametrlar asosida misol ko'rib o'tamiz:

```
<html>
<BODY BGCOLOR = AQUA TEXT = "#848484" LINK = RED VLINK = PURPLE ALINK = GREEN>
<h1> Web dasturlash fani kursiga hush kelibsiz!</h1>
</html>
```

Bu dasturni ishga tushirish asosida quyidagi oynaga ega bo'lamiz:



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

Misol:

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

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

Misol:

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

Misol:

```
<HTML><HEAD><TITLE> - Sahifa fonini berish misoli </TITLE></HEAD>
```

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



</BODY>  
</HTML>

## 6. HTML shablonni yaratish

Notepad ni ochamiz. HTML faylni yaratishni boshlaymiz.

Sarlavxa yozish uchun:

<head>  
</head>

teglari yoziladi.

Brauzer bu teglar o'rtasidagi matnni sarlavxa deb tushunadi va Brauzerning eng tepa qismiga shu matnni yozadi. Endi saxifaning tanasini xosil qilamiz:

<body>  
</body>

Bu teglar o'rtasiga xamma matn va tasvirlarni joylashtirish kerak. HTML shablon bo'lishi uchun HTML saxifaning sarlavxa va tanasini o'z ichiga olgan quyidagi zarur teglar etishmayapti:

<html>  
</html>

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

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

<head> va </head> teglari o'rtasiga quyidagi teglarni joylashtirish mumkin:  
<title> ... </title> - xujjat nomi. Masalan:

<title> web saxifa </title>

<meta /> – ma'lumotlar xaqidagi ma'lumot. Ya'ni bu erda asosiy terminlar yoziladi. Qidiruv sistemalari ishlaganda aynan shu terminlar bo'yicha qidirish ishini olib boradi, saytlarni topadi. Masalan:

<meta name="keywords" content="TATU, Informatika, Institut, Fakultet, Student"/>

content 50-200 ta so'zni o'z ichiga olishi mumkin.  
name=("keywords", "autor", "copyright", "description") – maxfiy qidirishda qo'llaniladi, HTTP serverga dostup berish uchun http-equiv qo'llaniladi.

## 7. HTML shablonlarni saqlash va sinab ko'rish

Veb saxifani yaratishda xamma fayllarni to'g'ri saqlash kerak. Keyin sayt yaratuvchisi o'zi xoxlagan natijaga erishganligini tekshirib ko'rishi kerak.

Saqlash va sinab ko'rish ketma-ketligi:

1. File -> Save File xamma o'zgarishlarni saqlash;
2. Brauzerda shu faylni ochish kerak: Open -> File;
3. Natijani ko'rish;
4. Agar biror joyi to'g'ri ishlamasa, matn muxarririga qaytib xatolarni to'g'rilash kerak;
5. Agar saxifa Web brauzerda ochiq xolda turgan bo'lsa **Обновить** tugmasini bosib o'zgarishlarni tekshirish kerak.

**Nazorat savollari:**

1. Windows muxitidagi matn muxarrirlari haqida tushuncha bering?
2. Matn muxarrirlari turlarini ayting?
3. HTML tegi vazifasini tushuntirib bering?
4. Teglar haqida ma'lumot bering?
5. HTML tilida teglar qanday belgilanadi?
6. HEAD tegi va uning ichidagi teglar xaqida gapiring?
7. <TITLE> va </TITLE> teglar orasidagi sarlavha nomini belgilash qoidalarini ko'rsating?
8. Parametrlar qiymati qanday ko'rinishda beriladi?
9. BODY tegi uning parametrlari haqida ma'lumot bering?
10. HTML shablonlarni qanday saqlanadi?

### 3-ma'ruza. HTML.razmetkali tilining asosiy teglari.

#### Reja:

1. Hujjatning HEAD bo'limi;
2. Mantiqiy va fizik formatlash;
3. HTML-hujjatni formatlash;
4. HTML-hujjatning ichidagi sarlavha teglari.

**Kalit so'zlar:** HTML-hujjat, sarlavha, formatlash teglari, HTML tilining asosiy teglari, HTML tegi parametrlari.

**Ishdan maqsad:** Talabalarda web-sahifa yaratish jarayoni tushunchalari haqidagi bilim va ko'nikmalarni shakllantirish va o'rgatishni tashkil qilish. Bu jarayonni amaliyotga tatbiq qilish.

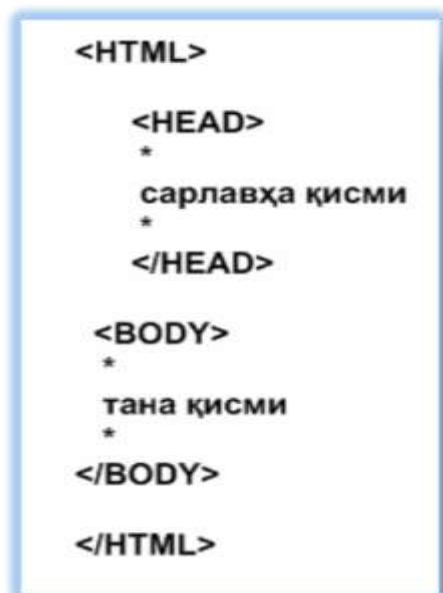
HTML-hujjatini yozishni boshlashda ishlatiladigan birinchi teg bu <HTML> tegidir. U har doim hujjat yozuvining boshida bo'lishi lozim. Yakunlovchi teg esa </HTML> shakliga ega bo'lishi kerak. Bu teglar, ular orasida joylashgan yozuvning hammasi butun bir HTML-hujjatini anglatishi bildiradi. Aslida esa hujjat oddiy matnli ASCII-faylidir. Bu teglarsiz brauzer hujjati formatini aniqlab, tarjima qila olmaydi. Ko'pincha bu teg parametrga ega emas. HTML 4.0 versiyasiga qadar VERSION parametri mavjud edi. HTML 4.0da esa VERSION o'rniga <!DOCTYPE> parametri paydo bo'ldi.

<HTML> va </HTML> orasida 2 bo'limdan tashqil topishi mumkin bo'lgan hujjatning o'zi joylashadi. Mazkur hujjatning birinchi bo'limi sarlavhalar bo'limi (<HEAD> va </HEAD>), ikkinchi bo'lim esa hujjat tana qismidir (<BODY> va </BODY>), uni hujjat tanasi ham deb yuritamiz. Freym tuzilishi hujjatlar uchun <BODY> bo'limining o'rniga <FRAMESET> bo'limidan foydalaniladi.

#### 1. Hujjatning HEAD bo'limi.

BODY bo'linmasida paydo bo'lishi mumkin bo'lgan ba'zi HTML-teglar blok darajasidagi (block level) teglar deb atalsa, boshqalari matn darajasidagi (text level) teglari yoki ketma-ket teg (inline) deb ataladi. Blok darajasidagi teglar o'zida matn darajasidagi teglar yoki blok darajasidagi boshqa teglarni mujassamlashtirishi mumkin. Blok teglari hujjat tizimini ta'riflaydi.

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



Matn muxarrirlari bilan ishlash jarayonidan bilamizki, matnlarni xar-xil ko'rinishda ifodalanishi mumkin: qalinlashtirilgan (полужирный), qo'lyozma shaklida (курсив), tagi chigilgan (подчеркнутый) ... Bu elementlarni ixtiyoriy grafik brauzerlar bir xil ko'rinishda ifodalaydilar.

<HEAD> bo'limida quyidagi ma'lumotlar joylashishi mumkin:

- ✓ Sahifalarni kodlash; (Masalan, bu sahifa Windows-1251 ((кириллица-Windows)) kodlash usulida yoziladi )
- ✓ Sahifalar sarlavhasi;
- ✓ Jadval usulidagi ssilkalar; (ihtiyoriy)
- ✓ Skriptlar;
- ✓ Kalit so'zlar;
- ✓ Muallif ismi;
- ✓ Yaratilgan fayldagi dasturning nomi;

<HEAD> bo'limidagi aksariyat ma'lumotlar «<meta>» tegida kodlanadi:

- ✓ Kodlash: <meta http-equiv="**content-type**" content="text/html;charset=**windows-1251**">
- ✓ Kalit so'zlar: <meta name="**keywords**" content="So'zlarni vergul bilan yozing!">
- ✓ Muallif ismi: <meta name="**author**" content="Odina Imomova">
- ✓ Dastur, ishlab chiqilgan fayl: <meta name="**generator**" content="Adobe GoLive 5">

Boshqa elementlar:

- ✓ Sahifa: <**title- ✓ Skript: <**script//--></script>**
- ✓ Bog'langan fayldagi ssilka: <**link** href="http://www.w3.org">
- ✓ Jadval usulidagi ssilka: <**link** rel="**stylesheet**" href="../usual.css">**

Eslatma: Brauzerda <HEAD> bo'limi aks ettirilmaydi.  
Bu sahifada kodni ko'rish mumkin.

## 2. Mantiqiy va fizik formatlash

HTML-hujjatlarda matnni formatlash uchun shartli ravishda mantiqiy va fizik formatlash teglariga taqsimlasa bo'luvchi teglar yaratilgan. Mantiqiy formatlash teglari fragmentning brauzer yordamida ekranda namoyish etilishiga ta'sir ko'rsatmaydigan strukturaviy belgilashni amalga oshiradi. Shu sababli bunday belgilash mantiqiy deb ataladi. <SITE> tegi sitatlar yoki kitoblar, maqolalar va boshqa manbalarga ssilkalarning nomlarini belgilashda foydalaniladi. Brauzerlar bunday matnni kursiv (qiya) shaklda chiqarib beradi.

HTML tilida quyidagi fizik usuldagi tahrirlovchi teglar mavjud:

Ter	Teg mohiyati
<I>...</I>	Kursiv (Italic)
<B>...</B>	Qalin (Vold)
<TT>...</TT>	Teletayp
<U>...</U>	Ostiga chiziq
<S>...</S>	Ustiga chiziq
<BIG>...</BIG>	Shrift o'lchamini kattalashtirish
<SMALL>...</SMALL>	Shrift o'lchamini kichraytirish
<SUB>...</SUB>	Indeksga yozish
<SUP>...</SUP>	Darajaga yozish

Fizik usuldagi teglarni ishlatish qoidasi:

1. Matnni kiriting;
2. Matn oldiga kursorni olib kelib kerakli tegni ochuvchisini yozing;
3. Matn oxiriga kursorni olib keling;
4. Yopuvchi tegni yozing.

**Masalan:**

<b> Qalinlashtirilgan matn (полужирный) </b>  
 <i> Qo'lyozma shaklidagi matn (курсив) </i>  
 <tt> Xarflar oraligini kengaytirish </tt>  
 <u> Tagi chizilgan matn (подчеркнутый) </u>  
 <big> KATTALASHTIRILGAN MATN </big>  
 <small> kichiklashtirilgan matn </small>  
 $C_n - C_{<sub>n</sub>}$   
 $ax^2+bx+c=0 - ax^{<sup>2</sup>}+bx+c=0$

Mantiqiy usuldagi teglar:

Element	Vazifasi
<em>, </em>	<i> tegiga analog teg.
<strong>, </strong>	<b> tegiga analog teg.
<cite>, </cite>	Mazkur xujjat matniga sitata keltirish.
<dfn>, </dfn>	Dastur kodi.

<samp>, </samp>	Dasturning ishlashiga misol. Oldingisi kabi ishlaydi.
<kbd>, </kbd>	Klaviaturadan kiritiladigan matn.
<var>, </var>	O'zgaruvchi yoki miqdor.
<abbr>, </abbr>	Abbreviatura.
<acronym>, </acronym>	Akronim.

Mantiqiy usullar brauzerga matnni qay tartibda ekranga chop etish kerakligini bildiradi. Mantiqiy usullar fizik usullar ishlamay qoladigan xolatlar uchun xam o'rinli bo'lishi mumkin: uyali telefon internetga ulanganda <b> tegi orqali qalinlashtirilgan matn uchrab qolsa, uni o'qiy olmaydi. Bu xolatda <strong> elementi kerakli natijani bera oladi.

Misol:

<CITE> Darakchi </CITE> eng ommabop gazetalardan biri.

<EM> tegi – (Emphasis – ajratib ko'rsatish, ta'kidlash) matnning muxim fragmentlarini ajratib ko'rsatishda foydalaniladi. Brauzerlar odatda bunday matnni kursiv shaklda aks ettiradi.

Misol:

Matnning <EM> muxim so'zlarini </EM> ajratib ko'rsatish. Bu erda muxim so'zlarini degan ifoda kursivda ajratib ko'rsatiladi.

<DEL> tegi matnni o'chirib tashlangan sifatida belgilaydi. Bu teg orqali belgilangan matnning ustiga chizilgan bo'ladi.

Misol:

Bu <DEL> o'chirilgan matndir </DEL> Bu erda ushbu teglar orasidagi «o'chirilgan matndir» jumlasini quyidagi ko'rinishga ega bo'ladi: o'chirilgan matndir (ustiga chiza olmadik)

Teg CITE va DATETIME parametrlariga ega bo'lishi mumkin. CITE - fragmentning o'chirilib tashlash sabablarini aniqlashtiruvchi hujjatning URL-manzilini ko'rsatadi.

DATETIME – o'chirib tashlanish vaqtini: YYYY-MM-DDThh:mm:ssTZD formatida ko'rsatadi. Bu format o'chirib tashlanish yili, oyi, kuni, soati, daqiqasi va soniyalarini, shuningdek soat tizimi(TimeZone)ni aniqlaydi.

<KBD> tegi matnni foydalanuvchi tomonidan klaviaturada kiritilganidek, ya'ni bir xil kenglikdagi (monoshirinni) shriftida aks ettiradi. Misol uchun, matn muxarririni ishga tushirish uchun <KBD> NOTEPAD </KBD> deb kiriting.

### 3. HTML-hujjatni formatlash

Xar qanday matn ma'lum bir tuzilishga ega bo'ladi. Odatda bunday tuzilishning elementlari – sarlavhalar, ro'yxatlar, kichik sarlavhalar, jadvallar, xatboshi va boshqalar tashkil etadi.

**Xatboshilarga bo'lish.** Xatboshi darajasi tegi.Odatda xatboshilar matnda fikr tugallanganligini ifodalaydi. Oddiy matnli muxarirlarda xatboshi boshqa qatorga o'tish belgisini kiritish (<ENTER> tugmasini bosish) orqali tuziladi. Ammo HTML-hujjatini tuzish jarayonida boshqa qatorga o'tish belgilari xatboshining hosil bo'lishiga olib kelmaydi. Asl hujjatning boshqa qatorga surish belgilari e'tiborga olinmaganligi sababli, muallif hujjati oynasida a'lo darajada ko'ringan matn, brauzer oynasida umuman o'qib bo'lmaydigan darajada bo'lishi mumkin. Shuning uchun HTML-tilida matnlarni xatboshilarga bo'lish uchun <P> tegi kiritilgan. Bu tegni xar bir xatboshi oldidan qo'yish kerak. Yopuvchi </P> tegi bu qolda majburiy emas. Brauzerlar bir necha ketma-ket joylashgan <P> tegini bir tegdek izohlaydi. Odatda baruzerlar xat boshilarni bir-biridan bitta bo'sh qator bilan ajratadi. <P> tegi atributlari:ALIGN gorizontal tekislash atributining qiymatlari:LEFT, RIGHT, CENTER, JUSTIFY.

Misol:

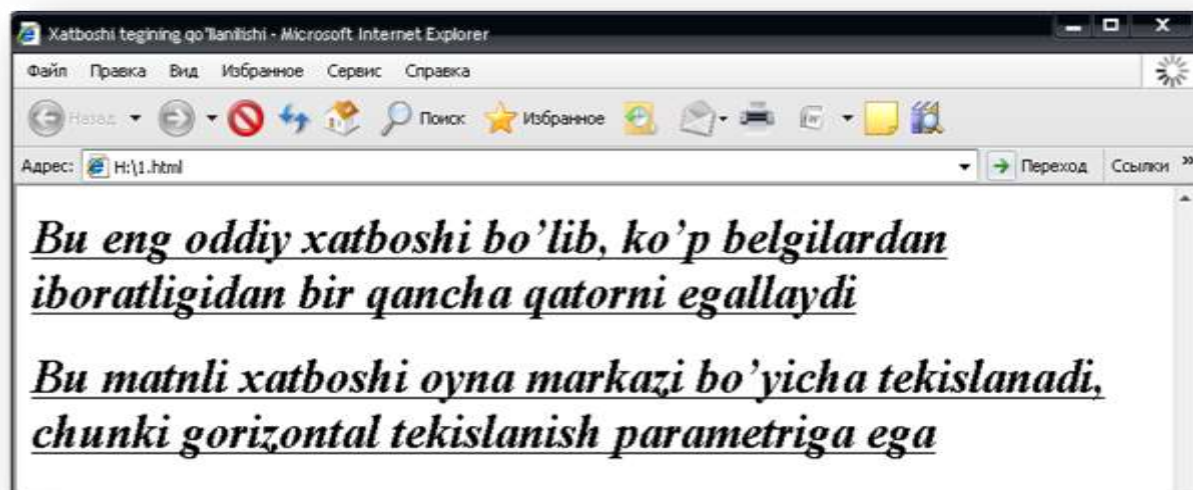
<HTML>

```

<HEAD>
<TITLE> Xatboshi tegining qo'llanilishi </TITLE>
</HEAD>
<BODY>
<h1> <u> <i> <p> Bu eng oddiy xatboshi bo'lib, ko'p belgilardan iboratligidan bir qancha qatorni
egallaydi </P>
<P ALIGN=QCENTER> Bu matnli xatboshi oyna markazi bo'yicha tekislanadi, chunki gorizontal
tekislanish parametriga ega </P> </i> </u> </h1>
</BODY>
</HTML>

```

Ushbu dasturni ishga tushirish natijasida quyidagi oyna yaratiladi:



<BR> tegi majburiy boshqa qatorga surish vazifasini bajarishga hizmat qiladi (yopuvchi tegga ega emas). <BR> tegining kiritilishi keyingi matn yangi qatordan boshlanishini ta'minlaydi. Masalan, bunday usuldan mahsus ro'yxatni belgilash teglarini ishlatmagan qolda ro'yxat shaklidagi tuzilmalarni tuzish yoki she'rlarni yozish kabi hollarda foydalanish mumkin.

<P> tegi kiritiluchi xatboshini inkor etishi tufayli, <BR> tegi yordamida kiritiluvchi xatboshini modellashtirish mumkin.

Masalan,

```

<HTML>
<HEAD>
<TITLE> boshqa qatorga surishning qo'llanilishi </TITLE>
</HEAD>
<BODY>
<P>Keraksiz so'z demagin
<BR> Ko'proq so'zlamog uchun <BR> Oxir dakki emagin, <BR> <BR> deb javob berdi.
</P>
<CITE> Komil Parpiev </CITE>
</BODY>
</HTML>

```

<BR> tegining parametrlari: CLEAR

Parametr ma'nolari: LEFT, RIGHT, ALL, NONE.

<NOBR> tegi boshqa qatorga o'tishni inkor qilish uchun qo'llaniladi. Bu teg orqali belgilangan matn uzunligidan qat'iy nazar bir qatorda joylashtiriladi.

Juda uzun qatorlar hosil bo'lishining oldini olish uchun qator surish mumkin bo'lgan joy ko'rsatilsa, bu zaruriyat tug'ilganda amalga oshiriladi ("yumshoq" tarzda boshqa qatorga surish). Bu maqsadda <WBR> (Word Break) tegidan foydalaniladi.

#### 4. HTML-hujjatning ichidagi sarlavha teglari

<H1 H2 ...H6>.hujjatning alohida bo'limlariga sarlavha berish uchun 6 bosqichli <H1> </H1>, <H2> </H2>, <H3> </H3>, <H4> </H4>, <H5> </H5>, <H6> </H6> teglardan foydalaniladi. 1 raqamli sarlavha eng yirigi hisoblanadi. Eng kichigi esa 6 raqamli sarlavhadir. Sarlavha tegi blok darajasidagi tegdir, shuning uchun ular yordamida alohida matn so'zlarining o'lchamlarini kattalashtirish uchun ularni belgilay olmaydi. Sarlavha teglaridan foydalanilayotganda sarlavhadan oldin va keyin bo'sh qator qoldiriladi, shuning uchun matn boshi tegi va boshqa qatorga surish kerak bo'lmaydi. Sarlavha teglari atributlari: ALIGNTeglar parametri qiymatlari:LEFT, RIGHT, CENTER, JUSTIFY (o'zgartirilmagan bo'lsa LEFT).

Misol:

```
<HTML><HEAD><TITLE> Sarlavha misollari </TITLE></HEAD>
<BODY>
<H1> Sarlavha o'lchami 1 </H1>
<H2> Sarlavha o'lchami 2 </H2>
<H3 ALIGN=CENTER> Sarlavha o'lchami 3 </H3>
<H4 ALIGN=RIGHT> Sarlavha o'lchami 4 </H4>
<H5> Sarlavha o'lchami 5 </H5>
<H6> Sarlavha o'lchami 6 </H6>
Bu erda hujjatning asosiy matni
</BODY> </HTML>
```

<HR> tegi - gorizontal chiziq. Sahifaning u yoki bu qismi tugallanganida vizual ajratib ko'rsatish uchun ishlatiladi. Bu teg yopishni talab qilmaydi. Qatordan oldin va keyin avtomatik ravishda bo'sh qator qoldiriladi.

<HR> tegi atributlari vazifasi:

ALIGN - LEFT, RIGHT, CENTER (o'zgartirilmagan x.)

WIDTH - chiziq uzunligi foiz ko'rsatkichida yoki pikselda belgilanadi.

SIZE - -chiziq qalinligi piksellarda o'rnatiladi.

NOSHADA - -chiziqning relefililigini olib tashlash.

COLOR -- chiziq ranggi o'rnatiladi.

Masalan:

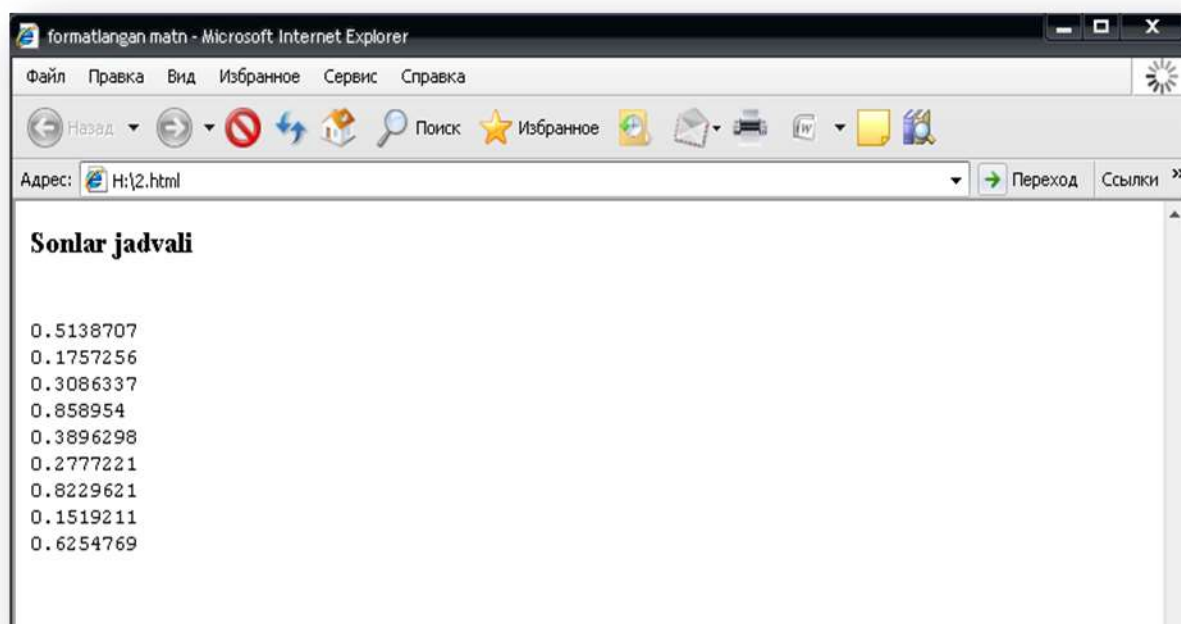
```
<HTML><HEAD><TITLE> gorizontal chiziq </TITLE></HEAD>
<BODY>
<P> Gorizontal chiziq ustiga joylashtirilgan matn </P>
<HR ALIGN=CENTER WIDTH=70% NOSHADE>
<P> Keyin matnning o'zi kiritiladi
</BODY>
</HTML>
```

<PRE> tegi oldindan formatlashtirilgan matnning qo'llanilishidir. An'anaviy usulda, boshqa qatorga surish belgilari yordamida bajarilgan formatlangan matnni kiritish uchun, kerakli bo'sh joy miqdori, tabulyatsiya ramzlari va boshqalar uchun maxsus <PRE> teg konteyneri mo'ljallangan. <PRE> tegi bilan belgilangan matn oddiy matn muxarririda qanday ko'rinishga ega bo'lsa, xuddi shunday shaklda aks etadi. Aks ettirish uchun xar doim bir xil kenglikdagi shrift qo'llaniladi. Bu teg jadvalni belgilovchi maxsus teglarni ishlatmasdan qurilgan jadvallarda qo'llaniladi. Yana bir qo'llanish holati katta bloklarni chiqarishdir.

Misol:

```
<HTML><HEAD><TITLE> formatlangan matn </TITLE></HEAD>
  <BODY>
    <H3> Sonlar jadvali </H3>
    <PRE>
      0.5138707
      0.1757256
      0.3086337
      0.858954
      0.3896298
      0.2777221
      0.8229621
      0.1519211
      0.6254769
    </PRE>
  </BODY>
</HTML>
```

Dastur natijasi quyidagicha bo'ladi:



<PRE> tegining atributi:

WIDTH - atribut qiymati piksel, yoki foizlarda berilishi mumkin va formatlangan matnning maksimal qator uzunligini ko'rsatadi.

<BLOCKQUOTE> tegi asosiy matndan sitatalarni ajratib ko'rsatish uchun qo'llaniladi. Bu teg konteyner bo'lib, turli xil teglarning formatlarini aniqlaydi.

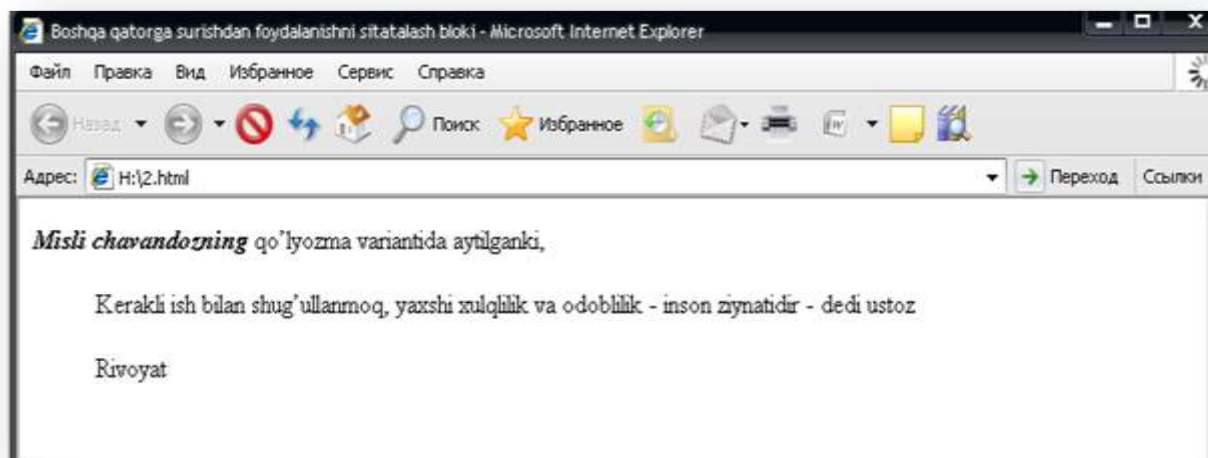
<BLOCKQUOTE> tegi yordamida belgilangan matn aks etish jarayonida asosiy matndan bo'sh qatorlar bilan ajratiladi va o'ng tomonga kichik siljish bilan kiritiladi.

Misol:

```
<HTML><HEAD>
<TITLE>Boshqa qatorga surishdan foydalanishni sitatalash bloki </TITLE></HEAD>
<BODY>
<P>
```

Datur natijasi quyidagi ko'rinishda bo'ladi:





<ADDRESS> tegi hujjat muallifini identifikatsiya qilish va muallif manzilini ko'rsatish uchun qo'llaniladi. Odatda bu teg hujjatning boshida yoki oxirida joylashtiriladi. Bu tegda tez-tez hujjat yaratilgan va so'nggi marotaba yangilangan sanasi ko'rsatiladi.

<ADDRESS> tegi konteyner hisoblanadi.

Misol:

```
<HTML><HEAD><TITLE> manzil bloki </TITLE></HEAD>
```

```
<BODY>
```

```
<h4>
```

```
<P> muloqot uchun ma'lumot </P>
```

```
    <ADDRESS> <P>Jukka Korpela, m.s. (Math) <BR> Helsinki University of Technology  
    Computing Centre <BR>FIN – 02150 Espoo <BR> FINLAND</P> <P> Telephone Internetal  
    Q 35894514319 </P>
```

```
<P> Electronic Mail: HYPERLINK "mailto:J.Korpela@hut.fi" ,J.Korpela@hut.fi </P>
```

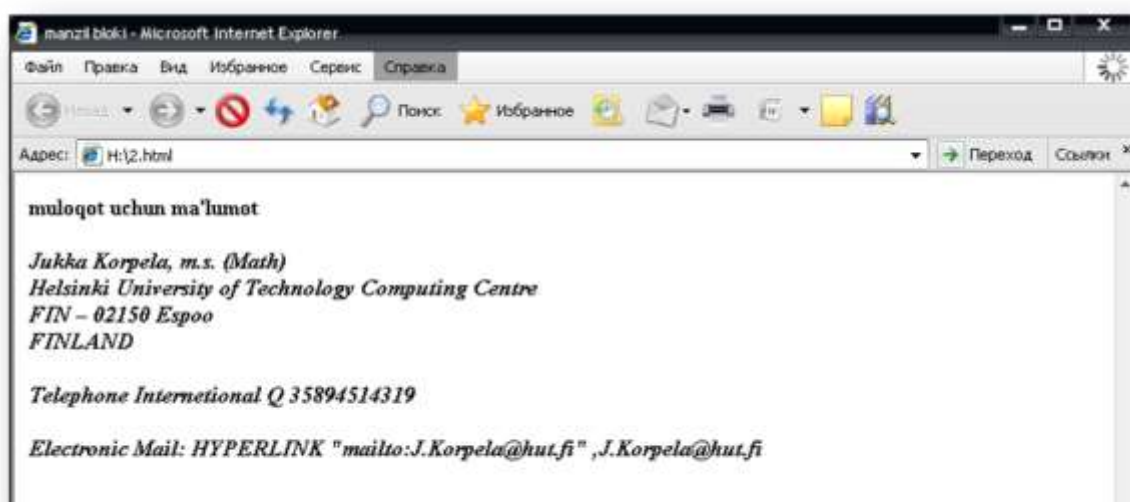
```
</h4>
```

```
</ADDRESS>
```

```
</BODY>
```

```
</HTML>
```

Dastur natijasi quyidagi ko'rinishda bo'ladi:



<! - - ... - - > hujjatga sharxlash (komentariy) kiritish teglari Sharxlar qatorlarning ixtiyoriy sonidan tashkil topishi mumkin. Va <!-- tegi bilan boshlanib, --> tegi bilan tugashi lozim. Teg ichiga kiritilganlarning barchasi sahifani tekshirish chog'ida ishlamaydi. Odatda sharxlar shaxsiy foydalanish uchun mo'ljallangan kuzatishlar uchun muallif tomonidan ishlatiladi. Sharxlarni yozish uchun yana bir teg-konteyner mavjud bo'lib, u ham bo'lsa <COMMENT> dir.

### Nazorat savollari:

1. HTML hujjatning umumiy tuzilishi nimadan iborat?
2. Qanday formatlash teglarini bilasiz?
3. Kommentariylardan foydalanishni izohlang?
4. HTML tegi tarkibida qanday elementlar joylashtirilishi mumkin?
5. Fizik usuldagi tahrirlovchi teglarni ayting va misol yaordamida tushuntiring?
6. Mantiqiy usuldagi tahrirlovchi teglar nimalardan iborat va fikringizni misollar yordamida tushuntiring?
7. Fizik usuldagi teglarni ishlatish qoidasi qanday?
8. Brauzerga matnni qay tartibda ekranga chop etish kerakligini qaysi usul belgilaydi?
9. Qaysi teglar yopuvchi tegga ega emas?
10. Oddiy matnli muharrirlarda hatboshilar nimani ifodalaydi?

## 4-ma'ruza. HTML tilida formalar, freymlar va ob'yektlar.

### Reja:

1. HTML tilida formalar;
2. Formaning berilishi-FORM elementlari;
3. HTML tilida freymlar;
4. Freymlar orasidagi o'zaro ta'sir.

**Kalit so'zlar:** Formalar (<FORM>), freymlar (<FRAME>), ob'ektlar, <INPUT>, ularning parametrlari, <SELECT>,<TEXTAREA>

**Ishdan maqsad:** Web-dasturlash fani bo'yicha HTML tili asosidagi formalar va freymlar bilan ishlash jarayonida talabalarda bilim, ishlash ko'nikmasi va malakalarini shakllantirish va bu jarayonni amalga oshirishda talabalarga ma'lumot berishni tashkil qilish.

### 1. HTML tilida formalar

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

HTML tili web-saxifalar tarkibiga matnli soxalar, menyular, tugmalar kabi interfeys elementlar qo'yish imkoniyatini beradi. Bu ma'lumotlar juda oddiy (elektron adres) yoki yetarlicha murakkab bo'lishi mumkin. Masalan, foydalanuvchi onlayn xolatida internet dasturlari bilan ishlashi mumkin (kasalxona sayti orqali vrach chaqirishi yoki ishxonasining sayti yordamida avtomobilni prokatga olishi mumkin). Elektron formalarni yaratishni maqsadi quyidagicha: u foydalanuvchidan informatsiya so'rashi va undan informatsiya olishi mumkin. Bunday holatlarda ma'lumotlarning bir qismini foydalanuvchi o'ziga tegishli bo'lgan ma'lumotni tanlashi mumkin bo'lgan tayyor menyular tashkil etishi mumkin. So'rovlar quyidagi turlarda bo'lishi mumkin:

Javoblari maxsus matnli soxalarga terilishi kerak bo'lgan savollar (masalan, ismi, familiyasi, elektron adresi, ...).

Javoblari maxsus menyulardagi yoki ro'yxatdagi javoblardan tanlashi kerak bo'lgan savollar (masalan qaysi davlatga tegishli ekanligi, qaysi mavzular qiziqirishi, ...).

Pereklyuchatellar orqali javoblardan yagonasini tanlashga mo'ljallangan savollar (masalan, erkak/ayol, xa/yo'q...).

Formalar HTML tilining kiritish oynasi, tugma, pereklyuchatel kabi elementlar naborlaridan foydalangan xolda yaratiladi. Foydalanuvchi tomonidan yaratilayotgan xar bir element o'zining nomiga ega bo'lib, keyinchalik oldindan aniqlangan biror o'zgaruvchiga beriladi. Masalan, matn kiritish maydoniga shahar nomini berilsa va foydalanuvchi maydonni Jizzax deb to'ldirsa shuni bildiradiki, shahar o'zgaruvchisiga Jizzax qiymati berildi.

O'zgaruvchilar va ularning qiymatlari serverlarga uzatiladi va o'z navbatida server «skriptlar» deb ataluvchi kichik programmalarga murojaat qiladi. Skriptlar olingan ma'lumotlarni qabul qilib, va ularni qayta ishlaydi. Natijada web-saxifada foydalanuvchining ma'lumoti qabul qilinganligi to'g'risida oddiygina ma'lumot (masalan, "raxmat") chiqarib qo'yish mumkin. HTML formalari bilan ishlash uchun skriptlar bilan ishlay olish kerak (ular CGI skriptlar deyiladi va ularni turli programmashtirish tillarida tuzish mumkin. CGI—Common Gateway Interface).

## 2. Formaning berilishi —FORM elementlari

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

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

Quyida FORM elementi parametrlari bilan tanishib chiqamiz:

### *Formaning boshqaruv elementlari —<INPUT> tegi*

Boshqarish elementining yana bir ko'rinishi bu — menyulardir. <Select> elementi yordamida xar xil menyularni xosil qilish mumkin. Bu xolatlarda foydalanuvchiga bir nechta variantlar berilib, undan o'zining javobini tanlash imkoniyati beriladi. Name atributi zaruriy atribut xisoblanadi. Size atributi yordamida esa ekranga bir vaqtda chiqadigan variantlar sonini berish mumkin. Bik .Yu <select> bu konteynordir. Qiymatlar <option> yordamida kiritilib, <select> va </select> lar orasiga joylashtiriladi. Agar foydalanuvchi menyuning biror-bir punktini tanlasa qiymat <select> tegida nomlangan o'zgaruvchiga beriladi. Bik .Yu Selected= "selected" atributi biror qiymatni avtomatik tanlash imkoniyatini beradi (znachenie po umolchaniyu):

```
<select name="javob1">
<option value="first"> Birinchi </option>
<option value="monthly"> Xar oyda </option>
<option value="weekly"> Xar haftada </option>
<option selected="selected" value="daily"> Xar kunda </option>
```

Yuqorida aytilganidek, size atributi yordamida ekranga chiqariladigan javoblar sonini kiritish mumkin:

```
<select name="javob1" size="6">
```

Birinchi misolda javoblar bir qatorda ifodalanadi va sichqon tugmasi bosilganda qolgan javoblar ko'rinadi. Ikkinchi misolda size da necha son ko'rsatilgan bo'lsa shuncha javob varianti ko'rsatiladi va qolgan javoblarni prokrutkalarni siljitish yordamida ko'rish mumkin. Agar javoblardan bir nechtasini bir vaqtda tanlash zarur bo'ladigan bo'lsa <select> ning multiple= "multiple" atributi yordamga keladi. Bu xolatda size da nechta javoblar ko'rsatilgan bo'lsa shulardan bir nechtasini tanlash imkoniyati bo'ladi.

Ushbu teg formaning qaysi nuqtasiga ma'lumot kiritilishini belgilaydi. U foydalanuvchi tomonidan kiritilayotgan ma'lumotlarni formaga keltiradi. Bular matn kiritish maydoni, ro'yxatlar, rasmlar yoki tugmalar bo'lishi mumkin. Maydon tipi TYPE atributi yordamida aniqlanadi.

### ***TYPE=text atributi***

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

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

Foydalanish mumkin bo'lgan yana uchta qo'shimcha atributlar mavjud. Birinchisi `MAXLENGTH` deb ataladi, u foydalanuvchi kiritayotgan matn maydoni maksimum uzunligini belgilaydi. Standart bo'yicha bu qiymat chegaralanmagan. Ikkinchi atribut `SIZE` hisoblanadi, u esa matn maydonini ko'rinib turuvchi qismini belgilaydi. Standart bo'yicha uning qiymati brauzerga bog'liq bo'ladi. Agar `MAXLENGTH` qiymati `SIZE` qiymatidan katta bo'lsa, brauzer ma'lumotni oynaga moslashtiradi. So'nga qo'shimcha atribut matn maydonini boshlang'ich qiymatini belgilovchi `VALUE` dir.

### ***TYPE=checkbox atributi***

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

```
<BR>Rossiya<INPUT NAME="Davlat" TYPE=checkbox  
VALUE="Rossiya">  
Strani SNG<INPUT NAME="Davlat" TYPE=checkbox  
VALUE="SNG">
```

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

### ***TYPE=radio atributi***

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

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

### ***TYPE=image atributi***

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

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

### ***TYPE=password atributi***

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

```
<BR>Nomingiz<INPUT NAME=login>Parol
```

<INPUT TYPE=password NAME="So'z">

### **TYPE=reset atributi**

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

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

### **TYPE=submit atributi**

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

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

### **TYPE=hidden atributi**

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

<TEXTAREA> – ko'p satrli matn kiritishni tashqil etish tegi

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

- Atribut COLS

Maydonning ustunlari (belgilar soni) sonini belgilaydi.

- Atribut NAME

Maydonning nomini belgilaydi.

- Atribut ROWS

Maydonning ko'rinuvchi satrlari sonini belgilaydi.

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

### **<SELECT>- formada ro'yxatdan foydalanish tegi**

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

- MULTIPLE atributi

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

- NAME atributi

Ob'ekt nomini ifodalaydi.

- SIZE atributi

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

- SELECTED atributi

Dastlaki holatda ushbu element tanlangan ekanligini bildiradi.

- VALUE atributi

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

<BR>Tanlash

<SELECT NAME="Tanlash">

<OPTION>Variant 1

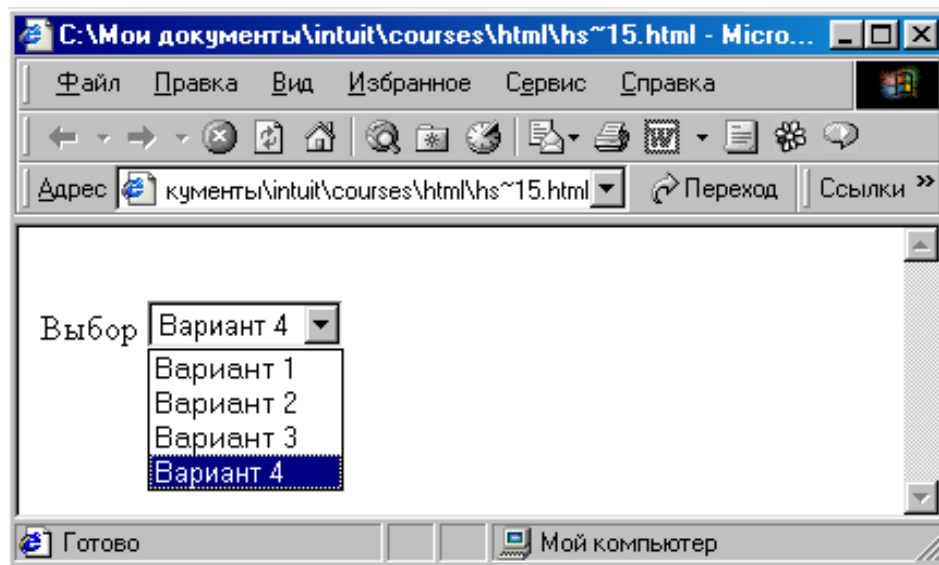
<OPTION>Variant 2

<OPTION VALUE="Variant 3">Variant 3

<OPTION SELECTED>Variant 4

</SELECT>

/



Qalqib chiquvchi menyu

### 3. HTML tilida freymlar

Uzoq vaqtlar web-sahifalar yaratuvchilari butun sayt hattoki portal sahifalari bo'ylab harakatning umumiy sistemasini topishga harakat qildilar. Standart instimentariyni qo'llash uchun bir hil menyuni har bir sahifaga nusxasini qo'yishga to'g'ri kelardi. HTML Frames spetsifikatsiyasini rivojlanishi natijasida brauzer oynasini bir nechta oynalarga ajratib, ularga bir-biridan mustaqil sahifalarni qo'yish imkoniyati paydo bo'ldi.

Freymlar kirib kelishi bilan bog'lik eng asosiy muammo har doimdagidek brauzerlarning o'zaro to'g'ri kelmasligi bo'ldi. Ko'pgina brauzerlar freymlarni «tanimasdi». Freymlar spetsifikatsiyasi HTML standartlariga 1997 yilda yaratilgan HTML 3.2 versiyasidan boshlab qo'llanila boshlandi. U vaqtlarda freymlar Netscape foydalanuvchilari orasida ommaviylashgan edi. HTML 4.0 dan boshlab brauzerlar freymlarni normal «tushuna» boshladilar.

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

Freymlar baruzerni kuzatuv oynasini yonma-yon joylashgan bir nechta to'g'ri burchakli sohalarga bo'lish imkonini beradi. Mazkur bo'laklardan har biriga alohida HTML-fayl, ya'ni boshqalardan mustaqil ravishda ko'zdan kechiriluvchi fayllarni yuklash mumkin. Zaruriyat tug'ilganda freymlar orasida o'zaro bog'liqlikni tashkil etish mumkin. O'zaro bog'liqlik tashqil etilganda freymlardan birida ssilka tanlansa, boshqa freym oynasida kerakli hujjatning yuklanishiga olib keladi.

Garchi HTML-hujjatlarda foydalanuvchiga axborot aks ettirilishining turli usullari havola etilsada, axborotni ifodalashning freym tizimi ham o'zining afzalliklariga ega. quyidagi xollarda aynan freym tizimi qo'l keladi:

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

Freymlar tizimini tasvirlash uchun `<FRAMESET>`, `<FRAME>` yoki `<NOFRAME>` teglaridan foydalaniladi.

`<FRAMESET>` tegi freymlarni belgilaydi.

Freymlardan tashqil topgan Web-sahifalar `<BODY>` bo'linmasiga ega bo'lishi mumkin emas.

`<FRAMESET>` va `</FRAMESET>` konteynerlari xar bir freymni belgilash blokini o'rab turadi. Bunday konteynerning ichida faqat `<FRAME>` teglari yoki kiritilgan `<FRAMESET>` teglari mavjud bo'ladi.

`<FRAMESET>` tegining atributlari:

- ROWS,
- COLS.

Ushbu parametrlar qiymatlari piksellarda, foizlarda yoki nisbiy birliklarda berilishi mumkin. qator yoki ustunlar soni mos ro'yxatdagi qiymatlar soni bilan aniqlanadi. Masalan:

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

`<FRAMESET ROWS = "25%, 50%, 25%">` -

ekraning maqbul balandligidan yuqori qatorning qiymati 25 foiz, o'rta qatorniki 50 foiz, quyi qatorniki 25 foiz ekanligini bildiradi.

`<FRAMESET COLS q="*, 2*, 3*">` - qiymatlar nisbiy birliklarda. "Yulduzcha" – "\*" fazoni proporsional taqsimlash uchun ishlatiladi. Xar bir yulduzcha butunning bir qismini bildiradi. hisoblab topish uchun yulduzchalar oldidagi sonlarni qo'shish va xosil bo'lgan sondan kasrning maxraji sifatida foydalaniladi. Yuqoridagi misolda birinchi ustun darcha umumiy kengligining 1/6, ikkinchi ustun 2/6, uchinchi ustun 3/6 qismini egallaydi.

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

`<FRAME>` tegi alohida fayllarni belgilaydi, bu teg `<FRAMESET>` va `</FRAMESET>` teglari juftligining ichida joylashishi lozim. Masalan:

`<FRAMESET ROWS = "*, 2*">`

`<FRAME>`

`</FRAME>`

`</FRAMESET>`

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

`<FRAME>` tegi atributlari:

- SRC
- NAME
- MARGINWIDTH
- MARGINHEIGHT
- SCROLLING
- NORESIZE
- FRAMEBORDER=YES/NO (Faqat IE lar uchun)

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

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

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

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

NAME parametri berilgan freymga ssilka sifatida ishlatish mumkin bo'lgan freymning nomini belgilaydi. Masalan:

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

frame1 deb nomlangan ushbu freymga ssilka qilinishi mumkin. Masalan:

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

frame1 </A> freymiga other.html faylini yuklash uchun shu yerga sichqoncha kursori bosiladi.

MARGINWIDTH va MARGINHEIGHT atributi freym xoshiya(chegara) kengligini belgilaydi.

Atributlar qiymatlari piksellarda beriladi.

Masalan:

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

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

SCROLLING atributidan prokrutka yo'laklarini aks ettirishni boshqarishda foydalaniladi. Uning sintaksisi

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

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

```
<FRAME NORESIZE>.
```

Tabiiyki NORESIZE atributining bitta freymga nisbatan qo'llanilishi boshqa freymlar o'lchami o'zgartirilishining ham oldi olinishiga sabab bo'ladi.

Garchi freymlar tizini HTML 4.0da standart bilan mustaxkamlangan bo'lsada, <NOFRAMES> tegi freymnlarni qo'llab-quvvatlamaydigan brauzerlar yordamida ko'zdan kechirishda asqotadi. Demak, freymlarga bog'lanmagan brauzerlar uchun <NOFRAMES>va </NOFRAMES> teglari juflligidan foydalaniladi. Masalan:

```
<NOFRAMES> butun HTML-hujjat</NOFRAMES>
```

Mazkur teglar orasiga joylashtirilgan barcha ma'lumotlar freymnlarni qo'llab-quvvatlash imkoniyatiga ega bo'lmagan brauzerlar yordamida aks ettiriladi. Freymlarga bog'langan brauzerlar esa <NOFRAMES> va </NOFRAMES> orasidagi barcha axborotga bog'liq emas. Yuqorida keltirilgan parametrlar ishlatilgan misollarni ko'rib chiqamiz.

### ***Freymlarga misol.***

Ekranida biz 3 qismdan iborat sahifani ko'ryapmiz. Ular o'zgartirishga ehtiyoj bo'lganda kerak bo'ladi. Bu 3 qismda qaysiki, 4-hujjat ichida ta'minlangan turli hujjatlar joylashadi. Barcha hujjatlar bir-biridan mustaqil holda ta'minlanadi. Sahifaning o'ng tomonida Internetda ixtiyoriy kalit so'zlarda qidirish mumkin bo'lgan fomalar joylashgan, sahifani quyi tomonida oddiy surat-ssilkasi joylashgan. Bu strukturani barcha qismi "frameset" deb nomlanuvchi tamomila kichik hujjat asosida tashkil qilingan. (Inglizchadan tarjima qilinganda "frameset"- "frame tashkilotchisi" degan ma'noni anglatadi)

To'liq HTML kodi quyidagi ko'rinishda bo'ladi:

```
<html>
```

```
<head>
```

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

```
<title>HTMLda freymlar</title>
```

```
</head>
```

```
<frameset rows="*,112" border="1" framespacing="1" frameborder="yes">
```

```
<frameset cols="*,200" frameborder="no" border="1" framespacing="1">
```

```
<frame src="framebasic.html" name="basic" scrolling="yes">
```

```
<frame src="frameleft.html" name="left" scrolling="no" noresize>
```

```
</frameset>
```

```
<frame src="framedown.html" name="down" scrolling="yes">
```



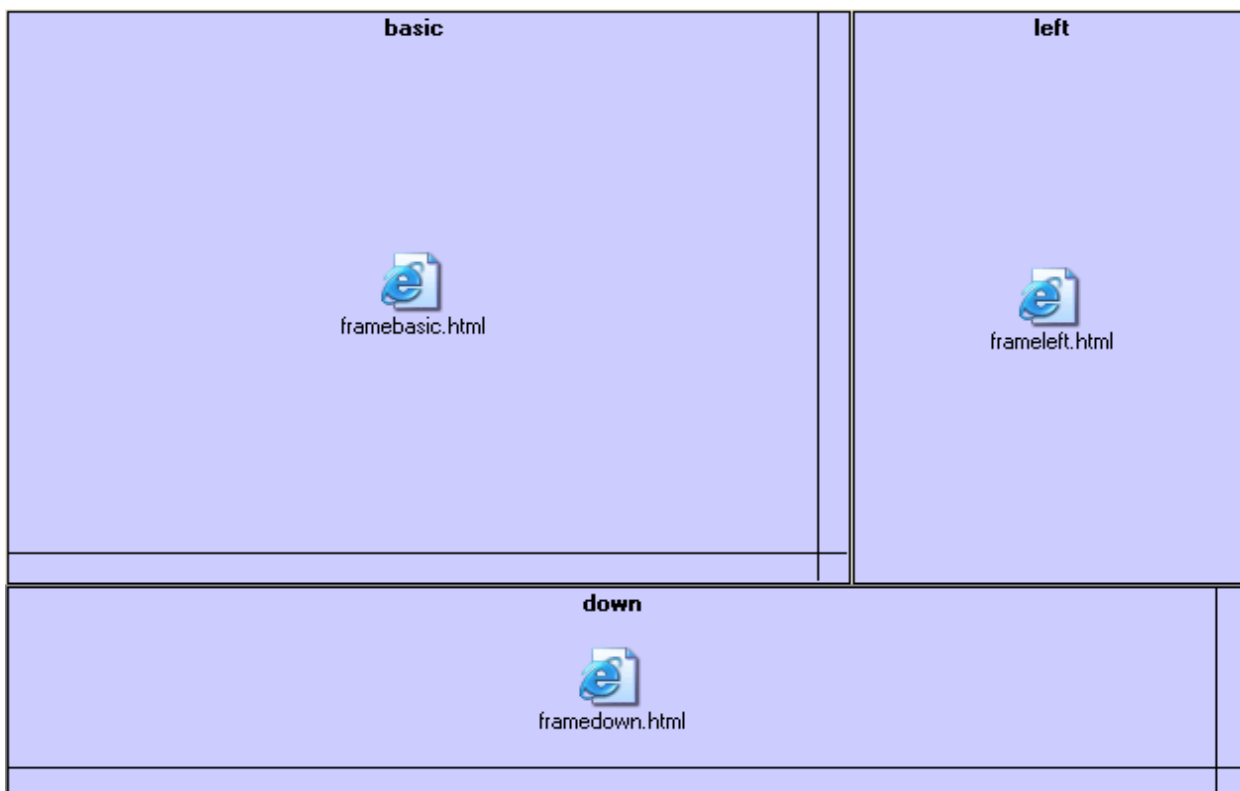
```

</frameset>
<noframes>
<body>
</body>
</noframes>
</html>

```

“Frameset” tegi parametrlari: “rows” parametri- satr balandligini, “frameborder” parametri- yuqori kadr va pastki kadr o’rtasidagi chegara qalinligini ko’rsatuvchi muhim vazifalarni amalgam oshiradi. Qolgan parametrlar majburiy emas. Birinchi freymset o’rniga asosiy framesetdagi yangilangan freymset bo’ladi.

Agar quyidagi rasmlarga yaxshilab qaralsa, hammasi tushunarli bo’ladi:



Biz o’qiyotgan sahifa “basic” nomli va uni o’ng tomonida joylashgan “left” deb nomlanuvchi sahifalar kiritilgan. Ushbu ikki sahifa o’z navbatida kadr paydo qiluvchi frameset hosil qiladi. U va pastki kadr “down” eng muhim freymsetda keladi. Kadrlar ierarhiyasi jadvallar ierarhiyasi o’xshaydi. Kadr jadval va freymset yacheykalarini taqqoslaydi:



<Frame> tegida aylanma yo’llar borligi va kadrlar fayli manzili ko’rsatilishi kerak.(Ko’rib turibmizki, chapdagi kadrda u yo’q.)

Kadrlar orasidagi o’tish ham o’ziga hos xususiyatlarga ega. Masalan, bir kadrda boshqa kadrda yuklangan sahifaga ko’rsatishi mumkin. Buning uchun **target**="..." tegi parametridan foydalaniladi:

`<a href="_url_" target="left">Ssilka matni</a>`

Bu ssilka o'ng kadrda web-sahifa yuklaydi, bu ssilka esa quyida joylashgan. Ssilka kodi-1;

`<a href="sample.html" target="left">I ssilka </a>`

Butun oynada hamma kadrlar o'rniga sahifalar qanday yuklanadi? Buning uchun `Target="_top"` zahira sifatida saqlanadi.

Butun oynaga sahifa yuklash. Butun oyna uchun kod ssilkasi:

`<a href="sample1.html" target="_top"> Butun oynaga sahifa yuklash </a>`

**Ichma-ich freymlar.** Agar ekranda gorizontal va vertikal freymnlarni birdaniga hosil qilmokchi bo'lsak, ichma-ich freymlardan foydalanishimiz mumkin. Masalan tepada to'liq satrli freym, uning pastida ikki ustunga ajratilgan freymnlarni hosil qilaylik. Buning uchun ikki ishni qilish kerak. Birinchidan bizga ikki qatorli freym kerak bo'ladi. Ikkinchidan ikkinchi qator freymni ikki ustunga ajratish kerak:

```
<frameset rows= "100, *">
<frame src= "banner.html" scrolling= "no" norisize = "noresize" />
<frameset cols= "25%, 75%">
<frame src= "index.html" />
<frame src= "viewer.html" marginwidth= "5" marginheight= "5"/>
</frameset>
```

#### 4. Freymlar orasidagi o'zaro ta'sir

Freymlar bilan ishlayotganda foydalanuvchi uchun qulay bo'lgan hujjat yuklash sxemasini yaratish mumkin. Freymlar orasidagi o'zaro aloqa hujjatlarni boshqa freymdagi buyruqlar yordamida aynan tanlangan freymga yuklash imkonini berishidir. Bu maqsadda `<A>` tegining `TARGET` atributidan foydalaniladi. `TARGET` atributi ushbu ssilka ko'rsatayotgan hujjat yuklanuvchi freym yoki brauzer oynasi nomini belgilaydi. O'zgartirilmagan holda ushbu parametr yo'q bo'lganda hujjat joriy freym yoki oynada yuklanadi. Freym nomi sifatida mavjud darcha yoki freym nomi berilishi yoki bo'lmasa yangi oyna ochish uchun yangi nom berilishi mumkin. 4 ta zaxiradagi nomlar bor. Ular:

`_blank`, `_self`, `_top`, `_parent`. Bulardan tashhari `"_"` belgisi bilan boshlanuvchi xar qanday nomdan foydalanish maqsadga muvofiq emas.

`TARGET="_blank"` – hujjatning yangi oynaga yuklanishini ta'minlaydi. Bu oyna nomga ega bo'lmasligi tufayli unga boshqa hujjatni yuklashning iloji bo'lmaydi.

`TARGET="_self"` dan foydalanilganda hujjat joriy freymga yuklanadi.

`TARGET="_top"` hujjatning butun darchaga yuklanishiga sabab bo'ladi.

`TARGET="_parent"` hujjatning joriy freymning freym-ota-onasi tomonidan egallangan soxasiga yuklanishiga olib keladi.

Tarkibi `"A"` freymiga yuklangan `frame_a.html` faylining bittagina `test.html` fayliga `TARGET` parametrining turli qiymatlariga ega 6 ta ssilkasi mavjud.

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

Misol:

```
<iframe src="jad.html" width="300" height="300" frameborder="0" scrolling="auto">
```

Mana sizga freym!!!

```
</iframe>
```

`<iframe>` elementi `<frame/>` elementining `frameborder`, `marginwidth`, `marginheight`, `scrolling` kabi hamma atributlarini qabul qilishi mumkin. Bulardan tashqari uning `width` va `height` atributlari ham mavjudki, ular qo'yilayotgan freymning bo'yi va eni o'lchamlarini piksellarda belgilaydi. Yana `<iframe>` elementi `align` atributiga ega bo'lib, u odatdagidek, `right` va `left` qiymatlar qabul qiladi. `<iframe>` elementining yana bir xususiyati shundaki u o'z ichiga olgan matnni qachonki foydalanuvchi brauzeri ichki freymlarni qo'llamasagina ekranga chiqaradi.

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

Target atributi quyidagi qiymatlar qabul qilishi mumkin:

- `/self`. Murojaat qo'yilgan freym o'zida hujjat ochilishini ta'minlaydi.
- `/top`. Bu qiymatni qo'llab, shu oynaning o'zida freymlarni o'chirib hujjatni yuklash mumkin.
- `/blank`. Hujjatni yangi oynada ochilishini ta'minlaydi.

### Nazorat savollari:

1. HTML tilida formani vazifasini izohlab bering?
2. Freymlar qanday ishlaydi?
3. Hujjatda linkni freymga bog'lashni tushuntiring?
4. Ob'ekt nima va qanaqa ob'ektlarni bilasiz?
5. Formaning qaysi nuqtasiga ma'lumot kiritilishini belgilovchi tegni misollar yordamida tushuntiring?
6. Kiritilayotgan matn maydoni parametrlari qanday belgilanadi?
7. Formada harakatlanuvchi ro'yhat qo'llash uchun qanday parametrlar kerak?
8. Freymlar va formalarni qo'llash bilan qanday imkoniyatlarga ega bo'lishimiz mumkin?
9. Frame tegi atributlarini misollar yordamida tushuntiring?
10. `<iframe>` elementini qo'llash bilan qanday imkoniyatlarga ega bo'lishimiz mumkin?

## 5-ma'ruza. Web-dasturlash fanida mijoz tomonida dasturlash va JavaScript tilini HTML hujjatga joylashtirish.

### Reja:

1. Mijoz tomoni (sohasi) da dasturlash. JavaScript ga kirish;
2. JavaScript ning ob'ektli modeli tushunchasi;
3. JavaScript ning URL-sxemasi;
4. Sinflar ierarxiyasi.

**Kalit so'zlar:** Mijoz tomonida dasturlash, JavaScript tili, Java tili, URL sxemasi, sinflar ierarhiyasi, obyektlar, usullar, xususiyatlar, holatlar.

**Ishdan maqsad:** Web-dasturlash fani asosida ishlash jarayonida mijoz tomonida dasturlash jarayonini o'rganish va JavaScript tilini HTML hujjatga joylashtirish bosqichlari va sinflar ierarhiyasi haqida talabalarda ko'nikmalarni shakllantirish.

## 1. Mijoz tomoni (sohasi) da dasturlash. JavaScript ga kirish

Web sahifani generatsiya qilish jarayonida "mijoz-server " arxitekturasi bilan bog'liq ravishda amalga oshiriladi. Sahifalar mijoz tomonida ham server tomonidagi kabi generatsiya qilinadi. 1995 yilda Netscape kompaniyasi mutaxassislari mijoz tomonidagi sahifalarni generatsiya qilish uchun maxsus dasturlash tili yaratishgan va uni JavaScript deb nomlashgan.

JavaScript – mijoz tomonidagi gipermatnli Web sahifaning ssenariylarini boshqaruvchi tildir. Aniqroq aytadigan bo'lsak, JavaScript – bu faqatgina mijoz tomonidagi dasturlash tili emas. JavaScript ning ajdodi Liveware - Netscape serveri tomonidagi vosita hisoblanadi. Shunday qilib, JavaScript ko'proq mijoz tomonidagi ssenariylarni tashkil etuvchi til sifatida ommaviylashgan.

JavaScript ning asosiy g'oyasi HTML sahifalarni ko'rish vaqtida HTML teg va konteynerlarning atributlari qiymatlarini va xususiyatlarini o'zgartirishdan iborat. Shu sabab sahifani qayta yuklash amalga oshmaydi.

Amaliyotda buni biz, sahifa fonining rangini yoki xujjatdagi rasm xususiyatlarini o'zgartirishda, yangi oyna ochish yoki ogohlantirish berish jarayonlarida ya++ol kuzatishimiz mumkin.

JavaScript tili ECMA (European Computer Manufacturers Association – Evropa kompyuter ishlab chiqarish assotsiatsiyasi) tomonidan standartlashtirilgan. Ushbu standartlar ECMA-262 va ISO-16262 nomlarini keltirib chiqardi. Bu standartlar JavaScript 1.1 ga mosh tushuvchi ECMAScript tilini taqdim etadi. Bugungi kunda JavaScript ning barcha turlari ham ECMA standartiga mos tushavermaydi.

JAVA bu – ko'pgina zamonaviy programmashtirish tillari bilan raqobatlasha oladigan programmashtirish tilidir. Uning boshqa programmashtirish tillaridan farqi unda Internet bilan ishlashga mo'ljallangan dasturlar yaratishga mo'ljallangan. U ayniqsa tarmoqlararo ishlaydigan programmalar yaratuvchilar orasida ayniqsa ommaviylashgan. JAVA dasturlarining ko'p kismini kichkinagina tarmoqlararo ishlaydigan programmalaridir. Ularning kichik o'lchamlari Internet bilan ishlashni optimallashtirishga mo'ljallangan. Ya'ni qancha kam xajmli ma'lumot uzatilsa, saytni, programmani yoki rasmlarni yuklash uchun shuncha kam vaqt ketadi. Shu tariqa appletlar xosil bo'ladi. Bu programmashtirish tilida ixtiyoriy programmalarini tuzish mumkin. JAVA dasturi yaratuvchisi Sun kompaniyasining saytidan ba'zi bir appletlarni olish mumkin:

<http://www.sun.com>

Internetdan ko'plab appletlarni bepul olish mumkin:

[http://www.yahoo.com/Computers\\_and\\_Internet/Programming\\_Languages/Java/Applets](http://www.yahoo.com/Computers_and_Internet/Programming_Languages/Java/Applets)  
<http://java.sun.com/openstudio/>  
va xokazo.

Mana appletlarni veb-saxifaga yuklash misoli:

```
<object codetype= "application/java"  
classid= "java:myapplet.class" standby= "Applet yuklash" width=400 height=350>  
</object>
```

Bu erdagi myapplet.class yuklanayotgan appletning nomi va u veb-saxifa yuklanayotgan katalogda joylashgan. Bundan tashqari appletning to'liq URL adresini ifodalovchi codebase atributini xam ishlatish xam mumkin:

```
<object codetype= "application/java" codebase= "http://www.jdpi.uz/applets/"  
classid= "java:myapplet.class" standby= "Applet yuklash" width= 400 height=350>  
</object>
```

Agar brauzer appletlar bilan ishlay olsa (xamma brauzerlar xam appletlar bilan ishlash imkoniyatiga ega emas) boshqa elementlar kabi appletlar ekranga chiqadi va o'z vazifasini bajaradi.

## 2. JavaScript ning ob'ektlil modeli tushunchasi

Mijoz tomonidagi sahifani yaratishni boshqarishda xujjatning ob'ektlı mexanizmidan foydalanilgan. Bunda har bir HTML-konteyner-bu ob'ekt hisoblanadi va quyidagi uchlikni tashkil etadi:

- Xususiyatlar;
- Usullar;
- Xolatlar.

Ob'ektlı model sahifalar va brauzerlar o'rtasidagi bog'lanish sifatida ko'rinishi mumkin. Ob'ektlı model – bu HTML kod orqali berilgan elementlarni ob'ekt, usul, xususiyat va xolatlar ko'rinishida tanish va ular bilan ishlash demakdir. U yordamida biz brauzerga va foydalanuvchiga murojaat qilishimiz, xabarlar yuborishimiz mumkin. Brauzer bizning buyruqlarimizni bajaradi va ekranda sahifaning kerakli qismlarini o'zgartiradi.

Ob'ektlar bir xil tipli xususiyatlar, usullar va xolatlar to'plamini bir xil tipli ob'ektlar sinflarida birlashtiradi. Ob'ektlarning o'zlari faqat xujjatni brauzer yordamida yuklashda yoki dasturning natijasi sifatida namoyon bo'ladi. Ushbu holatni doimo yodda tutish kerak.

### ***Xususiyatlar***

Ko'pgina HTML-konteynerlarda atributlar mavjud. Masalan, yakor konteynerida `<A ...>...</A>` HREF atributi mavjud. Ushbu atribut uni giper murojaatga aylantiradi:

```
<A HREF=intuit.htm>intuit</A>
```

Agar `<A ...>...</A>` yakor konteynerini ob'ekt sifatida ko'radigan bo'lsak, HREF atributi "yakor" ob'ektini xususiyati hisoblanadi:

```
document.links[0].href="intuit.htm";
```

Barcha atributlar qiymatlarini ham o'zgartirib bo'lavermaydi. Masalan grafik rasmlarning o'lchamlari dastlabki berilgan qiymati asosida aniqlanadi, ya'ni ularni o'zgartirib bo'lmaydi. Ketma-ket kelgan barcha rasmlar qiymatlari o'zining dastlabki qiymatigacha masshtablanishi mumkin. Microsoft Internet Explorer da rasm o'lchamlari o'zgartirilishi mumkin.

Umumiylik uchun rasm xususiyatlari JavaScript da HTML-razmetkada mavjud bo'lmagan ob'ektlarga bo'linadi. Masalan, vosita sifatida Navigator deb nomlanuvchi ob'ektni, yoki JavaScript dagi eng asosiy ob'ektlardan – brauzer oynasi ob'ektini olishimiz mumkin.

### ***Usullar***

JavaScript atamalarida ob'ekt usullari uning xususiyatlarini o'zgartiruvchi funksiyalarni anglatadi. Masalan, "dokument" ob'ektida `open()`, `write()`, `close()` usullar mavjud. Ushbu usullar mavjud xujjatning qayta ishlash yoki tarkibini o'zgartirish uchun xizmat qiladi. Oddiy misol keltiramiz:

```
function hello()  
{ id=window.open("", "example", "width=400, height=150");  
id.focus(); id.document.open();  
id.document.write("<H1>Salom!</H1>");  
id.document.write("<HR><FORM>");  
id.document.write("<INPUT TYPE=button VALUE='Oynani yopish '");  
id.document.write("onClick='window.opener.focus();window.close();'>");  
id.document.close();  
}
```

Ushbu misolda `open()` usuli xujjatga yozish imkoniyatini yaratadi, `write()` usuli ushbu yozishni amalga oshiradi, `close()` usuli xujjatga yozishni yopadi. Bularning barchasi oddiy faylga yozgan kabi amalga oshadi. Agar oynada xolat satri mavjud bo'lsa (odatda xujjatning yuklanish darajasi beriladi), xujjatga yozish jarayoni tugallanmagan bo'lsa, xujjat yuklanish vaqtida unda to'g'ri to'rtburchak shaklidagi yozuv davom etayotganligini ifodalovchi belgi "ko'rinadi".

## ***Holatlار***

Usullar va xususiyatlardan tashqari ob'ektlarni holatlar bilan ham xarakterlash mumkin. Shaxsan, JavaScript da dasturlashda ushbu holatlarni qayta ishlovchi vositalar mavjud. Masalan, button tipidagi ob'ekt bilan onClick holati amalga oshishi mumkin, ya'ni foydalanuvchi tugmani bosishi mumkin. Ushbu atribut qiymati sifatida dasturchi tomonidan JavaScript da tuzilgan holatni qayta ishlovchi dastur ko'rsatiladi:

```
<INPUT TYPE=button VALUE="Najat" onClick="window.alert('Salom!');" />
```

Holatlarni qayta ishlash jarayonlari ularning holatlari bilan bog'liq konteynerlarda ko'rsatiladi. Masalan, BODY konteyneri butun xujjatning xususiyatini aniqlaydi, shuning uchun butun xujjatni yopishni qayta ishlovchi holat onLoad atributining qiymati sifatida BODY konteyneri ichida beriladi.

Izoh. Qat'iy aytish mumkinki, har bir brauzer, Internet Explorer, Netscape Navigator yoki Opera da bo'lganidek, o'zining ob'ektki modeliga ega. Turli brauzerlar ob'ektki modellari (hatto turli versiyalari) bir biridan farqlanadi, lekin mantiqiy tarkibi bir hilda bo'ladi.

## ***Kodni HTML-sahifaga joylashtirish***

JavaScript kodini brauzerda bajarilayotganda brauzerdagi JavaScript interpretatori yordamida amalga oshiriladi. JavaScriptni qo'llashda to'rtta funktsional usuldan foydalanish mumkin:

- gipermatnli murojaat (URL sxema);
- holatni qayta ishlovchi (handler);
- podstanovka (entity) (Microsoft Internet Explorer ning 5.X va yuqori versiyalarida mavjud);
- qo'yish yoki o'rnatish (vstavka).

HTML xujjat yaratayotganda JavaScript ning bir nechta usullaridan foydalanish mumkin.

## **3. JavaScript ning URL-sxemasi**

URL (Uniform Resource Locator) sxemasi – bu Web-texnologiyaning asosiy elementlaridan biri hisoblanadi. Web da har bir axborot resursi o'zining unikal URL iga ega. URL A konteynerning HREF atributida, IMG konteynerning SRC atributida, FORM konteynerining ACTION atributida va boshqalarda beriladi. Barcha URL muloqot protokoli turiga qarab turli qismlarga bo'linadi, masalan, FTP-arxivga bog'lanish uchun ftp sxema qo'llaniladi, Gopher-arxivga bog'lanish uchun - gopher sxemadan foydalaniladi, elektron pochta jo'natish uchun - smtp sxemadan foydalaniladi. Sxema turi URL ning birinchi komponentasi orqali aniqlanadi:

<http://intuit.ru/directory/page.html>

Gipermatnli tizimli dasturlash tilining asosiy vazifasi gipermatnli o'tishlarni dasturlashdir. Bu shuni anglatadiki, u yoki bu gipermatnli ssilkaning bosilishi gipermatnli o'tishni amalga oshiruvchi dasturni ishga tushiradi. Web-texnologiyada shunga o'xshash standart dasturlar sahifani yuklash dasturlari hisoblanadi. JavaScript shu standart dasturlarni foydalanuvchi dasturiga aylantiradi. HTTP protokol bo'yicha standart o'tishlardan farqlanish maqsadida JavaScript da alohida URL sxema joriy etilgan:

```
<A HREF="JavaScript:JavaScript_kod">...</A>
```

```
<IMG SRC="JavaScript:JavaScript_kod">
```

Masalan, "Diqqat!!!" nomli gipermatnli ssilka bosilganda ogohlantirish oynasining ochilishi quyidagicha amalga oshiriladi:

```
<A HREF="JavaScript:alert('Vnimanie!!!');"> Diqqat!!!</A>
```



Submit tipidagi tugmani bosish orqali formadagi matn ob'ekti to'ldirilishi quyidagicha amalga oshiriladi:

```
<FORM NAME=myform METHOD=post ACTION="JavaScript:window.document.myform.
myname.VALUE='Siz Click tugmasini bosdingiz';void(0);">
<TABLE BORDER=0 width="100%">
<TR>
<TD><INPUT NAME= "myname" value= "" /></TD>
<TD><INPUT TYPE=submit VALUE=Click></TD>
<TD><INPUT TYPE=reset VALUE=Reset></TD>
</TR>
</TABLE>
</FORM>
```

URL da murakkab dasturlarni joylashtirish va funksiyalarni chaqirish mumkin. JavaScript ning bu sxemasi barcha brauzerlarda ham ishlayvermaydi, Netscape Navigator tipidagi va Internet Explorer ning to'rtinchi versiyasidan boshlab ishlaydi.

### ***Holatlarni qayta ishlovchilar***

Holatni qayta ishlovchi tipidagi (handler) dasturlar, shu holatga aloqador konteyner atributida beriladi. Masalan, tugma bosilgan vaqtda click holati amalga oshadi:

```
<FORM><INPUT TYPE=button VALUE="Tugma" onClick="window.alert('tuit');"> </FORM>
```

### ***Podstanovkalar***

Podstanovkalar (entity) Web-sahifada juda kam uchraydi. Shunga qaramay u HTML-sahifani brauzer tomonida generatsiya qilish uchun vosita hisoblanadi. Podstanovkalar HTML-konteyner atributining qiymati sifatida foydalaniladi. Masalan, standart holat bo'yicha forma ob'ektlari ma'lumotlarini jo'natish uchun adres sifatida joriy sahifa URL adresi ko'rsatiladi:

```
<script>
function myfunction(){
str = window.location.href;
return(str.length);
}</script>
<form><input value="&{window.location.href};" size="&{myfunction()};"
/></form>
<script><!--bu Izoh ...javascript-kod...// --></script>
<body>hujjat tanasi </body>
```

Bilamizki, hujjatning <head> qismidagi matnlar brauzer oynasida ko'rinmaydi. Shuning uchun bu qismga hujjat tanasida chaqiriluvchi va ishlatiluvchi o'zgaruvchilar va funksiyalar joylashtiriladi. Bu sohada Netscape Navigator brauzeri Internet Explorer ga qaraganda biroz qat'iyroq. Agar hujjat tanasidagi funksiya sarlavha qismida e'lon qilinmagan bo'lsa, ushbu funksiya aniqlanmaganligi haqida xabar beradi.

Misol: Funksiyalarni joylashtirish va foydalanish:

```
<html><head>
<script>
function time_scroll(){
```

```

d = new Date();
window.status = d.getHours()+":"+d.getMinutes()+":"+d.getSeconds();
setTimeout('time_scroll();',500);
}
</script>
</head>
<body onLoad= "time_scroll();">
<h1>holat satridagi soat </h1>

```

Internet Explorer 4.0 da podstanovkalar ishlaymaydi, shu bois ulardan foydalanishda ehtiyot bo'lish kerak. Bunda avvalo brauzer turini bilish talab etiladi.

O'rnatish (SCRIPT konteyneri-interpretatori majburiy chaqirish)

SCRIPT konteyneri – bu podstanovka usulining rivojlangan varianti hisoblanadi. Jumladan, SCRIPT odatda Server Side Includes, ya'ni server tomonidagi hujjatlarni generatsiya qiluvchi ham deb ataladi. Interpretator SCRIPT teglari orasidagi barcha qismni generatsiya qiladi va shundan so'ng yana HTML qismga qaytadi.

SCRIPT konteyneri ikkita asosiy funksiyani bajaradi:

- HTML-hujjatga kodni joylashtirish;
- HTML-razmetkalarni brauzer tomonida shartli generatsiyalash.

Birinchii funksiyasi o'zgaruvchilar va funksiyalarni joylashtirish uchun qo'llaniladi. Ikkinchisi - bu xujjatni yuklash yoki qayta yuklash vaqtida JavaScript kod natijasini joylashtirishdir.

### ***HTML-xujjatga kodni joylashtirish***

Shaxsan, bu erda asosiy xilma xillik yo'q. Kod sarlavha konteyneri HEAD orasiga ham, BODY konteyneri orasiga ham joylashtirilishi mumkin. Sarlavha qismida qo'llanilishini ko'rib o'tamiz.

Sarlavha qismida kod SCRIPT konteyneri orasiga joylashtiriladi:

```

<HTML>
<HEAD>
<SCRIPT>
function time_scroll()
{
d = new Date();
window.status = d.getHours()+":"+d.getMinutes()+":"+d.getSeconds();
setTimeout('time_scroll();',500);
}
</SCRIPT>
</HEAD>
<BODY onLoad=time_scroll()>
<CENTER>
<H1>holat satridagi soat </H1>
<FORM>
<INPUT TYPE=button VALUE="Oynani yopish " onClick=window.close()>
</FORM>
</CENTER>
</BODY>
</HTML>

```

Ushbu misolda biz hujjat sarlavhasida time\_scroll() funksiyasini yaratdik va unga BODY (onLoad=time\_scroll()) konteynerining load holatida murojaat qildik.

Quyidagi funksiyani yaratish va chaqirish orqali alohida oyna yaratish mumkin:

```

function sel()
{

```



```

id = window.open("", "example", "width=500,height=200,status,menu");
id.focus();
id.document.open();
id.document.write("<HTML><HEAD>");
id.document.write("<BODY>");
id.document.write("<CENTER>");
id.document.write("<H1>Change text into child window.</H1>");
id.document.write("<FORM NAME=f>");
id.document.write("<INPUT TYPE=text NAME=t SIZE=20 MAXLENGTH=20 VALUE='This
is the test'>");
id.document.write("<INPUT      TYPE=button      VALUE='Close      the      window'
onClick=window.close()></FORM>");
id.document.write("</CENTER>");
id.document.write("</BODY></HTML>");
id.document.close();
}
<INPUT TYPE=button VALUE="holat satrini o'zgartirish"
onClick="id.defaultStatus='Salom'; id.focus();">

```

#### 4. Sinflar ierarxiyasi

Ob'ektga-mo'ljallangan dasturlash tili ob'ektlar daraxtidan tashkil topadi. JavaScript da bu ierarxik daraxt Window ob'ektidan boshlanadi, ya'ni har bir ob'ekt u yoki bu oynada yoziladi. Ixtiyoriy ob'ektga yoki ob'ekt xususiyatiga murojaat qilish uchun undan yuqorida turgan ob'ekt orqali murojaat qilish kerak bo'ladi:

Umuman aytganda, JavaScript klassik ob'ektili til hisoblanmaydi (uni soddalashtirilgan ob'ektili til ham deyish mumkin). Unda merosiylik va nasldorlik mavjud emas. Dasturchi function operatori yordamida o'zining klassini, sinfini ob'ektini yaratishi mumkin, ammo ularni yaratishda odatda standart ob'ektlardan ham foydalanadi. Bu shuni anglatadiki, JavaScript-dasturning amal qilish sohasi joriy sahifa chegarasidan chiqib ketmaydi.

Ba'zan JavaScript ning turli ob'ektlarida bir hil nomli xususiyatlar bo'ladi. Bu holda dasturchi qaysi ob'ekt xususiyatiga murojaat qilayotganini aniq ko'rsatishi kerak. Masalan, Window va documentlarda location xususiyati mavjud. Faqat, Window uchun bu Location sinfi ob'ekti, Document – URL da ko'rsatilib yuklanayotgan hujjatni adresini ifodalaydi.

Ta'kidlash kerakki, ko'pgina ob'ektlarda ob'ekt xususiyatlarini oddiy qiymatga o'zgartiruvchi standart usullar mavjud bo'ladi. Masalan, standart holda barcha ob'ektlar uchun belgilarni satrga aylantiruvchi usul mavjud: toString().

#### Nazorat savollari:

1. Mijoz tomonda dasturlash deganda nimani tushunasiz?
2. JavaScript tili haqida ma'lumot bering?
3. Gipermatnli Web-sahifaning ssenariylari qanday boshqariladi?
4. JavaScript tili ob'ekti nima va qanaqa ob'ektlarni bilasiz?
5. JavaScript tili funksiyasini HTML hujjatda e'lon qiling.
6. JavaScript tili sinflari ierarxiyasini nima?
7. URL sxemasi qanday element hisoblanadi?
8. Qiymatlarini o'zgartirib bo'lmaydigan atributlarga misol keltiring?

## 6-ma'ruza. JavaScriptda ma'lumotlar tiplari va o'zgaruvchilar bilan ishlash.

### Reja:

1. JavaScriptda o'zgaruvchilar;
2. JavaScriptda ma'lumotlar tiplari;
3. JavaScript tili operatorlari;
4. JavaScript tilida funksiya.

**Kalit so'zlar:** o'zgaruvchilar, ma'lumotlar tipi, operatorlar, funksiyalar,

**Ishdan maqsad:** JavaScript tilida ma'lumotlar tiplari va o'zgaruvchilar haqida tushunchaga ega bo'lish va talabalarda ma'lumotlar va o'zgaruvchilar bilan ishlash jarayoni ko'nikmalarini tashkil qilish.

### 1. JavaScriptda o'zgaruvchilar

JavaScript tilida o'zgaruvchilarni ishlatish mumkin va ularni nomlari bilan adreslash mumkin. O'zgaruvchilar globali va lokalli bo'lishi mumkin. Globali o'zgaruvchilar ssenariyning xoxlagan joyida ruxsati bo'lishi mumkin. Lokalli o'zgaruvchilarning xarakati esa e'lon qilingan o'zgaruvchilar ichidagi funksiyalar bilan chegaralangan. Basic dasturlash tili singari JavaScript ssenariysini yarata yotgan vaqtda avvaldan e'lon qilinmagan o'zgaruvchilarni ishlatish mumkin.

#### O'zgaruvchilar e'loni

JavaScript da xamma o'zgaruvchilar var kalit so'zi orqali e'lon qilinadi va quyidagicha ko'rsatilgan:

```
var MyHelloMsg;
```

O'zgaruvchi tipi o'zlashtiriladiki qachonki, unga biror bir qiymat o'zlashtirilsa, quyida avvaldan e'lon qilinmagan matnli qator o'zgaruvchiga yozilmoqda:

```
MyMsg = "Salom!";
```

MyMsg o'zgaruvchi nomi o'zlashtirilgandan so'ng ruxsat beriladi.

O'zgaruvchi nomini tanlaganda, quyidagi oddiy qoidalarini eslab qo'yish kerak:

O'zgaruvchi nomi xarflardan yoki "\_", "\$" belgilardan boshlanish kerak va faqat xarflardan, sonlardan va "\_", "\$" belgilardan iborat bo'lishi kerak;

O'zgaruvchilar nomi JavaScript ning zaxiralangan kalit so'zlari bilar mos kelmasligi kerak.

Quyida JavaScript ning zaxiralangan kalit so'zlar keltirilgan:

<b>break</b>	<b>case</b>	<b>catch</b>	<b>class</b>	<b>continue</b>	<b>const</b>
<b>debugger</b>	<b>default</b>	<b>delete</b>	<b>do</b>	<b>else</b>	<b>enum</b>
<b>export</b>	<b>extends</b>	<b>false</b>	<b>finally</b>	<b>for</b>	<b>function</b>
<b>if</b>	<b>import</b>	<b>in</b>	<b>new</b>	<b>null</b>	<b>return</b>
<b>super</b>	<b>switch</b>	<b>this</b>	<b>throw</b>	<b>true</b>	<b>try</b>
<b>typeof</b>	<b>var</b>	<b>void</b>	<b>while</b>	<b>with</b>	

Bu so'zlar orasida JavaScript tilida va uning rivojlanishida o'zlashtirish rejalashtirilmoqda.

O'zgaruvchining qiymatini o'zlashtirish

"=" o'zlashtirish operatori yordamida o'zgaruvchilar qiymati o'zlashtiriladi. Misol qilib o'uyidagi o'zgaruvchi keltirilgan va unda matnli qator yozilgan:

```
var MyHelloMsg;
```

```
MyHelloMsg = "Hello, world!";
```

MyHelloMsg sonli o'zgaruvchini dasturning xoxlagan joyida o'zlashtirish mumkin, misol uchun:

```
MyHelloMsg = 4;
```

Bu operator bajarilgandan so'ng o'zgaruvchi tipi o'zgaradi, shuningdek interpretatsiya jarayonida brauzer xech qanday ogoxlantiruvchi xabarlarni yubormaydi.

O'zgaruvchini maxsus null qiymati orqali o'zlashtirish mumkin:

```
MyHelloMsg = null;
```

Bunday o'zlashtirish xech qanday tipda o'zgaruvchini belgilamaydi.

## 2. JavaScript da ma'lumotlar tipi

JavaScript tilida bir nechta ma'lumotlar tipi mavjud. Bular sonlar, matnli qatorlar, mantiqiy ma'lumotlar, ob'ektlar, aniqlanmagan tipli ma'lumotlar, xamda maxsus tip null.

### *Sonlar*

JavaScript tili xar xil formatdagi sonlarni ishlatishga ruxsat beradi, bular butun sonlar, suzuvchi nuqtali o'nli formatdagi sonlar va ilmiy notatsiya sonlar. Butun sonlar 8, 10, 16 asosida berilishi mumkin. misol uchun:

25      10 asosidagi butun son

0137    8 asosidagi butun son

0xFF    16 asosidagi butun son

386.7   Suzuvchi o'nli nuqtali son

25e5

Yoki 25E5 Ilmiy notatsiyadagi son, 2500000 ga teng.

Ayrim xollarda "son bo'lmagan" arifmetik funksiyalar kelib chiqishi mumkin. JavaScript da aytilganidek NaN (Not a Number). "Son bo'lmagan" – bu xech qanday songa loyiq bo'lmagan maxsus qiymat. Bu sonlar ustida operatsiya bajarilayotgan vaqtda, va natija son ko'rinishida taqdim etilmagan xollarda paydo bo'ladi. "Son bo'lmagan" qiymatga to'g'ri kelishini isNaN funksiyasi yordamida tekshirish mumkin.

### *Matnli qator*

Matnli qator – bu bir yoki qo'shtirnoq ketma ketlik belgisi, misol uchun:

```
"Hello, world!"
```

""

"12345"

'Bu matnli qator'

''' qatori –bo'shdir. quyidagi 2 ta o'zlashtirish ekvivalent emasligini aniqlaymiz:

MyStr=""

MyStr1=null

Birinchi xolda MyStr o'zgaruvchisida matnli qator saqlanmoqda (bo'sh bo'lsa xam), ikkinchisida esa xech narsa.

### ***Mantiqiy ma'lumotlar***

Mantiqiy ma'lumotlar faqat 2 ta qiymatni, ya'ni True va False ni o'z ichiga oladi. Bu qiymatlar 0 va 1 sonlar bilan bog'liq emas. Bu qiymatlarning asosiy obrazi solishtirish operatsiyasi bajarilayotgan vaqtga qaratilgan, xamda shartli operatsiyalar ishlatilganda xam.

### ***Aniqlanmagan tipli ma'lumotlar***

Agar o'zgaruvchi e'lon qilingan bo'lsa, ammo unga xali qiymat o'zlashtirilmagan bo'lsa, u xolda u aniqlanmagan tipga bo'ladi. Misol uchun quyidagi qatorda aniqlanmagan tipga ega bo'lgan MyVariable o'zgaruvchisi e'lon qilingan: var MyVariable;

Agarda bu o'zgaruvchini null qiymati bilan o'zgartirsak, u xolda o'zgaruvchi tipi o'zgaradi va null qiymatga ega bo'lgan o'zgaruvchiga aylanadi: MyVariable = null;

### ***Ma'lumotlar tipini o'zgartirish***

Agarda ifodalarda xar xil tipli o'zgaruvchilar uchrab qolsa, JavaScript interpretatori avtomatik xolda sonli ma'lumotlarni matnli qatorlarga o'zgartirishi mumkin. Teskari aylantirishni (qatorni-songa) maxsus funksiyalar yordamida, ya'ni parseInt va parseFloat funksiyalari yordamida o'zgartirish mumkin. Buni quyidagi misol orqali ko'rish mumkin:

Misol:

```
<HTML>
<HEAD><TITLE>Type conversion sample</TITLE></HEAD>
<BODY>
<H1>Type conversion sample</H1>
<TABLE>
<SCRIPT LANGUAGE="JavaScript">
var MyTextBuf = "";
MyTextBuf += 4 + " - to'rt soni" + "<BR>";
MyBuf2 = (parseInt("2") + 2) + "&nbsp; - to'rt soni" + "<BR>";
document.write(MyTextBuf + MyBuf2);
</SCRIPT>
</TABLE></BODY></HTML>
```

Bu erda biz MyTextBuf o'zgaruvchisini e'lon qildik va uni bo'sh qator bilan initsializatsiya qildik. Quyida biz bu qatorda 4 sonni summasini va 2 ta matnli qatorni o'zlashtirdik.

```
MyTextBuf = 4 + " - to'rt soni" + "<BR>";
```

Bu ifoda xisoblanayotgan vaqtda 4 qiymat avtomatik xolda matnli qatorga o'zgaradi. Keyingi yig'indi HTML xujjatlarida ishlatiladigan &nbsp; ajralmaydigan probel belgisiga ahamiyat berish lozim. Agarda uni oddiy probelga almashtirsak, qator konkatenatsiyasidan keyin bu probel yo'qoladi. Keyingi qatorda parseInt funksiyasi yordamida matnli qator "2" sonli qatorga o'zgaradi, natijada 2 soni qo'shiladi, undan keyin xar ikkala matnli qatorlarda konkatenatsiya bajariladi:

```
MyBuf2 = (parseInt("2")+2)+" - to'rt soni"+ "<BR>";
```

Natijada MyBuf2 o'zgaruvchisi MyTextBuf.o'zgaruvchisi kabi qatorga ega bo'ladi.

### 3. JavaScript tili operatorlari

#### *Unar operatori*

Unar operatori belgining o'zgarishi uchun to'ldirish operatsiyasini bajarishda, inkrementda xamda dekrementda ishlatiladi:

- teskari xolatda belgining o'zgarishi
- ! Qo'shimcha. Mantiqiy o'zgaruvchilarning qiymatini reversirovaniya qilish uchun ishlatiladi.
- ++ O'zgaruvchi qiymatini oshirish. O'zgaruvchi prefiksi yoki uning suffiksi bo'lib qo'llanishi mumkin.
- O'zgaruvchi qiymatini kamaytirish. O'zgaruvchi prefiksi yoki uning suffiksi bo'lib qo'llanishi mumkin.

Unar operatorini ishlatishga doir misollar:

```
I=0; // i teng 0 dagi o'zgaruvchining boshlang'ich qiymati
I++; // i teng 1 dagi qiymat
--i; // i teng 0 dagi qiymati
var j=3; // j teng 3 dagi o'zgaruvchining qiymati
i = -j; // i teng -3 dagi o'zgaruvchining qiymati
var fYes = true; // fYes teng true dagi o'zgaruvchining qiymati
testFlag(!fYes); // testFlag funksiyasiga false qiymati uzatilmo qda
```

#### *Binar operatori*

Binar operatori 2 ta operandni birlashtiradi. JavaScript tilida binar operatorlari ayirish, bo'lish, qo'shish, ko'paytirish xamda bo'linmani qoldiqini xisoblash uchun ishlatiladi (ko'riladi):

- Ayirish
- + Qo'shish
- \* Ko'paytirish
- / Bo'lish
- % Bo'linmani qoldiqini xisoblash

Bu operatorlar C tilida ishlatilganidek JavaScript da xam xuddi shunday ishlatiladi, misol uchun:

```
I=0; // i teng 0 dagi o'zgaruvchining qiymati
i = i + 1; // i teng 1 dagi qiymat
var j=9; // j teng 9 dagi o'zgaruvchining qiymati
i = j / 2; // i teng 4 dagi o'zgaruvchining qiymati
k = j % 2; // i teng 1 dagi o'zgaruvchining qiymati
```

#### *Aloxida bitlar bilan ishlash operatori*

Ssenariyalarda shunday operatorlar ishlatiladiki, ular aloxida bitlar bilan ishlash operatorlari xisoblanadi, ular quyidagilar: VA , YOKI , MANTIQUIY INKOR, EMAS:

& VA  
| YOKI  
^ MANTIQUIY INKOR  
~ EMAS

### ***Siljuvchi operatorlari***

JavaScript da siljish operatsiyasini bajarish uchun 3 ta operator ko'rilgan:

>> Siljish o'ng tomonga  
<< Siljish chap tomonga  
>>> Bo'shatiladigan razryadlarni nollar bilan to'ldirib o'ngga siljish

### ***Munosobat operatorlari***

Munosobat operatorlari o'zgaruvchilarning qiymatini solishtirish uchun ishlatiladi. Bu operatorlar solishtirish natijalariga bog'liqlik true yoki false mantiqiy qiymatlarni qaytaradi va shartli operatorlarda asosiy bo'lib ishlatiladi. True qiymatini qaytaradigan JavaScript tilining munosobat operatorlari ko'rsatilgan:

> Chap operand o'ng operanddan katta  
>= Chap operand o'ng operanddan katta yoki teng  
< Chap operand o'ng operanddan kichik  
<= Chap operand o'ng operanddan kichik yoki teng  
== Chap operand o'ng operandga teng  
!= Chap operand o'ng operandga teng emas

### ***Mantiqiy operatorlar***

|| YOKI operatori. True qiymat qaytaradi, qachonki operandlardan biri true bo'lsa.  
&& VA operatori. True qiymat qaytaradi, qachonki ikki operand true bo'lsa

### ***O'zlashtirish operatori***

O'zlashtirish operatori o'zgaruvchilarning qiymatini o'zlashtirish uchun ishlatiladi. JavaScript tilida va S dasturlash tilidagi kabi bu operator boshqa operatorlar bilan kombinatsiyasiga ruxsat etiladi. quyida o'zlashtirish operatorini boshqa operatorlar bilan kombinatsiyasi berilgan:

= Oddiy o'zlashtirish  
+= Sonli qiymatni kattalashtirish yoki qatorlarni ?o'shilishi  
-= Sonli qiymatni kichiklashtirish  
\*= Ko'paytirish  
/= Bo'lish  
%= Bo'lishdan qolgan qoldiqni xisoblash  
>>= O'ngga siljish  
>>>= Bo'shatiladigan razryadlarni nollar bilan to'ldirib o'ngga siljish  
<<= Chapga siljish  
|= YOKI  
&= I  
^= MANTIQUIY INKOR

C tili bilan tanish bo'lmaganlar uchun o'zlashtirish operatorini boshqa operatorlar bilan birgalikda ishlatilishi qiyinroq va g'ayrioddiy tuyilishi mumkin, lekin aslida ssenariyni osonlashtiradi boshlang'ich tekstni soddalashtiradi.

Masalan sonli o'zgaruvchilar qiymatini oshirish uchun += operatori ishlatiladi. Avval bu vazifani echimini += operatorini ishlatmagan xolatda ko'rib chiqamiz. Quyida nCounter o'zgaruvchisi e'lon qilindi va unga boshlang'ich 1 qiymati o'zlashtirildi, so'ng bu qiymat 5 ga oshirildi:

```
var nCounter = 1;
nCounter = nCounter + 5;
```

Endi buni += operatori yordamida bajaramiz:

```
var nCounter = 1;
nCounter += 5;
```

Ko'rinib turibdiki 2-usul 1-usulga nisbatan qisqa.

O'zgaruvchi qiymatini 3 razryadga o'ngga siljitish uchun >>= operatoridan foydalanish mumkin va u quyidagi matnda ko'rsatilgan:

```
nCounter >>= 3;
```

Natija esa quyidagi matnda ko'rsatilganidek bo'ladi:

```
nCounter = nCounter >> 3;
```

#### 4. JavaScript tilida funksiya

Boshlang'ich matn bo'lagini funksiya ko'rinishida yozish mumkin va ularni JavaScript ssenariysining turli joylaridan murojaat qilish mumkin. Odatda funksiyalar HTML dokumentini sarlavxa bo'limida aniqlanadi. Funksiyalar chaqirilishidan avval e'lon qilinishi kerak va barcha funksiya e'loni HTML dokument sarlavxasida joylashtirilgan bo'lishi kerak.

Funksiyaning umumiy e'loni quyida keltirilgan:

```
function nom([parametr 1] [,parametr 2] [...,parametr N])
{
    . . .
    Funksiya matni qatorlari
    . . .
    [return qiymat]}
```

Barcha parametrlar funksiyaga qiymatiga beriladi. Shuning uchun funksiya unga parametr sifatida beriladigan o'zgaruvchilar qiymatini o'zgartira olmaydi.

Return kalit so'zi yordamida funksiya qiymati qaytariladi.

#### Nazorat savollari:

1. O'zgaruvchilar nima va ularni vazifasi nimalardan iborat?
2. O'zgaruvchi tipini tushuntirib bering?
3. Qanday JavaScript tili operatorlarini bilasiz?
4. JavaScript tili funksiyasi nima va uning vazifasi?
5. JavaScript tili yordamida qanday ma'lumotlar bilan ishlash mumkin?
6. O'zgaruvchi nomini tanlaganda, qanday qoidalarga amal qilinadi?
7. Binar va unar operatorlari orasidagi farqni tushuntiring?
8. Unar operatorlarini ishlatishga doir misollar keltiring?
9. Binar operatorlarini ishlatishga doir misollar keltiring?
10. Funksiyalar qaysi bo'limda aniqlanishi kerak?

## 7-ma'ruza. JavaScript dasturlarida jarayonlarni boshqarish elementlari.

Reja:

1. JavaScript tilida shart operatorlari;
2. Boshqaruv o'tkazuvchi operatorlar;
3. Sikllar;
4. JavaScript tilida Switch operatori.

**Kalit so'zlar:** shartli va boshqaruv o'tkazuvchi operatorlar, sikl operatorlari, switch operatori.

**Ishdan maqsad:** JavaScript tilida dasturlash jarayonida shartli va boshqaruv o'tkazuvchi operatorlar, sikl operatorlari, switch operatori bilan ishlash ko'nikma va bilimiga ega bo'lish.

### 1. JavaScript tilida shart operatorlari

JavaScript tilida bir necha shart operatorlari bor. Hozir ulardan bazilari bilan tanishib chiqamiz.

#### *JavaScript tilida if operatori*

Bu operator JavaScript dasturlash tilidagi muhim operatorlardan biridir. U shartga bog'liq ravishda kod fragmentini bajarishga mo'ljallangan. if operatorining strukturasini quyidagicha ifodalash mumkin:

**1-variant.** if operatori sintaksisi:

```
if(shart)
{
  Amallar ketma ketligi
}
```

**2-variant.** if operatori sintaksisi:

```
if(shart)
{
  Amallar ketma ketligi 1
}
else
{
  Amallar ketma ketligi 2
}
```

Bu erda shart JavaScript tilidagi mantiqiy yoki ixtiyoriy shartdir. Agarda shart mantiqiy tipdagi o'zgaruvchi bo'lsa, qiymati rost (True) bo'lsa, u holda 1-amallar ketma ketligi bajariladi. Aks holda 2-amallar ketma ketligi bajariladi.

Quyida *if* shart operatoriga doir misol ko'rib o'tamiz:

```
function checkData()
{
  if (document.form1.threeChar.value.length==3)
  {return true;
  }
  else
  { alert('3 raqamini kiriting');
  return false;
  }
}
```



JavaScript tilida if operatori obraz kabilardan foydalanadi. If buyrug'i bo'yicha standart operatorga misol keltiramiz:

```
if (a == 1) window.alert("1 topildi!");
```

if operatori bir necha harakat operatorlaridan ham foydalanadi:

```
1: if (a == 1) {  
2: window.alert("1 topildi!");  
3: a = 0;  
4: }
```

Bu operatorlar bloki a o'zgaruvchi ustida tekshiruv o'tkazadi va agar u 1 ga teng bo'lsa, u holda habar aks ettiriladi va a o'zgaruvchi yangi qiymat 0 ga aniqlanadi.

JavaScript tilidagi shart operatorlarida quyidagilardan foydalaniladi:

- == (teng)
- != (teng emas)
- < (kichik)
- > (katta)
- <= (kichik yoki teng)
- => (katta yoki teng)

### ***Java Script tilida else operatori***

Biz yuqorida faqat if operatorining asosiy qisminigina ko'rdik. Bu operatorning bir nechta kengaygan shakli mavjud. else operatori if operatorida tekshirilayotgan ifoda noto'g'ri bo'lgan holatdagina kengaytiradi hamda bu holatda yangi shartda biror amal bajaradi.

else operatori yordamida kengaytirilgan if operatorining strukturasini quyidagicha ifodalash mumkin:

```
if (shartli ifoda) bajariladigan_blok1  
else bajariladigan_blok2
```

Bu if..else konstruktsiyasi quyidagicha interpretatsiya qilinishi mumkin: agar shart bajarilsa (ya'ni ifoda=true), u holda bajariladigan\_blok1 dagi amallar bajariladi, aks holda bajariladigan\_blok2dagi amallar bajariladi. else operatoridan foydalanish majburiy emas.

Quyida else operatoriga doir misol ko'rib o'tamiz:

```
1: if (a == 1) {  
2: alert("1 topildi");  
3: a = 0;  
4: }  
5: else {  
6: alert("Noto'g'ri belgi: " = a);  
1: }
```

### ***JavaScript tilida elseif operatori***

if shart operatorining yana bir kengaygan shakli – bu elseif operatorining qo'llanilishidir. elseif – bu else hamda if operatorlarining kombinatsiyasidir. U xuddi else operatori kabi if operatorida shart bajarilmagan holda kengaytiradi. Biroq else operatoridan farqi bir-biriga zid amallarni faqat agarda elseif shart rost bo'lgandagina bajaradi. else hamda elseif operatorlari yordamida kengaytirilgan if operatorining strukturasini quyidagicha ifodalash mumkin:

```
if (ifoda1) bajariladigan_blok1
elseif (ifoda2) bajariladigan_blok2
else bajariladigan_blokN
```

elseif operatorlari bitta if-blokida bir necha marta uchrashi mumkin. elseif tasdig'i faqat oldinda turgan if-shartlari hamda elseif-shartlari False qiymatni, berilgan elseif-sharti esa True qiymatni qaytargandagina bajariladi.

Quyida elseif operatoriga doir misol ko'rib o'tamiz:

```
function makeMinutes() {
    var minstring="";
    var now = new Date();
    var min = Date.getMinutes();
    if (min<10) {
        minstring+=":0"+min;}
    elseif(min>10){
        minstring+=":"+min;}
    return minstring;
}
```

## 2. Boshqaruv o'tkazuvchi operatorlar

Ba'zida sikl yoki uning alohida iteratsiya ishini tezda to'xtatishga to'g'ri keladi. Buning uchun **break** hamda **continue** boshqaruv o'tkazuvchi operatorlari kerak bo'ladi.

### **JavaScript tilida break operatori**

Break operatori mavjud siklni amalga oshirishni tugallaydi, for, while, do while yoki switch break strukturani boshqaruvchi, sikl yoki shartni tekshirishni tugallash kerakligini bildiruvchi, uning tarkibiga kiruvchi raqamli argument bilan qo'llaniladi.

Break operatoriga doir bo'lgan misolni ko'rib chiqamiz:

```
for (i = 0; i < a.length; i++) {
    if (a[i] = theValue)
        break;
}
```

Ushbu skriptda a massiv elementi theValue o'zgaruvchi qiymatiga teng bo'lganda sikl to'xtatiladi.

### **JavaScript tilida continue boshqaruv o'tkazuvchi operatori**

Ba'zan sikl ishini butunlay to'xtatish lozim bo'lmaydi, faqatgina uning yangi iteratsiyasini boshlash kerak. Continue operatori istalgan siklni amalga oshirish blokidan keyingi instruktsiyalarni o'tkazib yuborish va yangi doira bilan amalga oshirishni davom ettirish imkonini beradi. continue ni uning tarkibida boshqariluvchi konstruktsiyalar ishini yakunlash kerakligini ko'rsatuvchi raqamli argument tarzida ishlatish mumkin.

Oldingi paragrafda berilgan misoldagi break operatorini continue ga almashtiramiz. Bundan tashqari to'rt sikli miqdorini kamaytiramiz.

### **Misol:**

```
i = 0;
n = 0;
while (i < 5) {
    i++;
```

```

    if (i ++ 3)
        continue;
    n ++ i;
}

```

Bu skriptda agarda i uzgaruvchi i=3 qiymatga ega bo'lsa, u xolda n=(1,3,7,12) qiymatlarga ega bo'ladi.

### Misol:

```

Check i and j :
while (i<4) {
    document.write(i + "<BR>");
    i++;
    checkj :
    while (j>4) {
        document.write(j + "<BR>");
        j-=1;
        if ((j%2)++0)
            continue checkj;
        document.write(j + " is odd.<BR>");
    }
    document.write("i = " + i + "<br>");
    document.write("j = " + j + "<br>");
}

```

Ushbu siklda agarda continue operatori topilsa, check j sikli to'xtatiladi va checkj ning keyingi sikliga o'tiladi. Xar safar continue topilganda checkj iteratsiyani boshidan boshlaydi, toki uning sharti false bo'lmaguncha. checkj sharti false bo'lganda, checkiandj ning qolgan operatorlari bajariladi va bu sikl xam checkiandj sharti false bo'lguniga qadar davom etadi.

## 3. Sikllar

JavaScript tilida shartga bog'liq ravishda qaytariladigan amallardan iborat bir nechta konstruktiviyalar mavjud. Bu while, do..while, foreach hamda for sikllardir. Ularni batafsil ko'rib chiqamiz.

### *JavaScript tilida While sikli*

While siklining strukturasi quyidagicha bo'ladi:

```
while (ifoda) { bajariladigan_blok }
```

yoki

```
while (ifoda): bajariladigan_blok endwhile;
```

while – bu oddiy sikl. U ifoda qiymati True (bu erda xuddi if operatori kabi ifoda mantiqiy tipga o'zlashtiriladi) bo'lgunicha bajariladigan\_blokdagi buyruqlarni bajarishga buyuradi. Ifoda qiymati har sikl boshlanganda tekshirib boriladi, agarda uning qiymati bajariladigan\_blok bajarilish jarayonida o'zgargan taqdirda ham iteratsiya tugamaguncha (ya'ni bajariladigan\_blokdagi barcha buyruqlar bajarilmaguncha) sikl to'htatilmaydi.

While operatoriga doir misollar ko'rib o'tamiz:

```
i = 0
```

```

n = 0
while (i<5) {
  i ++;
  if (i ++ 3)
    continue n + = i
}

```

### **Misol:**

```

n = 0;
x = 0;
while( n < 3 ) {
  n ++;
  x += n;
}

```

### **JavaScript tilida do... while sikli**

do..while sikli while siklga o'xshaydi, ammo farqli tomoni shundaki, ifodaning rostligiga sikl boshida emas, balki oxirida tekshiriladi. Qulay tomoni shundaki, bajariladigan\_blok do..while sikli ichida hech bo'lmaganda bir marta bajariladi.

Strukturasi quyidagicha:

```

do { do..while sikli } while (ifoda);
do..while operatori asosidagi misolni ko'rib o'tamiz:

```

```

do {
  i+=1;
  document.write(i);
} while (i<5);

```

### **JavaScript tilida for sikli**

Bu JavaScript tili sikl operatorlaridan biri hisoblanadi. Ular C dasturlash tilidagi sikllar kabidir.

Strukturasi quyidagicha:

```

for (ifoda1; ifoda2; ifoda3) { bajariladigan_blok }
yoki
for (ifoda1; ifoda2; ifoda3): bajariladigan_blok endfor;

```

Bu erda ko'rinib turibdiki shart uchta ifodadan tashkil topadi. Birinchi ifoda1 ifoda sikl boshida shartsiz bajariladi. Har bir iteratsiyaning boshlanishida ifoda2 bajariladi. Agar u True qiymatni qabul qilsa, u holda sikl o'z ishini davom ettiradi va bajariladigan\_blokdagi barcha buyruqlarni bajaradi. Agar ifoda2 False qiymatni qabul hilsa, u holda sikl to'xtatiladi. har bir iteratsiya (ya'ni bajariladigan\_blokdagi barcha buyruqlarni bajarilishidan keyin) oxirida ifoda3 bajariladi.

Har bir 1-,2- va 3-ifodalar bo'sh bo'lishi mumkin. Agar ifoda2 bo'sh bo'lsa, bu siklni cheksiz bajarilishini bildiradi. Bu unchalik befoyda emas, chunki siklni break operatori yordamida to'xtatsa bo'ladi.

**Misol:** Formada tanlangan elementlar sonini ekranga chiqarish;

```

<SCRIPT>
function howMany(selectObject) {
  var numberSelected=0;
  for (var i=0; i < selectObject.options.length; i++) {
    if (selectObject.options[i].selected==true)

```

```

        numberSelected++;
    }
    return numberSelected;
}

</SCRIPT>
<FORM NAME="selectForm">
<P><B>Choose some music types, then click the button below:</B>
<BR><SELECT NAME="musicTypes" MULTIPLE>
<OPTION SELECTED> R&B
<OPTION> Jazz
<OPTION> Blues
<OPTION> New Age
<OPTION> Classical
<OPTION> Opera
</SELECT>
<P><INPUT TYPE="button" VALUE="How many are selected?"
onClick="alert('Number of options selected: '+howMany(document.selectForm.
musicTypes))">
</FORM>

```

Agarda ushbu operator ichidagi barcha uchala ifoda ham tushirib qoldirilsa, u holda schyotchik var i o'zgaruvchini boshlang'ich qiymati berilmaydi va har bir sikl oxirida u o'zgarmaydi. Bu barcha buyruqlarni alohida buyruqlar ko'rinishida yoki sikldan avval bajariladigan\_blok ichida yozsa ham bo'ladi:

```

var i=0; // hisoblagichni boshlang'ich qiymatini beramiz
for ( ; ; ){
    if (i>=10) break;
    // agar i katta yoki teng 10 bo'lsa, u holda sikl ishini to'xtatamiz.
    if (i % 2 == 0) print $i;
    // agar son juft bo'lsa, uni ekranga chiqaramiz.
    i++; // schyotchik qiymatini bittaga oshiramiz.
}

```

for sikli konstruktsiyasidagi uchinchi ifodada verguldan keyin yana bir nechta oddiy buyruqlarni ham yozsa bo'ladi. Masalan, agar biz oddiygina barcha sonlarni ekranga chiqarmoqchi bo'lsak, dasturni quyidagicha yozsa bo'ladi:

```

for (i=0; i<10; document.write(i), i++)
/* Agarda bajariladigan_blok buyruqlardan tashkil topmagan
yoki bitta buyruqdan tashkil topgan bo'lsa,
figurali qavsga olingan qismni
tashlab ketsa bo'ladi.*/

```

#### 4. JavaScript tilida Switch operatori

Yana bir shartni tekshirib turli amallarga bog'liq ravishda ish ko'rsatadigan konstruktsiya bu – switch operatoridir. Bu operatorni o'zbek tiliga tarjima qilinganda “yo'nalishni o'zgartirgich” ma'nosini beradi hamda bu operatorning vazifasi ham shunga o'xshashdir.

Switch operatori bir nechta asosiy elementlardan tashkil topgan:

- Switch boshlanish operatori. Bu operator qavs ichida kiritilgan belgilarni taqqoslaydi.
- Figurali qavslar {}. If operatoridagi qavslar kabi vazifani bajaradi.
- Bir yoki bir necha case operatori.
- Break kalit so'zi. Case operatoridan chiqish harakatini aniqlash uchun ishlatiladi.

O'zgaruvchini qanday qiymatni qabul qilishiga bog'liq ravishda u yo'nalishni o'zgartirib turli bloklardagi amallarni bajaradi. switch operatori if...elseif...else yoki if operatori majmuiga juda o'xshash bo'ladi. switch operatorining strukturasini quyidagicha ifodalash mumkin:

```
switch (ifoda yoki o'zgaruvchi){
  case qiymat1:
    amallar_bloki1
  break;
  case qiymat2:
    amallar_bloki2
  break;
  ...
  default:
    amallar_bloki_avtomatik_tarzda
}
```

if operatoridan farqli tomoni bu erda ifodalar mantiqiy tip qabul qilmay, balki faqat case kalit so'zidan keyingi qiymatlarni (qiymat1, qiymat2 va h.k.) taqqoslaydi xolos. Agar ifoda qiymati qandaydir variant bilan ustma-ust tushsa, ikki nuqtadan keyingi break operatorigacha bo'lgan amallar\_blokidagi amallarni bajaradi. Agarda ifoda qiymati berilgan variantlardan hech biriga ustma-ust tushmasa, default kalit so'zidan keyingi avtomatik tarzda bajariladigan blok (amallar\_bloki\_avtomatik\_tarzda) bajariladi. switch operatoridagi ifoda faqat bir marta hisoblanadi, elseif operatorida esa har bir tekshirishda hisoblanadi, shuning uchun agarda ifoda etarli darajada murakkab bo'lsa, u holda switch operatori tezroq ishlaydi.

Quyida switch operatoriga doir misollar ko'rib o'tamiz:

```
function getName(){
  var names = array("Name1", " Name2", " Name3");
  var selected="";
  switch (names[0]){
    case "Name1": selected="Name1 is selected";
    break;
    case "Name2": selected="Name2 is selected";
    break;
    case "Name3": selected="Name3 is selected";
    break;
    default: selected="Default select"=$names[0];
  }
  return selected;
}
```

Yuqoridagi misoldan ko'rinib turibdiki, getName() funksiyasi ishga tushirilganda, names massividagi birinchi element tekshiriladi. Bu xolda massiv 1-elementi Name1 bo'lgani uchun bizga selected="Name1 is selected"; qiymatni qaytaradi.

## 2-misol.

```
var change =prompt("Xarakatni tanlang:/n1 - Mashina sotib olish/n2 - Mashina sotish /n3 - Mashina almashtirish ");
switch (change) {
  case "1": {
    document.write("Siz mashina sotib olishni xoxlaysiz ");
    break;
  }
}
```

```

case "2": {
    document.write("Siz mashina sotishni xoxlaysiz ");
    break;
}
case "3": {
    document.write("Siz mashina almashtirishni xoxlaysiz ");
    break;
}
default: {
    document.write("Siz noto'g'ri buyruq kiritdingiz ");
    break;
}
}

```

switch operatorining konstruktiviyasi uchun xuddi if operatori kabi alternativ sintaksisi mavjud. Bu erda switch operatoridagi ochiladigan figurali qavs ikki nuqtaga o'zgartiriladi, yopiladigani esa mos ravishda **endswitch** kalit so'ziga o'zgartiriladi.

### Nazorat savollari:

1. JavaScript tilida shart operatorlarining ishlashi va vazifasini tushuntirib bering.
2. Boshqaruv o'tkazuvchi operatorlar nima va ularning vazifasi nimalarda namoyon bo'ladi?
3. Qanaqa sikl operatorlarini bilasiz va ular qanday ishlaydi?
4. Switch operatori ishlash prinsipini tushuntirib bering?
5. Turli shart operatorlarining bir-biridan farqli tomonlarini keltiring?
6. Qay holatda qaysi shart operatoridan foydalanish qulay deb o'ylaysiz?
7. Har bir operatorni tegishli misollar yordamida tushuntiring?
8. Switch operatoridan foydalanish qanday afzalliklarga ega?
9. JavaScript tilidagi operatorlar va sikllarning boshqa dasturlash tillaridagi operatorlardan farqli jihatlari bormi?
10. Boshqaruv o'tkazuvchi operatorlarning ishlash prinsipini ko'rsating?

## 8-ma'ruza. JavaScript tilida funksiya va usullar tushunchasi. Ob'yektlar va kattaliklar. JavaScriptda hujjatning obyektli modeli.

### Hususiyyatlar va usullar.

#### Reja:

1. JavaScript tilida funksiyalar;
2. Global klassi;
3. Math klassi;
4. Date klassi;
5. JavaScript tilidagi ob'ektlar.

**Kalit so'zlar:** JavaScript tilidagi funksiyalar, Global, Math, Date sinflari, ob'ektlar, kattaliklar, sinflar, usullar, ob'yektli model.

**Ishdan maqsad:** Talabalarda JavaScript tilidagi funksiya va usullar, ob'yektlar va kattaliklar tushunchasi haqida ko'nikma va bilim hosil qilish, dasturlash jarayonida ulardan foydalanishni o'rgatish, hujjatning ob'yektli modeli va hususiyyatlar haqida bilimga ega bo'lish.

## 1. JavaScript tilida funksiyalar

JavaScript ssenariyli tili ob'ektga-mo'ljallangan tildir. JavaScript ob'ektlari xususiyatlar va usullar to'plamini ifodalaydi. Ob'ekt xususiyati – bu, ob'ektga bog'liq bo'lgan ma'lumotlardir, usullar esa - ob'ekt ma'lumotlarini qayta ishlovchi funksiyalardir. JavaScript ssenariyda xususiyatlarni adreslash ularning nomlari bilan yoki ularning nomlari bilan amalga oshishi mumkin. Keyingi variant bo'yicha, har bir xususiyat massivning bir elementi sifatida olinadi va ular o'zlarining unikal nomlariga ega bo'ladilar.

JavaScript tilida C va Java dasturlash tillaridagi kabi protsedura va funksiyalar mavjud bo'lib, ular quyidagicha e'lon qilinadi:

```
function kalit so'zi;  
funksiya nomi;  
    vergul va qavs bilan ajratilgan argumentlar ro'yxati;  
figurali qavs ichiga olingan funksiya tanasi.  
function myFunction(arg1, arg2, ...)  
{  
    ...  
Operatorlar ketma-ketligi  
    ...  
}
```

Bu erda:

myFunction – funksiya nomi, arg1, arg2 – parametrlar.

**Misol:**

```
function Factorial(n) {  
    if((n<0)||((round(n)!=n))) {  
        alert("Factorial funksiyasi ushbu argumentda aniqlanmadi "+n);  
        return NaN;  
    } else {  
        Result=(n*Factorial(n-1));  
        return result;  
    } }  
}
```

Funksiyada return kalit so'zi orqali qiymat qaytarilmasligi ham mumkin.

**Misol:**

```
function Greeting(s) {  
    document.write("Hello,"+s+"!");  
    return ;  
}
```

Funksiyani chaqirish aniq parametrlar bilan chaqiriladi:

**Misol:**

Factorial(3); - bu funksiya natijasi 6 ga teng,  
Greeting("world"); - bu funksiya ekranga "Hello, world!" satrini chiqaradi.

Har bir funksiya, masalan, myFunction funksiyasi myFunction nomli ob'ekt xisoblanadi, agarda argumentlar arguments nomi bilan berilsa, unga murojaat quyidagicha:



`myFunction.arguments[i]`, bu erda `i` — argumenta nomeri (rahamlash 0 dan boshlanadi).

Funksiya e'lonida aniq parametrlar formal parametrlarga teng eki ko'p sonda bo'lishi lozim. Bunda funksiya ishga tushirilganda jo'natilayotgan argumentlar miqdori `myFunction.arguments.length` maydoni yordamida aniqlanadi va ushbu maydondagi qiymatni qayta o'zlashtirishni dinamik o'zgartirish mumkin.

### Misol:

Ekranga HTML formatidagi ro'yoxatni chiqarish.

Bu erda (`ListType`) ning birinchi argumenti tartiblanmagan ro'yxat uchun "o" eki "O", tartiblanmagan ro'yxat uchun "u" eki "U" bo'lishi mumkin.

```
function myList(ListType)
{
document.write("<" + ListType + "L");
for(var i=1; i < myList.arguments.length; i=i+1) {
document.write("<LI>" + myList.arguments[i]);
}
document.write("</" + ListType + "L>");
}
```

HTML hujjatida funksiyaga murojaat quyidagicha amalgam oshiriladi:

```
<script>
myList("o", "matn", 2, "3")
</script>
```

Natija:

matn  
2  
3

## 2. Global klassi

Ushbu klass JavaScript ning funksional qismi bo'lib, bu klass bir ob'ektda bir nechta usul va xossalarni birlashtirish vazifasini bajaradi. Usulga murojaat qilinganda ob'ekt ko'rsatilmaydi, aniqrog'i bu usul konstruktorga ega bo'lmaydi. Bunday xossa va usullarga quyidagilarni keltirish mumkin:

Hossa	Mazmuni
Nan	NaN (Not A Number)
Infinity	Number.POSITIVE_INFINITY qiymatni o'z ichiga oladi.

Usul	Mazmuni
escape	Qatorni barcha platformalarga mos holda tasvirlash
eval	JavaScript tili funksiyasi yoki usullarini uzatish
isFinite	Argumentning oxirgi raqamliligini aniqlash
isNaN	Argumentning raqam yoki raqam emasligini aniqlash
parseFloat	Qatorni ko'chib yuruvchi nuqtali son ko'rinishida tasvirlash
parseInt	Qatorni butun songa aylantirish
unescape	Escape funksiyasi natijasini qaytarish

e  
va  
l(  
s)  
fun  
ksi  
yas

i - s qatorni JavaScript operatorlari ketma-ketligi ko'rinishida tasvirlash.

`getClass(Jobj)` funksiyasi – `JavaObject` tipidagi argument uchun `JavaClass` ob'ektini qaytaradi.

#### **Misol:**

```
var myJavaRClassqnew java.awt.Rectangle()  
var myJavaRClass=getClass(myJavaRect)  
getClass() Java-metodi bilan adashtirmang:  
var myJavaRObject=myJavaRect.getClass() - bu java.awt.Rectangle  
klasining Java tilidagi realizatsiya holati.  
isNaN(x) funksiyasi - x "Not a Number", ya'ni son emasligini tekshirish.  
parseFloat(s) funksiyasi - Float tipidagi s rahamni aniqlash. Agar son  
topilmasa u holda NaN ("Not a Number") qiymati qaytariladi.  
parseInt(s) - Integer tipi uchun yuqoridagi holat.
```

#### ***eval(s) funksiyasi***

`eval(s)` funksiyasi – JavaScript ning ichki funksiyasi hisoblanadi. Ushbu funksiya bir yoki bir nechta JavaScript operatorlaridan iborat bo'lgan `s` satrni argument tomonidan uzatilgan kodni bajaradi. Bunda `s` satridagi operatorlar nuqtali vergul yordamida ajratiladi. Bu funksiya nafaqat operatorni bajarish, balki biror amallarni hisoblash imkonini ham beradi. Bunda u kodda keltirilgan amal xisobining oxirgi qiymatini qaytaradi.

#### ***isNaN(x) funksiyasi***

Bu funksiya `x` argumentning "son emas" ligini tekshiradi. Natija NaN qiymatga ega emasligini, ya'ni mumkin bo'lmagan son (masalan, nolni nolga bo'lish natijasi) ni tekshiradi. Ushbu funksiya JavaScript da literal ko'rinishda NaN qiymatni berish mumkin emasligi uchun muhimdir. Bundan tashqari `parseFloat(s)` va `parseInt(s)` funksiyalar natijalarini tekshirish (mumkin bo'lgan son ekanligini) va arifmetik xatolar mavjudligi, masalan, nol soniga bo'lish mavjudligini tekshiradi.

#### ***parseFloat(s) funksiyasi***

`s` satrini sintaktik analiz qilish va dastlab raqamni qaytarish (satrni raqamga aylantiradi). `parseFloat(s)` da `s` satrida ruxsat etilmagan raham elementlari (masalan, belgilar, raham, o'nli vergullar, daraja ko'rsatkichi va hokazo) mavjud bo'lsa analiz to'xtatiladi va qiymat qaytariladi. Agarda `s` satrda son bilan boshlanmasa, u holda `parseFloat(s)` funksiya NaN qiymatni qaytaradi.

#### ***parseInt(s) funksiyasi***

Bu funksiya satrni butun songa aylantiradi. `parseInt(s)` funksiyadagi `s` satrda hisoblash tizimida ko'rsatilmagan qiymatlarga ega bo'lganda sintaktik analiz to'xtatiladi va qiymat qaytariladi. Odatda, `parseFloat` va `parseInt` funksiyalar `s` satr son bilan boshlanmaganda NaN qiymat qaytaradi.

`parseInt(s,n)` holatida `n` asos hisoblanib, agarda `n=10` bo'lsa, `parseInt(s)` funksiya satrdagi 10 lik sanoq sistemasidagi sonlarni tekshiradi. `N=8` bo'lsa, 8 lik sanoq tizimidagi sonlar mavjudligini (bunda `n` 0 dan 7 gacha bo'lgan sonlar qiymatiga ega bo'lishi mumkin). `N=16` bo'lsa, 16 lik sanoq tizimidagi sonlar mavjudligini (bunda 0 dan 9 gacha bo'lgan sonlar va A dan F gacha bo'lgan harflar qiymatiga ega bo'linadi). Agarda `n=0` bo'lsa yoki qiymat berilmasa, u holda `parseInt(s)` funksiya satrning o'zidan asosni aniqlaydi. Bu holatda agarda satr 0x bilan boshlansa, unda funksiya satrning qolgan qismini 16 lik sanoq tizimidagi son sifatida analiz qiladi, agarda satr 0 dan boshlansa, satr 8 lik sanoq tizimidagi qiymat sifatida analiz qilinadi.

### **3. Math klassi**

Math – konstantalar va metodlardan iborat klassdir. Ular ob'ekt uchun odatdagidek murojaat qilinadi:

`Math.konstanta`

`Math.funksiya(i..)`

### ***Math klassi konstantalari***

- E – e soni (natural logorifm asosli)
- LN10 — 10 li natural logorifm (ln10 soni)
- LN2 — 2 li natural logorifm (ln2 soni)
- LOG10E — 10 asosli e lagorifm (log10e soni)
- LOG2E — 2 asosli e lagorifm (log2e soni)
- PI — p konstantasi ("pi" soni)
- SQRT1\_2 — 2 ning teskari kvadrat ildizi ( $1/\sqrt{2}$ )
- SQRT2 — 2 ning kvadrat ildizi ( $\sqrt{2}$ )

### ***Math klassi metodlari***

- abs(x) (x-son yoki ifoda) – absolyut qiymatni hisoblash;
- acos(x) (x bu erda [-1.0;1.0] radian intervaldagi son yoki ifoda) – arkkosinusni hisoblash qaytariladigan qiymat 0 dan pi radian oralig'ida bo'ladi.
- asin(x) (x bu erda [-1.0;1.0] radian intervaldagi son yoki ifoda) – arksinusni hisoblash qaytariladigan qiymat  $-\pi/2$  dan  $\pi/2$  radian oralig'ida bo'ladi.
- atan (x) (x – son yoki ifoda) – arktangensni radianlarda hisoblash. qaytariladigan qiymat  $-\pi/2$  dan  $\pi/2$  radian oralig'ida bo'ladi.
- atan2(x,y)(x,y — to'g'ri burchakli kordinata sistemasi kordinata nuqtalari) – Qutb kordinatasida (x,y) nuqtalar burchagini hisoblaydi. Qiymati 0 dan 2pi radian oralig'ida bo'ladi.
- ceil(x) (x — son yoki sonli ifoda) – sonni butun songa yo'naltirilgan holda yaxlitlash. Manfiy sonlar 0 soni yo'nalishiga qarab yaxlitlanadi.
- cos(x) (x – radiandagi burchak) – kosinusni hisoblash, qaytariladigan qiymat -1.0 dan 1.0 radian oralig'ida bo'ladi.
- sin(x) (x – radiandagi burchak) – kosinusni hisoblash, qaytariladigan qiymat -1.0 dan 1.0 radian oralig'ida bo'ladi.
- Exp(x) (x — son yoki sonli ifoda) — e eksponentsini hisoblash.
- Floor(x) (x — son yoki sonli ifoda) – sonni butun qismiga yo'naltirib yaxlitlash, masalan, floor(-1,1) teng (-2); floor(1,1) teng 1.
- Log(x) (x — musbat son yoki ifoda) – natural lagorifmni hisoblash.
- max(a,b) (a,b — son yoki ifoda) – ikki qiymatdan kattasini qaytaradi.
- min(a,b) (a,b — son yoki ifoda) – ikki qiymatdan kichigini qaytaradi.
- pow(x,y) — x ni hisoblash (birinchi argumentni darajaga ko'tarish).
- random — 0 dan 1 gacha intervaldagi tasodifiy sonlarni hisoblash.
- round — sonni butun qismiga qarab yaxlitlash (masalan, round(15.5) natijasi 16 ni beradi, round(-15.5) daet -15).
- Math.round(x) (x — son yoki ifoda)
- Math.sin(x) (x — radianda berilgan burchak)
- Math.sqrt(x) (x — 0 ga teng yoki katta bo'lgan son yoki ifoda)
- tan —tangensni hisoblash.
- Math.tan(x) (x — radianda berilgan burchak)

## **4. Date klassi**

Date() metodi argumentsiz berilganda qiymati joriy sana va vaqtga ega Date ob'ekti yaratiladi. Date() metodida yangi ob'ekt uchun argumenti sifatida sana va zarur hollarda vaqt ko'rsatiladi. Date metodi JavaScript tili ob'eksti hisoblanib, HTML tilida hech qanday analogga ega emas. Ko'p hollarda Date ob'ekti metodlari uning ekzemplyati yordamida chaqiriladi, masalan:

```
D=new Date(); // bugungi sana va vaqtni olish  
system.write("Today is: "+d.toLocalString()); // va uni tasvirlash
```

Date ob'ektini yaratishning yuqoridagi sintaktikasida ko'rsatilgani bo'yicha, sana va vaqt hududiy vakt bo'yicha beriladi. Agarda tuzilayotgan dastur foydalanuvchi joylashgan chasovoy poyasga bog'liq bo'lmagan holda ishlashi zarur bo'lsa, u holda Grinвич (GMT) eki universal koordinatsiya vaqti (UTC) bo'yicha sanani ko'rsatish kerak bo'ladi.

Date ob'ektini yaratishda quyidagi 5 ta sintaktik variantdan foydalanish mumkin. 3-5 variantlarda vaqt hududiy tarzda interpretatsiya qilinadi (Grinвич da emas):

```
new Date();
new Date(millisekund) – bu erda millisekund joriy sana bilan 01.01.1970 sana yarimkuni orasidagi son;
new Date(sana satri) – bunda sana satri q oy nomi, dd, gg [chch:mm[:ss]])
new Date(yil, oy, kun) – bunda, yil 2011; oy 0-11; kun 1-31;
new Date(yil, oy, kun, soat, minut, sekund) – 24 soatlik tizimda.
```

### ***Date klassi metodlari***

getDate() - Date ob'ektining 1 dan 31 gacha oraliqdagi qiymatini beradi;  
 getDay() - Date ob'ektining 0 [yakshanba] dan 6 [shanba] gacha oraliqdagi hafta kunlari beradi;  
 getHours() - Date ob'ektining 0 [yarim tun] dan 23 gacha oraliqdagi soat qiymatini beradi;  
 getMinutes() - Date ob'ektining 0 dan 59 gacha oraliqdagi minut qiymatini beradi;  
 getSeconds() - Date ob'ektining 0 dan 59 gacha oraliqdagi sekund qiymatini beradi;  
 getMonth() - Date ob'ektining 0 [yanvar] dan 11 [dekabr] gacha oraliqdagi oylarni beradi;  
 getTime() - Date ob'ekti vaqt ko'rsatkichining millisekundlardagi qiymatini beradi;  
 getYear() - Date ob'ekti vaqt ko'rsatkichining yillar maydoni qiymatini beradi; bunda 2011 yil 11 ko'rinishida beriladi;  
 parse() – sananing satr ko'rinishidagi holatini sintaktik analiz qiladi va natijani millisekund formatida beradi;  
 setDate() - Date ob'ekti vaqt ko'rsatkichini o'rnatadi;  
 data.setDate(oy soni) //oy soni 1-31 oraliqda.  
 toLocalString() – joriy hududiy vaqt maydonini asosida Date formatini matnli (String) ko'rinishga keltiradi;  
 UTC() – sana va vaqtning rahamli ko'rinishini millisekundli formatga aylantiradi.

## **5. JavaScript tilidagi ob'ektlar**

JavaScript tilida uch turdagi ob'ektlar mavjud:

1. standart ob'ektlar;
2. brauzer ob'ektlari;
3. dasturchi tomonidan yaratiluvchi ob'ektlar.

Ularning har biri o'zlarining tasnifi va xususiyatlariga ega.

### ***Standart ob'ektlar***

Quyida JavaScript da qo'llaniluvchi standart ob'ektlar, xususiyatlar va usullar keltirilgan. Ularni ishlatishda oldindan e'lon qilish talab etilmaydi.

Ob'yekt	Tasnifi
Array	Massiv
Boolean	Mantiqiy ma'lumotlar
Date	Kalendarli vaqt
Function	Funksiya
Global	Global usullar
Math	Matematik konstanta va funksiyalar

Number	Son
Object	Ob'yekt
String	Satr

Standart ob'ektlar bilan qanday ishlash mumkin? Ancha oddiy. Ob'ektni realizatsiya qiluvchi dastur yoziladi va uning xususiyat va usullariga murojaat qilinadi. Misol sifatida joriy vaqtni ko'rsatuvchi HTML xujjatni ko'ramiz.

```
<HTML> <HEAD> <TITLE>Joriy kun va vaqt </TITLE> </HEAD>
<BODY BGCOLOR=WHITE>
  <H1> Joriy kun va vaqt </H1>
  <SCRIPT LANGUAGE="JavaScript">
    var dt;
    var MyDate="";
    dt = new Date();
    MyDate = "Date: " + dt.getDate() + "." + dt.getMonth() + "." +
dt.getYear();
    document.write(MyDate);
    document.write("<BR>");
    document.write("Time: " + dt.getHours()
+ ":" + dt.getMinutes() + ":" + dt.getSeconds());
  </SCRIPT> </BODY></HTML>
```

Bu erda JavaScript ssenariy new kalit so'zi yordamida Date ob'ektini yaratadi. Bunda Date konstruktori parametrlarsiz keltiriladi:

```
var dt;
dt = new Date();
MyDate = "Date: " +dt.getDate() + "."
+ dt.getMonth() + "." = dt.getYear();
```

getDate, getMonth va getYear usullar yordamida joriy sana olinadi. Ushbu usullar dt ob'ekti uchun chaqiriladi.

Matn satri esa HTML xujjatga write usuli yordamida bosmaga chiqariladi. Bu usul document ob'ektining usuli hisoblanadi:

```
document.write(MyDate);
```

Date ob'ekti joriy vaqtni ham o'z ichiga oladi. Bu ma'lumotlar getHours, getMinutes va getSeconds (soat, minut va sekund) usullari yordamida ko'riladi:

```
document.write("Time: " + dt.getHours()
+ ":" + dt.getMinutes() + ":" + dt.getSeconds());
```

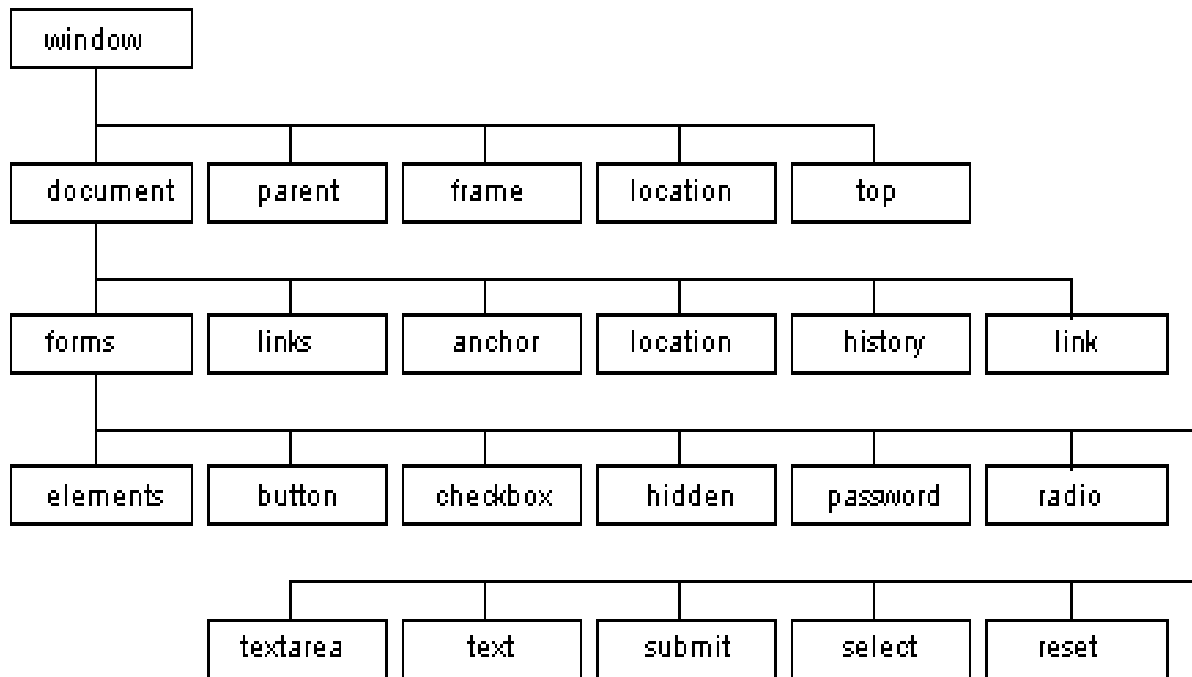
### ***Brauzer ob'ektlari***

JavaScript ssenariy nuqtai nazar bo'yicha ob'ektlar ierarxik daraxt ko'rinishda tashkil etiladi.

Brauzer ob'ektlari foydalanuvchi uchun yaratilgan, brauzer oynasida joylashgan ob'ektlar hisoblanadi. JavaScript ssenarida brauzer ob'ektlari, xususiyat va usullaridan foydalanib bir klass asosida boshqa klass yaratib bo'lmaydi.

### ***Brauzer ob'ektlari ierarxiyasi***

Quyidagi rasmda ob'ektlar daraxti ierarxiyasi keltirilgan.



Window ob'ekti bu ierarxiyaning ildizi hisoblanadi. Qachonki HTML xujjat yuklansa unmg ichida document, parent, frame, location va top boshqa ob'ektlar hosil bo'ladi.

### ***Ob'ektlar bilan bog'liq holatlar***

Brauzerning har bir ob'ekti bilan aniq bir holatlar to'plamidan tashkil topadi.

Masalan, window ob'ekti onLoad va onUnload holatlari bilan bog'liq holda ishlaydi. Birinchi holat brauzer oynani yuklab bo'lgach ishga tushadi. Ikkinchisi esa brauzer oynani yopish vaqtida ishga tushadi.

### ***Dasturchi tomonidan yaratiluvchi ob'ektlar***

Avvalo myRecord nomli klass yaratamiz. Hozircha unda usullar mavjud emas, ularni keyinchalik qo'shamiz. Bu klass quyidagicha yaratiladi:

```
function myRecord(name, family, phone, address) {
    this.name    = name;
    this.family  = family;
    this.phone   = phone;
    this.address = address;
    this.secure  q=false;
}
```

Yaratilayotgan ob'ektni xususiyatlarini ko'rsatish uchun maxsus this kalit so'zidan foydalaniladi. Bu kalit so'z ob'ektning xususiyatlariga bo'lgan murojaatini ko'rsatadi.

Keltirilgan klassdan qanday foydalanish mumkin? Yaratilgan klass asosida istalgan sondagi ob'ektlar yaratish mumkin. Quyida berilgan myRecord klassida ikkita rec1 va rec2 ob'ektlari yaratilgan:

```
var rec1;
var rec2;
rec1 = new myRecord("Ivan", "Ivanov",
    "000-322-223", "A. Temur ko'cha, d. 225, kv. 226");
rec2 = new myRecord("Petr", "Petrov",
    "001-223-3334", "Bobur ko'cha, d. 552, kv. 662");
rec2.secure = true;
```

Ob'ektlar new operatori yordamida yaratiladi.

### Nazorat savollari:

1. JavaScript tilining qanaqa funksiyalarini bilasiz?
2. Global klassi va uning operatorlarini sanab o'ting?
3. Math klassi va uning operatorlarini ayting?
4. Date klassi va uning operatorlarini ayting?
5. Qanday turdagi ob'ektlar mavjud?
6. Brauzer ob'yektlari ierarhiyasini chizma yordamida tushuntirib bering?
7. Turli klasslardan foydalanishning ahamiyatli tomonlarini aytib o'ting?
8. Turli klasslar orasidagi farqli jihatlarni ko'rsating?

## 9-ma'ruza.PHP dasturlash tili. Server tomondan dasturlash.PHPga kirish. PHPni o'rnatish va testlash.

### Reja:

1. PHP dasturlash tiliga kirish;
2. PHP dasturlash tili asoslari;
3. PHPdasturlash tili imkoniyatlari;
4. Dasturiy vositani sozlash va o'rnatish.

**Kalit so'zlar:** PHP dasturlash tili, izohlar, o'zgaruvchilar, konstantalar, tashqi o'zgaruvchilar, server tomonda dasturlash, denver paketi, Denver Distributivi.

**Ishdan maqsad:** PHP dasturlash tili bilan tanishish va ishlash jarayonini o'rganish. PHPni o'rganish va ma'lumotlar tiplaridan foydalanish. O'zgaruvchilar va amallar bilan ishlash o'rganish.

### 1. PHP dasturlash tiliga kirish

Hozirgi kunda internet keng ommalashgani sababli, zamon taraqqiyotini web-texnologiyasiz tassavvur etish mumkin emas. Web texnologiyalariga talab oshgan sari Web-dasturlash tillarini bilish har bir dasturchi uchun muhim vazifa sanalmoqda. Shularni inobatga olgan holda zamonaviy web-dasturlash tillaridan biri hisoblangan, sodda, o'rganishga qulay, barcha ma'lumotlar bazasi bilan ishlash oladigan PHP haqida batafsilroq to'xtalishga niyat qildik. Kelgusida bu til o'zbek tilida yoritilib boriladi hamda mutaxassis va o'rganuvchilar uchun forum tashkil qilinadi.

### PHP tarixi.

Ko'pgina boshqa dasturlash tillaridan farqli ravishda, PHP qandaydir tashkilot yoki kuchli dasturchi tomonidan yaratilgan emas. Uni oddiy foydalanuvchi Rasmus Lerdorf 1994 yili o'zining bosh sahifasini interaktiv uslubda ko'rsatish uchun yaratgan. Unga Personal Home Page (PHP – shaxsiy bosh sahifa) deb nom bergan.

1995 yili Rasmus PHPni o'zining HTML formalari bilan ishlaydigan boshqa dastur bilan umumlashtirib PHP/FI Version 2 ("Form Interpretator") hosil qildi. 1997 yilga borib PHP dan foydalanuvchi saytlar 50 mingdan oshdi. Shundan so'ng web texnologiya ustalari PHP g'oyasi asosida mukammal til yaratishga Ziva Suraski va Endi Gutmans asoschiligidan kirishildi. PHPni samarali deb hisoblanmagani uchun deyarli noldan boshlab, mavjud C va Perl tillaridan ibrat olib PHP3 talqinini yaratildi. 1999 yilga kelib PHP asosida qurilgan saytlar milliondan oshib ketdi. 2000 yilda esa Zend Technologies shirkati yangi ko'pgina funksiyalarni qo'shgan holda PHP4 sharhlovchisini yaratdi.

PHP – web texnologiya tili. PHPni o'rganish uchun avval HTML va dasturlash tilidan habardor bo'lish talab qilinadi. HTML/CSS va JavaScript larni mukammal bilguvchilar uchun PHPni o'rganish murakkablik tug'dirmaydi. PHPning vazifasi HTML faylini yaratib berish. JavaScript yordamida bajariladigan ko'pgina operatsiyalarni PHP orqali ham amalga oshirish mumkin, ammo e'tibor qilish lozimki, PHP – serverda; JavaScript – mijoz tomonida bajariladi. PHPda yozilgan kod serverning o'zida bajarilib, mijozga HTML shaklida etib boradi. Bu havfsizlik jihatdan ancha maqsadga muvofiq. JavaScript yordamida kod yozish, ma'lumot uzatish va qabul qilishni biroz tezlashtirsa-da, kodni mijoz ko'rish imkoniyatiga ega bo'ladi. Baribir har ikkisini boshqasi bosa olmaydigan o'z o'rnini bor, ravshanki bu o'rin PHPda muhimroq va kattaroq.

## 2. PHP TILI ASOSLARI PHP DASTURLARI

PHP dasturlari ikki usulda bajarilishi mumkin: Web-server tomonidan stsenariy ilovasi va konsol dasturi sifatida.

Bizning maqsadimiz web ilovalarni dasturlash bo'lgani uchun asosan birinchi usulni ko'ramiz.

PHP odatda Internet bilan bog'liq dasturlar yaratish uchun ishlatiladi. Lekin PHP dan komanda satrlar interpretatori, asosan \*nix tizimlarda foydalanish mumkin. Oxirgisi CORBA va COM interfeyslar hamda PHP-GTK kengaytmasi yordamida mumkin. Bu xolda quyidagi masalalarni echish mumkin:

- Interaktiv komanda qatorlari yordamida ilovalar yaratish;
- Kross-platformali GUI ilovalarni PHP-GTK bibliotekasi yordamida yaratish;
- Windows va Linux uchun ba'zi masalalarni avtomatizatsiya qilish

Serverga brouzerning murojlat qilishida yordamida php-stsenariylari bajarilishini ko'rib chiqamiz. Avval brouzer .php kengaytmali sahifani so'raydi, so'ngra web-server dasturni PHP mashinadan o'tkazadi va natijani html-kod shaklida qaytaradi. Agar standart HTML sahifani olib, kengaytmasini .php ga o'zgartirilsa va PHP mashinadan o'tkazilsa, foydalanuvchiga o'zgartirmasdan qaytaradi. Bu faylga PHP komandani qo'shish uchun, PHP komandalarni maxsus teglar ichiga olish kerak. Bu teglarning 4 xil shakli mavjud bo'lib, ixtiyoriysidan foydalanish mumkin:

1. XML qayta ishlash instruktsiyasi:
2. `<?php`
3. `...`
4. `?>`
4. SGML qayta ishlash instruktsiyasi:
5. `<?`
6. `...`
7. `?>`
7. HTML stsenariylari qayta ishlash instruktsiyasi:
8. `<script language = "php">`
9. `...`
10. `</script>`
11. ASP uslubidagi instruktsiya:
12. `<%`
13. `...`
14. `%>`



Biz XML yoki SGML uslubiga rioya qilamiz.

Xususan biror blok ichida PHPdan chiqish mumkin, faqat keyinchalik yana uning ichiga kirib kodni tugatish sharti bilan, quyidagi konstruktsiya mumkin:

```
<?
    if(5<3){
        echo("<p>Hello, world!<p>");
    ?>
    <p>Hello!</p>
    // bu qator PHP kodi sifatida qaralmaydi
    // va kod bloki bajarilayotgan bo'lsa chiqariladi
<?
    echo("<p>Hello, world!<p>");
}
?>
```

PHP da echo komandasi web – sahifalarda uchraydigan har qanday ma'lumotni( matn, HTML ajratuvchi simvoli, son) chiqarish uchun qo'llanadi. Uning ma'nosi misolda ko'rsatilgan.

### ***IZOXLAR***

PHP tilida izoxlarni joylash uchun bir necha usullar mavjud. Eng soddasi ikkilik slesh (//) dan foydalanish, shundan so'ng PHP satrlar oxirigacha yozilganni o'tkazib yuboradi. Bundan tashqari S (/\*...\*/) uslubidagi ko'p qatorli izoxlardan foydalanish mumkin. Bir qatorli izoxlar uchun (#) simvoldan foydalanish qulay.( UNIX script tillaridagi izox).

```
<php
    echo("<p>Hello</p>"); // izox
    echo("<p>Hello</p>"); # izox
    /*
        bu ham izox
    */
?>
```

Shuni esdan chiqarmaslik lozimki PHP uslubi izoxlari faqat PHP chegaranishlari orasida ta'sir qiladi. Agar PHP bu izoxlar simvollarini chegaranishlari tashqarisida uchratsa, ularni boshqa matnga o'xshab, html-sahifaga joylashtiradi.

Masalan:

```
<php
    echo("<p>Hello</p>"); // normal izox
?>
// bu izox brouzerda ko'rinadi.
<!-- HTML izoxi.
Bu izox HTML kodda ko'rinadi, brouzerda emas -->
```

Izoxlarni faqat operator oxiriga emas, quyidagicha joylash ham mumkin:

```
<?
    $a = "Hello, world";
    echo strstr($a,"H");
    // bu funktsiyani keyinchalik qarab chiqamiz
?>
```

#### **O'ZGARUVCHILAR VA KONSTANTALAR**

PHP da o'zgaruvchilar dollar (\$) belgisidan boshlanadi. Bu simvoldan ixtiyoriy sondagi harf, raqam va ostiga chizio' simvollar kelishi mumkin, lekin birinchi simvol albatta harf bo'lishi kerak. Shuni esda tutish kerakki, PHPda o'zgaruvchilarning nomlari kalit so'zlardan farqli registrga bog'liqdir.

PHP da o'zgaruvchilarni ta'riflaganda oshkora tipini ko'rsatish shart emas va dastur davomida itta o'zgaruvchi har xil tiplarga ega bo'lishi mumkin.

O'zgaruvchi unga qiymat berilganda initsializatsiya qilinadi va dastur bajarilguncha mavjud bo'ladi. Ya'ni web-sahifa xolida to so'rov tugamaguncha.

#### **TASHQI O'ZGARUVCHILAR**

Klient so'rovi veb-server tomonidan taxlil qilinib, PHP mashinaga uzatilgandan so'ng, u so'rovga tegishli ma'lumotlarni o'z ichiga olgan va bajarish davomida murojaat qilish mumkin bo'lgan bir necha o'zgaruvchilarni yaratadi. Oldin PHP sizni tizimingiz atrof muxit o'zgaruvchilarini oladi vash u nomdagi va shu qiymatdagi PHP stsenariysi atrofidagi o'zgaruvchilarni yaratadi, toki servedagi stsenariylarga klient tizimi xususiyatlari bilan ishlash mumkin bo'lsin. Bu o'zgaruvchilar **\$HTTP\_ENV\_VARS** assotsiativ massivga joylashtiriladi.

Tabiiyki **\$HTTP\_ENV\_VARS** massivi o'zgaruvchilari tizimga bog'liqdir (chunki ular aslida atrof muxit o'zgaruvchilaridir). Atrof muxito'zgaruvchilari qiymatlarini sizni mashinangiz uchun env (Unix) yoki set (Windows) komandasi yordamida ko'rishingiz mumkin.

So'ngra PHP u GET-o'zgaruvchilarning guruxini yaratadi. Ular so'rov satrini taxlil qilishda yaratiladi. So'rov satri **\$QUERY\_STRING** o'zgaruvchida saqlanadi va so'ralgan URL dagi "?" simvoldan keyingi informatsiyadan iborat. PHP so'rov satrini & simvollar bo'yicha aloxida elementlarga ajratadi, va har bir elementda "=" belgisini qidiradi. Agar "=" belgisi topilgan bo'lsa, tenglik chap tomonidagi simvoldan iborat o'zgaruvchi yaratadi. Quyidagi formani ko'ramiz:

```
<form action = "http://localhost/PHP/test.php" method="get">
```

```
    HDD: <input type="text" name="HDD"/><br>
```

```
    CDROM: <input type="text" name="CDROM"/><br>
```

```
<input type="submit"/>
```

Agar siz bu formada HDD qatorda "Maxtor", CDROM qatorda "Nec" tersangiz, quyidagi so'rov shaklini xosil qiladi:

```
http://localhost/PHP/test.php?HDD=Maxtor&CDROM=Nec
```

Bizning misolimizda PHP quyidagi o'zgaruvchilarni yaratadi: **\$HDD** = "Maxtor" va **\$CDROM** = "Nec".

Siz o'zingizni scriptingizdagi (bizda – test.php) bu o'zgaruvchilar Bilan oddiy o'zgaruvchilar bilan ishlagandek ishlashingiz mumkin. Bizning misolimizda ular ekranga chiqariladi:

```
<?
    echo("<p>HDD is $HDD</p>");
    echo("<p>CDROM is $CDROM</p>");
```

?>

Agar sahifa so‘rovi POST usuli yordamida bajarilsa, POST-o‘zgaruvchilarning guruxi yaratilib, interpretatsiya qilinadi va **\$HTTP\_POST\_VARS** massivga joylashtiriladi.

### KONSTANTALAR

Konstantalar PHP da **define()** funktsiyasi yordamida e‘lon qilinadi:

**define**(CONSTANT, value)

Bu funktsiya birinchi parametri – konstant nomi, ikkinchisi – uning qiymati. Konstantadan foydalanilganda nomi bo‘yicha ilova qilinadi:

<?

```
define(CONSTANT1,15);
define(CONSTANT2,"\x20"); // kod probela
define(CONSTANT3,"Hello");
echo(CONSTANT1);
echo(CONSTANT2);
echo(CONSTANT3);
```

?>

Odatga ko‘ra konstantalar nomlari yuqori registr harflari bilan yoziladi. Bu faqat odat bo‘lsa ham unga rioya qilishni maslahat beramiz, chunki yaxshi odatlarga rioya qilmaydigan dasturchilardan yomon dasturchilar chiqadi. Konstantalar aniqlanganligini **defined()** funktsiyasi yordamida tekshirish mumkin:

<?

```
define(CONSTANT,"Hello");
if(defined("CONSTANT"))
{
    echo("<p>CONSTANT is defined</p>");
}
```

?>

### 3. PHP imkoniyatlari

«PHP da har qanday dastur bajarsa bo‘ladi», – degan edi uning yaratuvchisi. Birinchi navbatda PHP tili server tomonidan bajariladigan skriptlar yaratish uchun foydalaniladi va aynan shuning uchun u yaratilgan. PHP tili ixtiyoriy CGI-skriptlari masalalarini echishga va bundan tashqari html formali ma’lumotlarni qayta ishlashga hamda dinamik ravishda html sahifalarni ishlab chiqishga qodir. Biroq PHP tili foydalaniladigan boshqa sohalar ham mavjud. Bu sohalarni biz uchta asosiy qismga bo‘lamiz:

Birinchi soha – biz yuqorida aytib o‘tganimizdek, server tomonidan bajariladigan ilovalar (skriptlar) yaratish. PHP tili bunday turdagi skriptlarni yaratish uchun juda keng qo‘llaniladi. Bunday ish ko‘rsatish uchun PHP-parser (ya’ni php-skriptlarni qayta ishlovchi) va skriptlarni qayta ishlovchi web-server, skriptlarni natijasini ko‘rish uchun brauzer va albatta php-kodini yozish uchun qanday bo‘lsa ham matn muharriri kerak bo‘ladi. PHP-parser CGI-dasturlar ko‘rinishida yoki server modullari ko‘rinishida tarqalgan. Uni va web-serverni kompyuterimizga qanday o‘rnatamiz, biz bu haqida keyinroq ko‘rib o‘tamiz.

Ikkinchi soha – buyruqlar satrida bajariladigan skriptlarni yaratish. Ya’ni PHP tili yordamida biror-bir kompyuterda brauzer va web-serverlardan mustahil ravishda o‘zi bajariladigan skriptlarni ham yaratish mumkin. Bu ishlarni bajarish uchun hech bo‘lmaganda PHP-parser (bu holatda biz uni buyruqlar satri interpretatori (CLI, command line interpreter) deb ataymiz) talab etiladi. Bunday ishlash uslubi turli

masalalarni rejalashtirish yordamida bajarilishi uchun kerak bo'lgan skriptlar yoki oddiy matnni qayta ishlash uchun kerak bo'lgan masalaga o'xshash ishlaydi.

Va nihoyat oxirgi uchinchi soha – bu mijoz tomonidan bajariladigan GUI-ilovalarni (grafik interfeys) yaratish. Bu soha PHP tilini endigina o'rganayotgan foydalanuvchilar uchun uncha muhim bo'lmagan sohadir. Biroqagarda siz PHP tilini chuqur o'rgangan bo'lsangiz, bu soha siz uchun ancha muhimdir. PHP tilini bu sohaga qo'llash uchun php kengaytmali maxsus yordamchi – PHP-GTK talab etiladi.

Shunday qilib, PHP tilini qo'llanilish sohalari keng va turlichadir. Yuqoridagi masalalarni echa oladigan boshqa turlicha dasturlash tillari ham mavjud, unda nima uchun PHP tilini o'rganishimiz kerak? U til bizga nima beradi? Birinchidan, PHP tili o'rganish uchun juda qulay. PHP tilini sintaksisi asosiy qoidalari va ishlash printsiplari bilan yetarlicha tanishib chiqib o'zingizni shaxsiy dasturingizni tuzib ko'rib, so'ngra uni boshqa dasturlash tillarida tuzilgan variantlari bilan solishtirsangiz bunga guvohi bo'lasiz.

Ikkinchidan, PHP tili barcha bizga ma'lum platformalarda, barcha operatsion tizimlarda hamda turlicha serverlarda erkin ishlay oladi. Bu xususiyat juda muhim. Masalan, kimdir Windows operatsion tizimdan Linux operatsion tizimga yoki IIS serverdan Apache serverga o'tmoqchi bo'lsa PHP tilini o'rganishi shart.

PHP dasturlash tilida dasturlashning ikkita hammabop paradigmalarini ishlatiladi, bular protsedurali va ob'ektlilik dasturlash. PHP4 dasturlash tili protsedurali dasturlashni butunlay qo'llab quvvatlaydi, Biroq ob'ektlilik dasturlashni ham qo'llasa bo'ladi. PHP5 dasturlash tilining birinchi testlash versiyasida PHP4 dasturlash tilida uchraydigan ob'ektga yo'naltirilgan dasturlash modellarining kamchiliklari to'ldirilgan. Shunday qilib, Hozirda tanish bo'lib ulguragan ishlash printsiplarini tanlash kerak.

Agarda PHP tilini Hozirgi imkoniyatlari to'g'risida gaplashadigan bo'lsak, u holda biz PHP tilini birinchi versiyasidan ancha yiroqlashib ketgan bo'lamiz. PHP dasturlash tili yordamida tasvirlar, PDF-fayllar, flesh-rolklar yaratish mumkin; Hozirgi vaqtdagi zamonaviy ma'lumotlar bazasini qo'llab quvvatlaydi; ixtiyoriy matnli fayl formatlari bilan, hamda XML va fayllar tizimi bilan ishlaydigan funksiyalar ham qo'shilgan. PHP tili turli servislari o'rtasidagi protokollarning o'zaro aloqasini qo'llab quvvatlaydi. Bularga misol papkalarga kirishni boshqarish protokoli LDAP, tarmoq qurilmalari bilan ishlaydigan protokol SNMP, ma'lumotlarni uzatish protokollari IMAP, NNTP hamda POP3, gipermatnlarni uzatish protokoli HTTP va boshqalarni olish mumkin.

PHP dasturlash tilini turli dasturlash tillari o'rtasidagi o'zaro aloqasiga diqqatni qaratsak, bunga Java dasturlash tilini aytib o'tish kerakki, Java dasturlash tili ob'ektlarini PHP tili o'z ob'ektlari sifatida qaraydi. Ob'ektlarga murojaat sifatida CORBA kengaytmalaridan foydalaniladi.

Matnli axborotlar bilan ishlash uchun PHP tili o'ziga Perl dasturlash tilidagi tartiblangan ifodalar bilan ishlay oladigan mexanizmlarni (katta bo'lmagan o'zgarishlarsiz) va UNIX-tizimini meros qilib oladi. XML-hujjatlarini qayta ishlash uchun standart sifatida DOM va SAX, XSLT-transformatsiyasi uchun API dan foydalanishi mumkin.

Elektron tijorat ilovalarini yaratish uchun bir qator to'lovni amalga oshiradigan Cybercash, CyberMUT, VeriSign Payflow Pro hamda CCVS kabi foydali funksiyalar mavjud.

#### **4. Dasturiy vositani sozlash va o'rnatish**

Yuqorida PHP tili imkoniyatlarini, qo'llanilish sohalarini muhokama hildik va tarixini o'rgandik. Endi dasturiy vositani o'rnatishga kerak bo'lgan uskunalar majmuini ko'rib o'tsak. Modomiki, asosiy kursning amaliyoti sifatida biz quyidagi masalalarni ko'rib chiqamiz: mijoz-server texnologiyasi sifatida ishlanadigan masalalar, mos ravishda skriptlar yaratilishida qo'llanilishi, serverlarni qayta ishlash. Bular uchun bizga web-server hamda PHP tili interpretatori kerak bo'ladi. Web-server sifatida, masalan, web-mutaxassislar o'rtasida mashhur bo'lgan Apache serverni olamiz. Dastur natijasini ko'rish uchun web-brauzer kerak bo'ladi, bunga misol Internet Explorer.

##### ***Denver Distributivi***

Biz Yuqorida Linux va Windows platformalari uchun PHP dasturiy vositasini sozlash va o'rnatish bilan etarlicha tanishmiz. PHP dasturiy vositasi va uni ishlashi uchun kerak bo'ladigan komponentalarni o'rganishni xohlamaydiganlar uchun PHP dasturining tayyor PHP tilini to'ldiradigan distributivlari mavjud. Bunday distributivlar ichida keng tarqalgani - Denver (<http://dklab.ru/chicken/web/>). Uni o'rnatishni o'rganish uchun web-mutaxassislar saytlariga murojaat qilish kerak. Denverni o'rnatish juda oddiy hamda unga hech qanday bilim talab etilmasligini aytib o'tish kerak. Bu distributivni PHP tilini endigina

o'rganayotgan yosh dasturchilar uchun tavsiya etamiz. Jiddiy masalalarni hal etish uchun esa PHP dasturlash tilini to'liq o'rnatish va sozlash kerak bo'ladi.

### Nazorat savollari:

1. Qanaqa dasturlash tillarini bilasiz?
2. PHP qanday dasturlash tili hisoblanadi;
3. Bu tilni vujudga kelishiga sababchi bo'lgan ehtiyojlar nimadan iborat deb o'ylaysiz?
4. PHP dasturlash tili haqida umumiy ma'lumot bering?
5. PHP dasturlash tilining asosiy tushunchalarini keltiring?
6. PHP dasturlash tili imkoniyatlari haqida gapiring.
7. Server tomonida dasturlash deganda nimani tushunasiz?
8. PHP ni ishga tushirish, Denver paketi va undan foydalanish haqida ma'lumot bering?
9. Boshqa dasturlash tillari bilan PHP dasturlash tilining tilining farqini tushuntiring?
10. PHP tilining qanday afzalliklari bor?

### **10-ma'ruza. PHP dasturlash tilining tuzilishi. Ma'lumotlar tiplari va ifodalar bilan ishlash. Jarayonlarni boshqarish.**

#### Reja:

1. PHP dasturlash tilining asosiy tuzilishi;
2. PHP dasturlash tilidagi asosiy sintaksislar;
3. Ma'lumotlar tiplari bilan ishlash;
4. Alternativ sintaksislar.

**Kalit so'zlar:** PHP kod sintaksisi, ma'lumot tiplari, asosiy sintaksislar, alternativ sintaksislar , ifodalar, jarayonlarni boshqarish

**Ishdan maqsad:** PHP dasturlash tili bilan tanishish va ishlash jarayonini o'rganish. PHPni o'rganish va bu tilning asosiy tuzilishi haqida ma'lumotga ega bo'lish. Ma'lumotlar tiplari bilan ishlash jarayoni bilan tanishib chiqish. Bu tilni Web texnologiya yaratish jarayoniga tatbiq qilish.

### **1. PHP dasturlash tilining asosiy tuzilishi**

Ko'p hollarda PHP tilini interpretatori ishlayotganligini tekshirib ko'rish uchun tuziladigan dastur eng sodda dastur deb ataladi. Hozir biz PHP tilidagi ushbu dasturni chuqur o'rganamiz hamda uni boshqa dasturlash tillari Ci, Perl va JavaScript lardan farqli tomonini tekshiramiz. Ushbu misolni ko'ramiz:

```
<html>
    <head>
        <title> Misol </title>
    </head>
    <body>
        <?php
            echo "<p> Salom, bu men - PHP skript! </p>";
        ?>
    </body>
</html>
```

Bu PHP dasturlash tilining maxsus kodli teglari yordamida tuzilgan sodda html-fayldir.

Yuqorida aytib o'tganimizdek, PHP dasturlash tili Ci va Perl dasturlash tiliga o'xshash. Biroq keltirilgan dastur Ci va Perl dasturlash tilidagi dasturdan ancha katta farq qiladi. Bu erda HTML sahifaga chiqarish uchun bir qator maxsus buyruqlarni yozish shart emas. Bevosita PHP-kod asosida qurilgan biror vazifani bajaradigan HTML-skript yoziladi (bizni misolda ekranda chiqarilgan matn). PHP dasturlash tilining Ci va Perl dasturlash tillaridan kamchiligi shuki, murakkab skriptlarni PHP dasturlash tili ancha sekin bajaradi.

PHP-skriptlar – bu serverda bajariladigan va qayta ishlanadigan dasturlardir. Bu skriptlarni JavaScript tipidagi skriptlar bilan taqqoslash mumkin emas, chunki JavaScript tilidagi skriptlarda yozilgan buyruqlar faqat mijoz kompyuteridagina bajariladi. Mijoz kompyuterida va server kompyuterida bajariladigan skriptlarning farqi nimada? Agarda skript serverda qayta ishlansa, mijoz kompyuteriga faqatgina natija yuboriladi. Masalan, agarda serverda skript bajarilayotgan bo'lsa, Yuqorida keltirilganga o'xshab mijoz HTML-sahifa ko'rinishdagi natijani oladi:

```
<html>
    <head>
        <title> Misol </title>
    </head>
    <body>
        <p> Salom, bu men – PHP skript! </p>
    </body>
</html>
```

Bu holatda mijoz qanday kod bajarilayotganini bilmaydi. O'z serveringizni HTML-fayllarni PHP protsessori qayta ishlaydigan qilib sozlab olishingiz ham mumkin. Ya'ni mijozlar oddiy HTML-faylni qabul hildimi yoki skript natijasini ko'rdimi buni bila olmaydi. Agarda skript mijoz kompyuterida qayta ishlansa (masalan, JavaScript tilidagi dastur), u holda mijoz skript kodidan iborat HTML-sahifani ko'radi.

Biz yuqorida aytib o'tgandikki, PHP-skriptlar HTML-kod ichida yoziladi. qanday qilib degan savol tuqiladi. Buning bir nechta usullari mavjud. Bulardan biri birinchi misolda keltirilganidek, <?php tegi bilan boshlanib ?> tegi bilan tugagan sintaksis. Bunday ko'rinishdagi maxsus teglar HTML va PHP rejimidagina ishlatiladi. Bu sintaksis PHP tilini XML hujjatlari bilan birgalikda ishlaydigan dasturlarida juda ma'qul ko'riladi (masalan, XHTML tilida yozilgan dasturlarda). Biroqbazan quyidagi alternativ variantdan foydalansa ham bo'ladi(echo "Some text" buyrug'i «Some text» matnini ekranga chiqaradi.):

```
<? echo "Bu PHP tilida
        oddiy qayta ishlashning
        instruktsiyasi"; ?>

<script language="php">
    echo "Bir nechta redaktorlar
        (FrontPage) quyidagicha
        qabul qilishadi";
</script>

<% echo " ASP texnologiyasidagi tegdan
        ham foydalansa bo'ladi"; %>
```

Bu keltirilgan usullardan birinchisi har doim ham bajarilavermaydi. Undan foydalanish uchun qisqa teglarni ishlatish kerak, yoki PHP3 uchun short\_tags() funksiyani ishlatish kerak, yoki PHP tilining konfiguratsion fayliga short\_open\_tag buyruqni o'rnatish kerak, yoki PHP dasturlash tilida enable-short-tags parametr bilan kompilyatsiya qilish kerak. Agarda php.ini-dist buyruqqa Yuqoridagilar avtomatik qo'shilgan bo'lsa, u holda qisqa teglardan foydalanish tavsiya etilmaydi. Ikkinchi usul xuddi o'rniga qo'yishga o'xshaydi, masalan, JavaScript kodlari va uning uchun mos html teglar. Shuning uchun undan har doim foydalanish mumkin, lekin bu noqulayligi uchun kamdan-kam ishlatiladi. Uchinchi usuldan faqat ASP texnologiyasidagi teglar asp\_tags konfiguratsiyasida ishlatilgandagina foydalaniladi.

PHP dasturlash tili faylni qayta ishlayotganda u oddiy matnni PHP kod interpretatsiya qilishi kerak bo'lgan maxsus teglarni uchratmaguncha qaytarib beradi. Interpretator haqida gapirganda u topilgan barcha kodni yopiladigan teggacha bajaradi, so'ng yana oddiy matn qaytariladi. Bu mexanizm PHP-kodni HTML sahifaga aylantiradi, ya'ni barcha PHP teglardan tashqari barcha matnlarni o'zgarishsiz saqlaydi va ichkaridagilarni esa interpretatsiyalaydi. Yana shuni aytish kerakki, PHP-fayl CGI-skriptga o'xshamaydi. PHP-fayl bajarilishi shart emas, yoki yana qandaydir belgilanadi.

PHP-faylni serverda qayta ishlash uchun jo'natishda server tomonidan brauzer satrida bu faylni yo'lini ko'rsatish shart. PHP skriptlar www orqali kirishga ruxsat etilgan joyda joylashishi shart. Agarda PHP-fayl lokal kompyuterda mavjud bo'lsa, u holda uni buyruqlar satri interpretatori yordamida qayta ishlash mumkin.

Shunday qilib, biz PHP dasturlash tili haqida ma'lumotga ega bo'ldik, u qanday dunyoga kelgan va tarqalgan, uni qanday va qayerda foydalanilishini o'rgandik, dasturiy vositani o'rnatdik hamda uni ishlashi uchun barcha sozlashlarni bajardik va php-dastur nimalardan tashkil topishini angladik. Keyingi bo'limlarda biz PHP dasturlash tilining asosiy sintaksislarini ko'rib chihamiz hamda bir qancha amaliy masalalarni hal etamiz.

## 2. PHP dasturlash tilidagi asosiy sintaksislar

Instruktsiyani bir nechta qismga bo'lib ko'rib chiqamiz, ya'ni:

- kommentariylar yaratish;
- o'zgaruvchilar;
- o'zgarmaslar;
- ma'lumot tiplari;
- operatorlar.

Biz endi PHP dasturlash tilining asosiy sintaksis elementlarini o'rganishga o'tamiz. Misol sifatida elektron maktub tayyorlash masalasini ko'rib o'taylik. Uning ma'nosi quyidagidan iborat.

Faraz qilamizki, sizda qandaydir e'lon va e'lonni jo'natishingiz kerak bo'lgan bir nechta odamlar mavjud bo'lsin. Buning uchun siz e'lonni ichida o'zgaradigan (qabul qiluvchi bilan bog'liq bo'lmagan) bir nechta parametrlari mundarijasi bilan tayyorlaysiz.

Birinci navbatda PHP dasturlash tili sintaksisiga nisbatan nimalarni bilish kerak. Bu HTML-kod ichiga o'rnatilgan va PHP dasturlash tilidagi koddir, uni interpretator farqlay biladi. Avvalgi bo'limlarda bular haqida aytib o'tgandik. hammasini qaytarib o'tmaymiz, faqat biz ko'p hollarda misollarda `<?php ?>` variant o'rniga qisqartirilgan `<? ?>` teglardan foydalanishni aytib o'tamiz.

### *Instruktsiyalarni ajratilishi.*

PHP dasturlash tilidagi dastur(ixtiyoriy dasturlash tilidagi) – bu buyruqlar (instruktsiyalar) to'plamidir. Dasturni qayta ishlash uchun bir buyruqni boshqa buyruqdan farqini bilish kerak. Buning uchun maxsus simvollar – ajratgichlardan foydalaniladi. PHP dasturlash tilida instruktsiyalarni xuddi Ci yoki Perl dasturlash tillari kabi ajratiladi, ya'ni har bir ifoda nuqtali vergul (“;”) bilan tugaydi.

«?>» yopiladigan teg ham instruktsiyani tugashini anglatadi, shuning uchun undan oldin nuqtali vergul qo'yilmaydi. Masalan, quyidagi ikki fragmentlar ekvivalentdir:

```
<?php
    echo "Hello, world!"; // buyruqlar oxirida nuqtali vergul qo'yish
shart
?>
<?php
    echo "Hello, world!" ?>
    <!-- "?"> borligi uchun
    nuqtali vergul tashlab ketildi -->
```

## ***Kommentariylar.***

Ko'p hollarda dastur tuzganda kodni tushunarli bo'lishi uchun unga qandaydir Izoh-kommentariylar qo'yish kerak bo'lib qoladi. Bu holat katta hajmdagi dasturlar yaratganda hamda agarda bitta dastur ustida bir nechta dasturchi ishlayotganda juda muhim. Kommentariylar dasturning kodi tushunarli bo'lishi uchun yoziladi. Bundan tashqari masalani qismlarga ajratib hal qilinganda ishning kamchiligi bor joyida keyinchalik esdan chiqmasligi uchun kommentariya yozib qo'yiladi. Barcha dasturlash tillarida dastur ichiga kommentariya qo'shish imkoniyati mavjud. PHP dasturlash tili bir qancha ko'rinishdagi kommentariylarni qo'llab quvvatlaydi: C, C++ dasturlash tillari stilidagi hamda Unix qobig'idagi kommentariylar. // va # belgilar bir satrli kommentariylarni anglatsa, /\* va \*/ belgilar esa mos ravishda ko'p satrli kommentariylarning boshlanish va tugashini anglatadi.

**Misol:** PHP dasturlash tilida kommentariyning qo'llanilishi:

```
<?php
    echo "Meni ismim Alisher";
        // Bu bir satrli kommentariy
        // S++ dasturlash tili stilidagi
    echo "Meni familiyam Boliev";
    /* Bu ko'p satrli kommentariy. Bu erga bir qancha satr yozish mumkin.
    Dastur bajarilish jarayonida bu erdagi barcha yozuvlar (kommentariylangan),
    o'qilmaydi. */
    echo "Men PHP dasturlash tilini INTUIT.ru dan o'rganyapman";
        # Bu kommentariy
        # Unix qobig'idagi kommentariy.
?>
```

## ***O'zgaruvchilar, o'zgarmaslar va operatorlar***

Har bir dasturlash tilida muhim elementlardan biri bu o'zgaruvchilar, o'zgarmaslar va ular qo'llaniladigan operatorlardir. Bu kattaliklar haqida biz 9-ma'ruzada ham to'xtalib o'tgan edik. PHP dasturlash tili bu elementlarni qanday belgilashi va qayta ishlashini ko'rib chiqamiz.

### ***O'zgaruvchilar***

PHP dasturlash tilida o'zgaruvchilar oldiga dollar belgisi ("\$\$") qo'yib e'lon qilinadi, masalan, \$my\_var.

O'zgaruvchilar nomlari registrlarni farqlaydi, ya'ni \$my\_var hamda bosh harfli \$My\_var o'zgaruvchilari turli xil o'zgaruvchilardir.

PHP dasturlash tilida o'zgaruvchilar nomi qolgan dasturlash tillari qoidalari kabi e'lon qilinadi: o'zgaruvchi nomi lotin alfaviti bilan boshlanishi va undan keyin harflar yoki tagiga chizilgan belgi yoki rahamlar bo'lishi mumkin.

PHP4 dasturlash tilida bulardan tashqari o'zgaruvchiga qiymat o'zlashtirishning yana bir usuli mavjud: ssilka bo'yicha o'zlashtirish. Ssilka bo'yicha o'zgaruvchiga qiymat o'zlashtirish uchun uni nomi bo'lishi shart, ya'ni u qandaydir o'zgaruvchini taqdim etishi kerak. Bir o'zgaruvchi qiymatini boshqa o'zgaruvchiga Ssilka bo'yicha o'zlashtirish uchun birinchi o'zgaruvchi oldiga ampersand & belgisi qo'yish shart. Bunga Yuqoridagi misolni ko'rib chihamiz, faqat first o'zgaruvchi second o'zgaruvchiga ssilka bo'yicha o'zlashtiriladi:

**Misol.** Ssilkalalar bo'yicha o'zlashtirish.

```
<?php
$first = ' Text '; // $first o'zgaruvchiga
                // ' Text ' qiymat o'zlashtirildi
```



```

$second = &$first;
/* $second.orqali $first o'zgaruvchiga ssilka qilamiz
   Endi bu o'zgaruvchilar qiymatlari
   har doim tengdir */
// $first o'zgaruvchi qiymatini
// ' New text ' qiymatga o'zgartiramiz
$first = ' New text ';
echo "first nomli o'zgaruvchi qiymati $first ga teng <br>";
// $second o'zgaruvchi qiymatini ekranga chiqaramiz
echo "second nomli o'zgaruvchi qiymati " . "$second ga teng";
?>

```

Bu skriptni natijasi esa quyidagicha bo'ladi:

first nomli o'zgaruvchi qiymati New text ga teng.  
second nomli o'zgaruvchi qiymati New text ga teng.

Ya'ni \$first o'zgaruvchi qiymati o'rniga \$second o'zgaruvchi qiymati o'zlashtirildi.

### ***O'zgarmlar***

Skript bajarilish jarayonida o'zgaraydigan qiymatli kattaliklarni saqlash uchun o'zgarmlardan foydalaniladi. Bunday kattaliklar matematik o'zgarmlar, parollar, fayllarning yo'llari va boshqalar bo'lishi mumkin. O'zgarmlarning o'zgaruvchilardan asosiy farqi shuki, ularni faqat bir martagina o'zlashtiriladi va uni qiymatini e'lon qilingandan keyin bekor qilib bo'lmaydi. Bundan tashqari o'zgarmlar oldida dollar belgisi qo'yilmaydi hamda uni oddiy qiymat o'zlashtirish kabi qarash mumkin emas. O'zgarmlar qanday aniqlanadi? Buning uchun maxsus define() funksiyasi mavjud, uning sintaksisi quyidagichadir:

```
define("O'zgarmlar nomi", "O'zgarmlar qiymati", [registrga_sezgirli_kichik])
```

O'zgarmlar nomi registrga segirli katta. har bir o'zgarmlarda uni o'zgartirish mumkin, ya'ni registrga\_sezgirli\_kichik argumentni qiymati sifatida True qiymati ko'rsatiladi. O'zgarmlar nomi har doim katta registr bilan yozishga kelishib olingan.

O'zgarmlarni qiymatini bilish uchun uni nomini ko'rsatish kerak. O'zgaruvchidan farqi o'zgarmlar nomi oldiga \$ belgi qo'yilmaydi. Bundan tashqari o'zgarmlarni qiymatini bilish uchun konstanta nomi bilan parametr sifatida constant() funksiyasidan foydalanish mumkin.

**Misol.** PHP dasturlash tilida o'zgarmlar.

```

<?php
// o'zgarmlarni aniqlaymiz PASSWORD
define("PASSWORD","qwerty");
// registrlanmagan PI o'zgarmlarni qiymatini aniqlaymiz 3.14
define("PI","3.14", True);
// PASSWORD o'zgarmlar qiymatini olamiz, ya'ni qwerty
echo (PASSWORD);
// bu ham qwerty ni chiqaradi
echo constant("PASSWORD");
echo (password);
/* password ni chiqaradi va biz registrlangan o'zgarmlar PASSWORD ni
   kutgandik.*/
echo pi;
// 3.14 ni chiqaradi, chunki o'zgarmlar PI registrlanmagan va
   aniqlangan.
?>

```

Dasturchi tomonidan o'zgaruvchilardan tashqari Yuqorida aytib o'tganimizdek PHP dasturlash tilida mavjud o'zgarmlar ham interpretator tomonidan aniqlanadi. Masalan, `__FILE__` o'zgarma dastur bajarilish jarayonida fayl nomini (va fayl yo'lini), `__FUNCTION__` funksiya nomidan tashkil topadi, `__CLASS__` - sinf nomi, `PHP_VERSION` – PHP dasturlash tili interpretatori versiyasini o'zida saqlaydi. Bunday o'zgarmlarning barcha ro'yxatini PHP dasturlash tili uchun mo'ljallangan qo'llanmalardan topish mumkin.

### Amallar.

O'zgaruvchilar, o'zgarmlar va ifodalar ustida turli hisoblashlarni bajaradigan bu amallardir. Biz hali bu ifodalar haqida to'xtab o'tganimiz yo'q. Ifodalar qiymatini ushbu amallar yordamida aniqlanadi. O'zgaruvchilar va o'zgarmlar – bu ifodalarning asosiy va juda sodda shaklidir. Shunday ifodalarni ko'paytirishi mumkin bo'lgan amallar to'plami mavjud. Ularni quyida to'liqroq muhokama qilamiz:

**9.1-jadval . Arifmetik amallar.**

Belgilanishi	Nomlanishi	Misollar
+	Qo'shish	$\$a + \$b$
-	Ayirish	$\$a - \$b$
*	Ko'paytirish	$\$a * \$b$
/	Bo'lish	$\$a / \$b$
%	Bo'lishdagi qoldiq	$\$a \% \$b$

**9.2-jadval. Satrli amallar.**

Belgilanishi	Nomlanishi	Misollar
.	Konkatenatsiya (satrlarni qo'shish)	$\$c = \$a . \$b$ (bu $\$c$ satr $\$a$ va $\$b$ satrlardan iborat)

**9.3-jadval. O'zlashtirish amallari.**

Belgilanishi	Nomlanishi	Izohlar	Misollar
=	O'zlashtirish	Operatordan tomonda turgan o'zgaruvchilar ustida bajarilgan amallardan hosil bo'lgan natija qiymati o'zlashtiriladi.	$\$a = (\$b = 4) + 5;$ ( $\$a$ 9 ga teng, $\$b$ 4 ga teng)
+=		Qisqartirish. O'zgaruvchiga son qo'shiladi va keyin natija o'zlashtiriladi.	$\$a += 5;$ ( $\$a = \$a + 5$ ifodaga ekvivalent; )
.=		zlashtirish va konkatenatsiya amallari kombinatsiyasini qisqartirilgan shakli (dastavval satrlar qo'shiladi, so'ngra hosil bo'lgan satr o'zgaruvchiga o'zlashadi).	$\$b = "Hammaga ";$ $\$b .= "salom";$ ( $\$b = \$b . "salom"$ ifodaga ekvivalent;) Natijasi: $\$b = "Hammaga salom"$

**9.4-jadval. Mantiqiy amallar.**

Belgilanishi	Nomlanishi	Izohlar	Misollar
and	VA	\$a VA \$b rost (True)	\$a and \$b
&&	VA		\$a && \$b
or	YOKI	\$a yoki \$b o'zgaruvchilardan hech bo'lmaganda bittasi rost bo'lsa (ikkalasi ham rost bo'lishi mumkin).	\$a or \$b
	YOKI		\$a    \$b
xor	INVERSIYA YOKI	O'zgaruvchilardan bittasi rost bo'lsa . Agarda ikkalasi ham rost bo'lsa inersiyalanadi.	\$a xor \$b
!	INVERSIYA (NOT)	Agarda \$a=True, u holda !\$a=False va aks holda teskari bo'ladi.	! \$a

**9.5-jadval. Taqqoslash amallari.**

Belgilanishi	Nomlanishi	Izohlar	Misollar
==	Tenglik	O'zgaruvchilar qiymatlari teng.	\$a == \$b
===	Ekvivalentlik	O'zgaruvchilar qiymatlari va tiplari teng.	\$a === \$b
!=	Tengsizlik	O'zgaruvchilar qiymatlari teng emas.	\$a != \$b
<>	Tengsizlik		\$a <> \$b
!=	Noekvivalentlik	O'zgaruvchilar ekvivalent emas.	\$a != \$b
<	Kichik		\$a < \$b
>	Katta		\$a > \$b
<=	Kichik yoki teng		\$a <= \$b
>=	Katta yoki teng		\$a >= \$b

**9.6-jadval. Inkrement va decrement amallari.**

Belgilanishi	Nomlanishi	Izohlar	Misollar
++\$a	Pre_ inkrement	\$a qiymati birga oshiriladi va \$a qiymati qaytariladi.	<? \$a=4;
\$a++	Post- inkrement	\$a qiymati qaytariladi va so'ngra \$a qiymati birga oshiriladi.	echo "4 bo'lishi shart:" .\$a++; echo "6 bo'lishi shart:" .++\$a;
--\$a	Pre- decrement	\$a qiymati birga kamaytiriladi va \$a qiymati qaytariladi.	?>
\$a--	Post- decrement	\$a qiymati qaytariladi va so'ngra \$a qiymati birga kamaytiriladi.	

### 3. Ma'lumotlar tiplari bilan ishlash

PHP dasturlash tili sakkizta sodda ma'lumot tiplarini qo'llab quvvatlaydi:

To'rttasi skalyar tiplar:

- boolean (mantiqiy);
- integer (butun);
- float (nuqtasi siljiydigan);
- string (satrlı).

Ikkitasi aralish tiplar:

- array (massiv);
- object (ob'ekt).

Ikkitasi maxsus tiplar:

- resource (resurs);
- NULL.

PHP dasturlash tilida o'zgaruvchilar tiplari oshkora e'lon qilinmaydi. Ko'pincha o'zgaruvchi qo'llanilgan kontekstdan, ya'ni o'zgaruvchiga o'zlashtirilgan qiymat itpidan mustahil ravishdagi dastur bajarilish jarayonidan interpretator o'zi bu ishni bajaradi. Quyida yuqorida sanab o'tilgan ma'lumotlar tiplarini birma-bir ko'rib chiqamiz.

#### ***Boolean tipi(Bul yoki mantiqiy tip).***

Bu sodda tip qiymatni rost ekanligini ifodalaydi, ya'ni o'zgaruvchi faqat ikkita qiymat qabul qiladi – rost TRUE yoki yolg'on FALSE.

Mantiqiy tiplarni aniqlash uchun TRUE yoki FALSE kalit so'zlaridan foydalanamiz. Bu ikkala tiplar registrlanmagan.

**Misol.** Mantiqiy tip.

```
<?php
$test = True;
?>
```

Mantiqiy tiplar turli boshqariladigan konstruktsiyalarda (sikllar, shartlar va shunga o'xshash, bular haqida keyinroq aytib o'tamiz) qo'llaniladi. Bir qancha amallar (masalan, tenglik amali) ham mantiqiy tip qabul qilishi mumkin, ya'ni faqat ikki qiymat rost yoki yolg'on qiymatni qabul qiladi. Ular boshqariladigan konstruktsiyalarda shartlarni tekshirish uchun qo'llaniladi. Masalan, shartli konstruktorda amallar yoki o'zgaruvchilar qiymati haqiqiylikini tekshiradi va natijadan qat'iy nazar shu yoki boshqa amallarni bajarilishini tekshiradi. Bu erda shart rost yoki yolg'on bo'lishi mumkin, chunki mantiqiy tip amallari va o'zgaruvchilar ko'rsatilgan.

**Misol.** Mantiqiy tiplarning qo'llanilishi.

```
<?php
// '++' amal tenglikka tekshiradi mantiqiy qiymatni qaytaradi
if ($know ++ False) { // agar $know qiymat false bo'lsa
echo "PHP dasturlash tilini o'rgan!";
}
if (!$know) { // xuddi Yuqoridagidek $know qiymati false bo'ladi
echo " PHP dasturlash tilini o'rgan!";
```

```

}
/* ++ amal $action o'zgaruvchi qiymati bilan "PHP dasturlash tilini o'rganish!" satrni ustma-ust
tushishini tekshiradi. Agar ustma-ust tushsa true qiymatni qaytaradi, boshqa holda false ni qaytaradi. Agar
true ni qaytarsa figurali qavs ichidagi amallar bajariladi. */
if ($action == "PHP dasturlash tilini o'rganish ")
{ echo "O'rganishni boshladim";}
?>

```

### ***Integer (butun) tipi.***

Bu tip butun sonlar to'plamidan  $Z = \{..., -2, -1, 0, 1, 2, ...\}$  birini qaytaradi. Butun sonlar xohishga qarab oldiga «-» yoki «+» belgilarni qo'yib sanoq sistemasini o'nlik, o'n oltilik yoki sakkizlik tizimlarida ko'rsatilgan bo'lishi mumkin.

Agar siz sakkizlik sanoq sistemasidan foydalanayotgan bo'lsangiz, oldindan 0 (nol) rahamini ko'rsatishingiz kerak. O'n oltilik sanoq sistemasida esa raqamlar oldiga 0x belgini qo'yish shart.

```

<?php
# o'nlik raham
$a = 1234;
# manfiy son
$a = -123;
# sakkizlik son (o'nlik sistemasidagi
# 83 songa ekvivalent)
$a = 0123;
# o'n oltilik son (o'nlik sistemasidagi
# 26 songa ekvivalent)
$a = 0x1A;
?>

```

Butun sonni o'lchami platformaga bog'liq, lekin qoidaga ko'ra maksimal qiymati ikki milliard (bu ishorali 32 bitli qiymat) atrofida bo'ladi. Ishorasiz butun sonni PHP dasturlash tili qo'llab quvvatlamaydi.

Agar siz butun son chegarasidan tashqari biror qiymat bersangiz interpretator bu sonni qo'zg'aluvchan vergulli songa o'zgartiradi. Xuddi shunday butun son chegarasidan tashqari chiqib ketadigan biror amal bajarsangiz ham bu sonni qo'zg'aluvchan vergulli songa o'zgartiriladi.

PHP dasturlash tilida butun sonlarni bo'lish amali mavjud emas. 1/2 ifoda qiymati qo'zg'aluvchan vergulli son 0.5 ga teng. Siz natijangizni butun tipga standart qoida asosida yoki round() funksiyasidan foydalangan taqdirda o'zgartirishingiz mumkin. O'zgaruvchini aniq bir tipga o'zgartirish uchun uning oldiga qavs ichida kerakli tipni yozish kerak bo'ladi. Masalan, \$a=0.5 o'zgaruvchini butun tipga o'zgartirish uchun (integer)(0.5) yoki (integer) \$a ko'rinishda yoki qisqartirilgan (int)(0.5) ko'rinishda yozish kerak bo'ladi. Bunday oshkora yangi tipga o'tish imkoniyati barcha ma'lumotlar tiplari uchun o'rinli bo'ladi (albatta, har doim ham qiymatni bir tipdan boshqasiga olib o'tish shart emas). Biz keltirilgan barcha tiplarni chuqur o'rganishimiz shart emas, chunki PHP dasturlash tili kontekstdan mustahil ravishda o'zi bu ishlarni bajaradi.

### ***Float (qo'zg'aluvchan vergulli son) tipi.***

Qo'zg'aluvchan vergulli sonlar (ular ikki karra aniqlik yoki haqiqiy sonlardir) quyidagi sintaksislar yordamida aniqlanishi mumkin:

```

<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>

```

Qo'zg'aluvchan vergulli sonni o'lchami ham platformaga bog'liq, lekin qoidaga ko'ra maksimal qiymati  $\sim 1.8e308$  aniqlik bilan 14 xonali raqam atrofida bo'ladi.

### **Resource (resurslar) tipi.**

Resurs – bu tashqi resursga (masalan, ma'lumotlar bazasi bilan bog'lanish ) ssilka orqali bog'langan maxsus o'zgaruvchidir. Resurslar maxsus funksiyalar (masalan, `mysql_connect()`, `pdf_new()` va shunga o'xshashlar) yordamida yaratiladi va foydalaniladi.

### **Null tipi.**

Maxsus NULL qiymati o'zgaruvchini qiymatga ega emasligi haqida ogohlantiradi.

O'zgaruvchi NULL qiymat qabul qiladi, agarda:

- unga o'zgarmas NULL (`$var = NULL`) o'zlashtirilgan bo'lsa;
- unga hech qanday qiymat berilmagan bo'lsa;
- u `unset()`funksiya yordamida tozalangan bo'lsa.

NULL tipli faqat bitta qiymati mavjud – registrga sezgirligi kichik NULL kalit so'zidir.

### **Masalaning echilishi.**

Endi bo'limning boshida qo'yilgan masalaga qaytsak. U turli sabablar bo'yicha har xil odamlarga tuzilgan maktubni jo'natishdan iborat edi. Bu masalani hal etish uchun o'rganilgan vositalardan – o'zgaruvchilar, amallar, o'zgarmaslar, satrlar va massivlardan foydalanishga harakat qilamiz. Ko'rsatilgan maktub qabul hiluvchiga bog'liq ravishda murojaat va holati o'zgaradi, shuning uchun tabiiy ravishda bu kattalikni o'zgaruvchi deb belgilaymiz. Bundan tashqari hodisalar va odamlar ko'p, shuning uchun massiv o'zgaruvchi tipidan foydalanish qulay. Maktub matni har doim o'zgarmas, shuning uchun uni o'zgarmas deb berish maqsadga muvofiqdir. Juda uzun va qo'pol satrlarni yozmaslik uchun satrlar konkatenatsiya (qo'shish) amalidan foydalanamiz. Shunday qilib, quyidagiga ega bo'lamiz:

```
<?
// bizning yozuvimiz
// o'zgarmas bo'lsin.
define("SIGN","Hurmat bilan, Azamat");
// odamlar va hodisalar massivini beramiz
$names = array("Ivan Ivanovich",
               "Petr Petrovich",
               "Semen Semenovich");
$events = array(
    "f" => "ochiq eshiklar kuni",
    "o" => "ko'rgazmaning ochilishi",
    "p" => "bitiruvchilar bali");

// taklifnoma matnini tuzamiz.
$str = "Hurmatli, $names[0]";
$str .= "<br> Sizni taklif etamiz ".
        $events["f"];
$str .= "<br>" . SIGN;
echo $str; // matnni ekranga chiqaramiz.
?>
```

Shunday qilib, bu bo'limda biz PHP dasturlash tilining asosiy sintaksisi bilan tanishib chiqdik, turli tipdagi o'zgaruvchilar, o'zgarmaslar va amallar bilan ishlashni, PHP dasturlash tilidagi mavjud tiplarini o'rgandik. Massivlar va satrlar ma'lumot tiplari haqida gap ketganda ularni chuqur va qismlarga ajratib o'rgandik. Bu konstruktsiyalar foydalanishga qulay va soddadir. Bular haqida keng ma'lumotlar keyingi bo'limlarda keltirilgan. Masalaning echilishi bor bilimlarga asoslangan holda soddacha echilgan, shuning uchun echim amaliyotda qo'llashga juda yaqin kelmaydi. Keyingi bo'limlarda bu kamchiliklarni to'g'rilaymiz va elektron maktubni umumiy shablonini yaratamiz.

## **4. Alternativ sintaksislar**

PHP dasturlash tili o'zining bir nechta if, while, for, foreach hamda switch boshqariladigan strukturalari uchun alternativ sintaksisini taqdim etadi. har bir holatda ochiladigan qavs ikki nuqtaga (:), yopiladigani esa mos ravishda endif, endwhile; va hokazolarga o'zgartiriladi.

Masalan, if shart operatori sintaksisini quyidagicha ifodalash mumkin:

if (ifoda) : bajariladigan\_blok endif;

Ma'nosi o'zgarmasdan qoladi: agar if shart operatori dumaloq qavsi ichidagi shart rost bo'lsa, ikki nuqtadan «>» to endif; buyruqigacha barcha kod bajariladi. Bunday sintaksisdan foydalanish html-kod ichida qurilgan php-kod uchun qulaydir.

**Misol.** Alternativ sintaksisdan foydalanish.

```
<?php
$names = array("Karim","Salim","Sodi?");
if ($names[0]++"Karim"):
?>
Salom, Karim!
<?php endif ?>
```

Agarda else hamda elseif konstruktsiyalaridan foydalanilsa, u holda ham alternativ sintaksisdan foydalansa bo'ladi:

```
<?php
if ($a ++ 5):
    print "a o'zgaruvchi 5 ga teng";
    print "...";
elseif ($a ++ 6):
    print "a o'zgaruvchi 6 ga teng ";
    print "!!!";
else:
    print "a o'zgaruvchi na 5 ga va na 6 ga teng ";
endif;
?>
```

### Nazorat savollari:

1. PHP kodni tuzilishini tushuntirib bering.
2. PHP da qanday ma'lumot tiplaridan foydalaniladi?
3. PHP da ifodalar qanday e'lon qilinadi?
4. Alternativ sintaksislar deganda nimani tushunasiz?
5. Boshqa dasturlash tillari bilan PHP dasturlash tilining tilining farqini tushuntiring?
6. PHP tilining qanday afzalliklari bor?

## 11-ma'ruza. PHP dasturlash tilida satr va massivlar bilan ishlash.

### Reja:

1. PHP dasturlash tilida satr tipi;
2. PHP dasturlash tilida massivlar tipi;
3. Kvadrat qavs sintaksisi yordamida massivni aniqlash;
4. Massivlar initsializatsiyasi;
5. Massivlarni ko'rib chiqish uchun foreach sikli;

6. Ko'p o'lchovli massivlar;
7. Massivlarni tartiblash funksiyalari.

**Kalit so'zlar:** PHP da satr tipi, massiv tipi, ko'p o'lchovli massivlar, kvadrat qavs sintaksisi, massiv initsializatsiyasi, foreach sikli, heredoc sintaksisi, tartiblash funksiyalari.

**Ishdan maqsad:** PHP dasturlash tili bilan tanishish va ishlash jarayonini o'rganish. PHP dasturlash tili bilan ishlash jarayonida satrlar va massivlar bilan ishlash ko'nikmalarini hosil qilish.

## 1. PHP dasturlash tilida String (satr) tipi

Satr – bu belgilar to'plamidir. PHP dasturlash tilida belgi bu bir bayt va 256 ta turli belgilar mavjud. PHP dasturlash tili Unicode tipidagi belgilarni qabul qilmaydi. PHP dasturlash tilida amalda satrlarga chegirma mavjud emas, shuning uchun satrlarni ishlatganda uning aniq uzunligi haqida o'ylash shart emas.

PHP dasturlash tilida satrlar uchta turli xil usullarda aniqlanadi:

- bittalik qo'shtirnoqlar yordamida ('');
- qo'shtirnoqlar yordamida ("");
- heredoc-sintaksisi yordamida.

### ***Bittali tirnoqlar***

Satrlarning aniqlashning oddiy usuli – u «\» bittali qo'shtirnoqlar ichida yoziladi. Agarda satr ichida ham bittali tirnoq ishlatishga to'g'ri kelib qolsa, bittali tirnoqdan oldin «\» belgini qo'yish, ya'ni uni ekranolash shart. Agarda «\» belgi bittali tirnoqdan oldin yoki satrning oxirida bo'lsa, u holda belgini ikkilantirish kerak, ya'ni «\\».

Agarda bittali tirnoq ichidagi satr ichida ixtiyoriy belgidan oldin («\» va «'» lardan farqli ravishda) teskari slesh «\» belgisi uchrasa, u holda uni oddiy belgi deb qarab barcha belgilarni o'z holicha ekranga chiqaradi. Shuning uchun teskari slesh «\» belgisini satr oxirida yopiladigan qo'shtirnoqdan avval turganini ekranolash shart.

PHP dasturlash tilida teskari slesh «\» belgisi bilan ifodalanadigan bir qator belgilar majmui mavjud. Ularni ketma-ketlikni boshqaruvchilar deb ataladi hamda ular maxsus vazifalarni bajaradi. Ular haqida keyinroq to'xtalib o'tamiz. O'zgaruvchilar va ketma-ketlikni boshqaruvchilar bittalik qo'shtirnoqlar satri ichida uchrashsa, ular o'rtasidagi farq ketma-ketlikni boshqaruvchilarni qayta ishlanmaydi.

### **Misol:**

```
<?php
echo 'Satrlar majmui';
//Ekranga chiqaradi: ' belgini chiqarish uchun undan oldin \ belgi
qo'yiladi.
echo ' Belgini \' chiqarish uchun undan oldin'
    ' //belgini qo'yish kerak';
//Ekranga chiqaradi: Siz shuni o'chirmoqchimisiz C:\*.*?
echo ' Siz shuni o'chirmoqchimisiz C:\\*.*?';
//Ekranga chiqaradi: Buni qo'ymang: \n
//yangi qatorga
echo ' Buni qo'ymang: \n yangi qatorga ';
//Ekranga chiqaradi: o'zgaruvchi $expand ham
//$either qo'yilmaydi
echo 'o'zgaruvchi $expand ham $either' .
    'qo'yilmaydi';
?>
```

## 2. PHP dasturlash tilida Array (massiv) tipi.



PHP dasturlash tilida massiv tipi tartiblangan kartalarga o'xshaydi va qiymatini kalitga o'zlashtiradigan tipdir. Bu tip bir necha yo'nalishlarda optimallashtiriladi, shuning uchun siz uni xususiy massiv, ro'yxat (vektor), xesh-jadvali (kartani amalga oshirish uchun ishlatiladi), stek, navbat va boshhalar sifatida foydalanishingiz mumkin. Modomiki, PHP dasturlash tilida bir massivni qiymatini boshqasiga o'zlashtirish uchun daraxtlardan foydalanasiz.

Massivlarni array() konstruktsiyasi yordamida aniqlanadi yoki elementlariga qiymat berish bilan aniqlanadi.

array() konstruktsiyasi yordamida aniqlash.

```
array ([key] => value,  
      [key1] => value1, ... )
```

PHP dasturlash tilining array() konstruktsiyasi vergul bilan ajratilgan juft parametrlar kalit => qiymat bilan ajratilgan. => belgi mos ravishda qiymat va uning kaliti o'rtasida aloqa o'rnatadi. Kalit butun son bo'lishi mumkin, uning qiymati esa PHP dasturlash tilidagi ixtiyoriy tipni qabul qilishi mumkin. Kalit raqamini biz ko'pincha indeks deb ataymiz. PHP dasturlash tilida indekslash noldan boshlanadi. Massiv elementining qiymatini olish uchun massiv nomi va kvadrat qavs ichida uning kaliti ko'rsatilishi kerak. Agar massiv kaliti standart butun son bo'lsa, u holda uning qiymatini butun son deb qarasa bo'ladi, aks holda u satr deb qaraladi. Shuning uchun \$a["1"] yozuv \$a[1] yozuvga teng kuchli, \$a["-1"] yozuv esa \$a[-1] yozuvga teng kuchli.

**Misol.** PHP dasturlash tilida massivlar.

```
<?php  
$books = array ("php" =>  
                "PHP users guide",  
                12 => true);  
echo $books["php"];  
//ekranga chiqaradi: "PHP users guide"  
echo $books[12];      //ekranga chiqaradi: 1  
?>
```

Agarda element uchun kalit berilmagan bo'lsa, u holda kalit sifatida kalitning maksimal qiymatiga bir qo'shib hisoblanadi. Agarda qiymati mavjud kalit ko'rsatilgan bo'lsa, u holda shu kalit qiymatini ekranga chiqaradi. PHP 4.3.0 dasturlash tili versiyasidan boshlab kalitning maksimal qiymati manfiy son deb qaralsa, u holda massivning keyingi kaliti nol (0) bo'ladi.

**Misol.** PHP dasturlash tilida massivlar.

```
<?php  
//$arr hamda $arr1 massivlar ekvivalentdir.  
$arr = array(5 => 43, 32, 56, "b" => 12);  
$arr1 = array(5 => 43, 6 => 32,  
              7 => 56, "b" => 12);  
?>
```

Agarda TRUE yoki FALSE kalit sifatida qo'llanilsa, u holda uning qiymati mos ravishda integer tipining bir va noliga o'zlashtiriladi. Agar NULL dan foydalanilsa, u holda kalit o'rniga bo'sh satr hosil bo'ladi. Bu bo'sh satrni kalit sifatida foydalansa bo'ladi, ammo uni qo'shtirnoqqa olish kerak bo'ladi. Bu usul bo'sh kvadrat qavsni ishlatish kabi emas. Massivlar yoki ob'ektlar kaliti sifatida foydalanish mumkin ham emas.

### 3. Kvadrat qavs sintaksisi yordamida massivni aniqlash.

Massivga qiymat berish orqali massiv yaratish mumkin. Biz yuqorida aytib o'tganimizdek, massiv elementi qiymatiga ega bo'lish uchun kvadrat qavs ichiga uning kaliti ko'rsatilishi kerak, masalan,

\$book["php"]. Agarda yangi kalit va yangi qiymat ko'rsatsangiz quyidagicha bo'ladi: \$book["new\_key"]="new\_value" hamda massivga yangi element qo'shiladi. Agarda kalitni ko'rsatmay faqat qiymatni o'zlashtirsak, ya'ni \$book[]="new\_value", u holda massivga yangi element qo'shiladi va uni kaliti mavjud maksimal qiymatga bir qo'shiladi. Agarda biz qiymat bergan massiv yaratilmagan bo'lsa, u holda biz qiymat bergandan keyin u yaratiladi.

```
<?
$books["key"]= value; //key kaliti bilan value qiymat $books massiviga
qo'shiladi
$books[] = value1; \* 13-kalit bilan value1 qiymati massivga qo'shiladi,
chunki
bizda kalitning maksimal qiymati 12 edi. *\
?>
```

Massivning aniq bir elementini o'zgartirish uchun uning shu kaliti bilan yangi qiymatga o'zlashtirish kerak. Massiv elementi kalitini o'zgartirish mumkin emas, faqat o'chirish (kalit va elementi juftligini) va yangi qo'shish mumkin xolos. Massiv elementini o'chirish uchun unset() Funksiyasidan foydalanish kerak.

```
<?php
$books = array ("php" =>"PHP users guide",12 => true);
$books[] = "Book about Perl"; \* 13-kalit(indeks) bilan yangi element
qo'shildi,
bu quyidagiga ekvivalent $books[13] = "Book about Perl";
$books["lisp"] = 123456; \* Bu massivga yangi "lisp" kalitli 123456 qiymatli
yangi element qo'shish*\
unset($books[12]); //Bu 12-kalitli elementni massivdan o'chirish
unset ($books); //massivni butunlay o'chirish
?>
```

Bo'sh kvadrat qavsdan foydalanganda kalitning maksimal qiymati massivda mavjud oxirgi qayta indekslangan kalitlar orasidan qidiriladi. Massivni array\_values() funksiyasi yordamida qayta indekslash mumkin.

**Misol.** Massivni qayta indekslaymiz.

```
<?php
$arr = array ("a","b","c");
\* "a", "b" va "c" qiymatli massivni yaratamiz. Bu erda kalit ko'rsatilmagan
Biroq mos ravishda ular 0,1,2 bo'ladi. *\
print_r($arr); //massivni ekranga chiqaramiz (kaliti va qiymatini)
unset($arr[0]);
unset($arr[1]);
unset($arr[2]);
//massivdan hamma elementini o'chiramiz
print_r($arr); //massivni ekranga chiqaramiz (kaliti va qiymatini)
$arr[] = "aa"; //massivga yangi element qo'shamiz. Uni indeksi(kaliti) 3
bo'ladi, 0 emas.
print_r($arr);
$arr =
array_values($arr); //massivni qayta indekslaymiz.
$arr[] = "bb"; //bu elementni kaliti 1 bo'ladi.
print_r($arr);
?>
```

Bu skriptning natijasi quyidagicha bo'ladi:

```

Array ( [0] => a [1] => b [2] => c )
Array ( )
Array ( [3] => aa )
Array ( [0] => aa [1] => bb )

```

#### 4. Massivlar initsializatsiyasi

PHP da massivlarni initsializatsiya qilishning 2 usuli mavjud. Birinchisi massiv elementlariga qiymat berishdan iborat:

```

<?
    $car[] = "passenger car";
    $car[] = "land-rover";
    echo($car[1]); // chiqaradi "land-rover"
?>

```

Massiv indeksini ochiq ko'rsatish mumkin:

```

<?
    $car[0] = "passenger car";
    $car[1] = "land-rover";
    echo($car[1]); // chiqaradi "land-rover"
?>

```

Agar massiv elementlarini e'lon qilishda oshkora indeksatsiyali va indeksatsiyasiz o'zgaruvchilar aralashib kelsa indeks berilmagan elementga ishlatilgan indekslar ichida eng kattasidan keyin keluvchi ruxsat berilgan indeksni beradi. Masalan agar biz yaratgan massiv elementlar indekslari 10, 20 va 30 bo'lsa va indeks ko'rsatmasdan Yangi element yaratsak, uning indeks avtomatik ravishda 31 bo'ladi:

```

<?
    $car[10] = "passenger car";
    $car[20] = "land-rover";
    $car[30] = "station-wagon";
    $car[] = "victoria";
    echo($car[31]);
?>

```

Alternativ usul **array()** konstruktsiyasidan foydalanishdan iborat:

```

<?
    $car = array("passenger car","land-rover");
    echo($car[1]); // chiqaradi "land-rover"
?>

```

Indekslarni oshkora ko'rsatish uchun => operator qo'llanadi:

```

<?
    $car = array("passenger car", 5 => "land-rover",
        "station-wagon","victoria");
    echo($car[0]); echo("<br>"); // chiqaradi "passenger car"
    echo($car[5]); echo("<br>"); // chiqaradi "land-rover"
    echo($car[6]); echo("<br>"); // chiqaradi "station-wagon"
    echo($car[7]); // chiqaradi "victoria"
?>

```

Massiv indeksleri satrlar ham bo'lishi mumkin:

```
<?
    $car = array("pc" => "passenger car", "lr" => "land-rover");
    echo($car["lr"]); echo("<br>"); // chiqaradi "land-rover"
    echo($car["pc"]); // chiqaradi "passenger car"
?>
```

## 5. Massivlarni ko'rib chiqish uchun foreach sikli

PHP4 da massiv elementlarini ko'rib chiqish uchun **foreach** operatoridan foydalanish mumkin. Bu operator sintaksisi:

```
foreach (array as [$key =>] $value)
{
    statements;
}
```

Busikl ma'nosi sodda: har bir element ko'rilganda uning indeksi **\$key** o'zgaruvchiga, qiymati bo'lsa **\$value** o'zgaruvchiga joylashtiriladi. Bu ikki o'zgaruvchilarning nomlari ixtiyoriydir.

**Misol:**

```
<?
    $car = array("passenger car", "land-rover",
        "station-wagon","victoria");
    foreach($car as $index => $val)
    {
        echo("$index -> $val <br>");
    }
?>
```

Sintaksisdan ko'rinib turibdiki, **\$key** o'zgaruvchidan foydalanish shart emas, shuning uchun tashlab yuborilishi mumkin:

```
<?
    echo(
        "available cars: <br> <ul>"
    );
    $car = array("passenger car", "land-rover",
        "station-wagon","victoria");
    foreach($car as $val)
    {
        echo("<li>$val</li>\n");
    }
    echo("</ul>");
?>
```

## 6. Ko'p o'lchovli massivlar

Ko'p o'lchovli massivlarni ko'rib chiqish uchun ichki joylashgan **array()** konstruktsiyasidan foydalaniladi. Ko'p o'lchovli massivlarni o'qib chiqish joylangan sikllar yordamida amalgam oshiriladi. Quyidagi scriptda ko'p o'lchovli massiv yaratish va ko'rib chiqish ko'rsatilgan.

**Misol:**

```

<?
    $ship = array(
        "Passenger ship" => array("Yacht", "Liner", "Ferry"),
        "War ship" => array("Battle-wagon", "Submarine", "Cruiser"),
        "Freight ship" => array("Tank vessel", "Dry-cargo ship", "Container
        cargo ship")
    );
    foreach($ship as $key => $type)
    {
        echo(
            "<h2>$key</h2>\n"."<ul>\n");
        foreach($type as $ship)
        {
            echo("\t<li>$ship</li>\n");
        }
    }
    echo("</ul>\n");
?>

```

Bu script bajarilish natijasi:

```

Passenger ship
Yacht
Liner
Ferry
War ship
Battle-wagon
Submarine
Cruiser
Freight ship
Tank vessel
Dry-cargo ship
Container cargo ship

```

Endi PHP da mavjud massivlar bilan ishlash funksiyalarini ko'ramiz. Biz massivlarni tartiblash funksiyalardan boshlaymiz. Lekin avval misollarimizda ko'p foydalanadigan uchta funksiyani ko'rib chiqamiz.

### 1. Funksiya count()

*Sintaksis:*

**int count(mixed var)**

Bu funksiya argument sifatida massivni qabul qilib, undagi elementlar sonini qaytaradi.

### 2. Funksiya in\_array()

*Sintaksis:*

**boolean in\_array(mixed needle, array haystack [, bool strict])**

Bu funksiya **haystack** massivda **needle** qiymatni qidiradi va agar u mavjud bo'lsa **true** qaytaradi, aks xolda **false** qaytaradi.

### 3. Funksiya reset()

*Sintaksis:*

**mixed reset(array array)**

Funksiya **reset()** massiv ko'rsatkichini birinchi elementga o'rnatadi va massiv birinchi elementi qiymatini qaytaradi.

Kichkina izox. PHP da har bir massiv joriy elementga ko'rsatkichga ega. Quyidagi **foreach** kabi konstruktsiyalar bilan ishlashda ko'rsatkich haqida o'ylash kerak emas, chunki foreach uni massiv boshiga o'rnatadi. Lekin boshqa massivlar funksiyalari masalan **prev()**, **next()**, massiv ko'rsatkichlarini suradi, bu esa **array\_walk()**, kabi qayta ishlashni ko'rsatkichni turgan joyidan boshlovchi funksiyalar uchun katta ahamiyatga ega.

Endi tartiblash bilan shug'ullanamiz.

## 7. Massivlarni tartiblash funksiyalari

Quyidagi tartiblash funksiyalari mavjud:

- **sort()**

Massivni o'sish bo'yicha tartiblash funksiyasi.

*Sintaksis:*

```
void sort(array array [, int sort_flags])
```

Funksiya **array** massivini o'sish bo'yicha tartiblaydi. Majburiy bo'lmagan element **sort\_flags** elementlar qanday tartiblanishi kerakligini ko'rsatadi (tartiblash bayroqlarini belgilaydi). Argumentning mumkin bo'lgan qiymatlari quyidagilar:

- ✓ **SORT\_REGULAR** – elementlar normal solishtiradi normalnoe sravnenie elementov (elementlarni "boricha" solishtiradi)
- ✓ **SORT\_NUMERIC** – elementlarni sonlar sifatida solishtiradi
- ✓ **SORT\_STRING** - elementlarni satrlar sifatida almashtiradi

Umuman olganda bu funksiya ro'yxatlarni tartiblash uchun mo'ljallangan. Ro'yxat deganda kalitlari nulldan boshlangan va bo'shliklarga ega bo'lmagan massiv tushuniladi. Funksiya **sort()** ixtiyoriy massivni ro'yxat deb qaraydi.

**Misol:**

```
<?
$arr = array("2", "1", "4", "3", "5");
sort($arr);
for($i=0; $i < count($arr); $i++)
{
    echo ("{$i}:$arr[$i] ");
}
// chiqaradi "0:1 1:2 2:3 3:4 4:5"
?>
```

*Natija:*

0:1 1:2 2:3 3:4 4:5

Agar siz satrlarni tartilayotgan bo'lsangiz, misol uchun massiv quyidagi ko'rinishga ega bo'lsa

```
array("one", "two", "abs", "three", "uic", "for", "five");
```

Bu ajoyib funksiya quyidagi natijani qaytaradi:

*Natija:*

0:abs 1:five 2:for 3:one 4:three 5:two 6:uic

Ya'ni satrlarni u alfa-beta tartibda, soddaroq aytganda birinchi harflari alfavitda kelishi bo'yicha tartiblaydi.

- **rsort()**

Massivlarni kamayish bo'yicha tartiblash.

*Sintaksis:*

```
void rsort(array arr [, int sort_flags])
```

Shunga o'xshash **sort()** funksiyasi faqat kamayish bo'yicha tartiblaydi. Oldingi **sort()** funksiyasi uchun ko'rilgan scriptni olamiz, faqat **sort(\$arr)** o'rniga **rsort(\$arr)** qo'yamiz.

*Natija:*

```
0:5 1:4 2:3 3:2 4:1
```

- **asort()**

Assotsiativ massivni o'sish bo'yicha tartiblash.

*Sintaksis:*

```
void asort(array arr [, int sort_flags])
```

Funksiya **asort()** berilgan **arr** massivni shunday tartiblaydiki uning qiymatlari alfavit tartibida (agar satr bo'lsa) yoki o'sish tartibda (sonlar uchun) tartibda joylashadi. Bu funksiyaning **sort()** funksiyasidan muxim farqi shundaki **asort()** funksiyasi qo'llanilganda kalitlar va ularga mos qiymatlar orasida bog'liklik saqlanadi, **sort()** funksiyasida bo'lsa bu bog'liklik uziladi.

**Misol:**

```
<?
```

```
$arr = array("a" =>"one","b" => "two","c" => "three","d" => "four");  
asort($arr);  
foreach($arr as $key => $val)  
{  
    echo (" $key => $val ");  
}
```

```
?>
```

*Natija:*

```
d => four a => one c => three b => two
```

Ko'rinib turibdiki "kalit-qiymat" bog'lanishlari saqlanib qolgan.

Ko'zda tutilgan bo'yicha **asort()** funksiyasi massivni alfavit bo'yicha tartiblaydi. Tartiblashbayroqlari **sort\_flags** qiymatlari **sort()** funksiyasi ta'rifida keltirilgan.

- **arsort()**

Assotsiativ massivlarni kamayish bo'yicha tartiblash.

*Sintaksis:*

```
void arsort(array arr [, int sort_flags])
```

Bu funksiya **asort()** funksiyasiga o'xshash, faqat u massivni o'sish bo'yicha emas kamayish bo'yicha tartiblaydi.

- **ksort()**

Massivlarni kalit o'sishi bo'yicha tartiblash.

*Sintaksis:*

```
int ksort(array arr [, int sort_flags])
```

Bu funksiyada tartiblash qiymatlar bo'yicha emas, balkim kalitlar bo'yicha o'sish tartibida amalga oshiriladi.

```
<?
```

```
$arr = array("a" =>"one","b" => "two","c" => "three","d" => "four");  
ksort($arr);  
foreach($arr as $key => $val)  
{  
    echo (" $key => $val ");  
}
```

```
}  
?>
```

*Natija:*

a => one b => two c => three d => four

- **krsort()**

Indekslar kamayishi bo'yicha massivlarni tartiblash.

*Sintaksis:*

int **krsort**(array **arr** [, int **sort\_flags**])

Xuddi **ksort()** funksiyaga o'xshash, faqat massivni kalitlar bo'yicha teskari tartibda (kamayish bo'yicha) tartiblaydi.

- **array\_reverse()**

Massiv elementlarini teskari joylashtirish.

*Sintaksis:*

array **array\_reverse**(array **arr** [, bool **preserve\_keys**])

Funksiya **array\_reverse()** elementlari parametrdagi berilgan **arr** massivi elementlariga nisbatan teskari joylashtirilgan massivni qaytaradi. Kalitlar va qiymatlar orasidagi bog'lanish saqlanib qoladi. Agar majburiy bo'lmagan parametr **preserve\_keys** ga **true** berilsa, kalitlar ham teskari tartibda joylashadi.

**Misol:**

```
<?  
$arr = array ("php", 4.0, array ("green", "red"));  
$result = array_reverse ($arr);  
echo "Massiv: <br>";  
foreach($result as $key => $val)  
{  
    echo ("$key => $val <br>");  
}  
echo("<br>");  
echo "Tartiblangan massiv: <br>";  
$result_keed = array_reverse ($arr, false);  
foreach($result_keed as $key => $val)  
{  
    echo ("$key => $val<br> ");  
}  
?>
```

Birinchi xolda:

*Natija:*

Massiv:

0 =>Array

1 =>4

2 =>php

Tartiblangan massiv:

0 =>Array

1 =>4

2 =>php

Agar ikkinchi parametrga **true** qiymat berilsa:

*Natija:*



Massiv:

0 => Array

1 => 4

2 => php

Tartiblangan massiv:

2 => Array

1 => 4

0 => php

- **shuffle()**

Massiv elementlarini tasodifiy joylashtirish.

*Sintaksis:*

void **shuffle**(array **arr**)

**shuffle()** funksiyasi **arr** massivi elementlarini tasodifiy aralashtiradi.

- **natsort()**

Tabiiy tartiblashni bajaradi.

*Sintaksis:*

void **natsort**(array **arr**)

Bunday tartiblashni satrlarni tartiblashda uchratgan edik. Tabiiy tartiblash deb elementlar tushunarli tartibda joylashga aytiladi.

**Misol:**

```
<?
    $array1 = $array2 = array("pict10.gif", "pict2.gif", "pict20.gif",
"pict1.gif");
    echo ("oddiy tartiblash:"); echo ("<br>");
    sort($array1);
    print_r($array1);
    echo ("<br>"); echo ("tabiiy tartiblash:"); echo ("<br>");
    natsort($array2);
    print_r($array2);
?>
```

*Natija:*

oddiy tartiblash:

Array ( [0] => pict1.gif [1] => pict10.gif [2] => pict2.gif [3] => pict20.gif )  
estestvennaya sortirovka:

Array ( [3] => pict1.gif [1] => pict2.gif [0] => pict10.gif [2] => pict20.gif )

**Nazorat savollari:**

1. PHP dasturlash tilida qanday tiplarni bilasiz?
2. PHP dasturlash tilida satr tipini tushuntirib bering?
3. PHP dasturlash tilida satrlar qanday aniqlanadi?
4. PHP da massivlar tushuntirib bering?
5. Massiv tipi qanday aniqlanadi?
6. Massivlarni iinitsializatsiya qilishning qanday usullarini bilasiz?
7. Qanday tartiblash funksiyalarini bilasiz?
8. Turli tartiblash funksiyalarini ishlashga doir misollar bilan tushuntiring?
9. Ko'p o'lchovli massivlar haqida ma'lumot bering?
10. Foreach siklini ishlash mohiyatini tushuntiring?

## 12-ma'ruza. Funksiyalar. Klasslar va ob'ektlar.Xatoliklar bilan ishlash. PHP funksiyalar.

### Reja:

1. Funksiyalar;
2. Funksiyalarning argumentlari;
3. O'zgaruvchan uzunlik argumentlari ro'yxatlari;
4. Funksiyalar ichida o'zgaruvchilardan foydalanish;
5. Funksiyaning o'zgaruvchilari;
6. Ichki joylashgan (ichma-ich) funksiyalar;
7. Sinflar va ob'ektlar;
8. O'zgaruvchilarni initsiallashtirish;
9. Ob'ektlar;

**Kalit so'zlar:** Funksiyalar, argumentlar, ro'yhatlar, o'zgaruvchilar, ichma-ich funksiyalar, klasslar va ob'yektlar, xatoliklar, o'zgaruvchilarni initsiallashtirish.

**Ishdan maqsad:** Talabalarga funksiyalar, klasslar va ob'yektlar, funksiyalarning argumentlari haqida ma'lumot berish va ular bilan ishlash ko'nikmalarini o'rgatish.

### 1. Funksiyalar

Funksiyalar nima uchun kerak? Bu savolga javob berish uchun, funksiya o'zi nima ekanligini tushunib olish lozim bo'ladi. Dasturlashda, xuddi matematikadagi kabi, unga bog'liq ko'pgina argumentlarning uning ko'pgina ma'nolarida aks etishidir. Demak, funksiya argumentning har bir ma'nolari jamlanmasi uchun uning bajargan ishi natijasi sifatida qandaydir ma'no qaytaradi. Funksiyalar nima uchun kerak, buni misollar bilan oydinlashtirishga qarakat qilamiz. Dasturlashdagi funksiya klassik misol – bu sonning faktorial ahamiyatini hisoblab beruvchi funksiya. Demak, biz unga son beramiz, u esa bizga uning faktorialini qaytaradi. Bunda biz faktorialini olishni xohlagan har bir son uchun aynan bir xil kodni qaytaravermaymiz – bu songa teng bo'lgan argumentli funksiyaning chaqirishning o'zi kifoya qiladi.

#### *Natural son faktorialini hisoblash funksiyasi*

##### Misol:

```
<?php
function fact($n){
if ($n++>0) return 1;
else return $fact = $n * fact($n-1);
}
echo fact(3);
//echo (3*2) deb yozish mumkin edi; lekin son katta bo'lsa,
echo fact(50);
//Funksiyadan foydalanish qulayroq, echo (50*49*48*...*3*2) deb yozgandan;
?>
```

Shu yo'l bilan biz biron-bir ma'lumotga bog'liqlik zaruriyati tug'ilgan amalni bajarganimizda, bu holda ham biz aynan shunday amallarni bajarishimiz oshirishimiz lozim bo'ladi, faqat boshqa boshlang'ich ma'lumotlardan foydalanamiz, funksiyalar mexanizmidan foydalanish– funksiya tanasi ko'rinishidagi amallar blokini taxt qilish, o'zgaruvchan ma'lumotlarni esa – uning parametrlari sifatida foydalanish qulayroq bo'ladi.

Funksiya topshirig'i (e'loni) umumiy tarzda qanday bo'lishini ko'ramiz. Funksiya quyidagi sintaksis yordamida aniqlanadi:

```
function Funksiya_nomi (1-parametr, 2-parametr, ... N-parametr) {
    Amallar bloki
    return "Funksiyaga aylanuvchi ma'no";
}
```

Agar php-dasturda to'g'ridan-to'g'ri yozilsa, hech narsani ishlab bo'lmaydi. Birinchidan, funksiya nomi funksiya parametrlari nomlari (1-parametr, 2-parametr va b.) PHP da nomlanish qoidalariga muvofiq kelishi kerak (unda yaxshisi kirillcha simvollarini ham ishlatmagan ma'qul). Funksiya nomlari registrga nisbatan sezuvchan bo'ladi. Ikkinchidan, funksiya parametrlari – tilning o'zgaruvchan qismlari, shuning uchun ularning har birining nomlari oldidan \$ belgisi turishi lozim bo'ladi. Parametrlar ro'yxatida hech qanday ko'p nuqtalarni qo'yish mumkin emas. Uchinchidan, amallar bloki so'zi bilan birga funksiya tanasida istalgan to'g'ri PHP-kod mavjud bo'lishi kerak (parametrlarga muvofiq bo'lishi majburiy emas). Va nihoyat, return kalit so'zidan so'ng tartibli php-ifoda kelishi lozim (ma'noga ega bo'lgan qandaydir simvollar). Bundan tashqari, funksiyada qaytariluvchi ma'no kabi parametrlar bo'lmasligi ham mumkin. Funksiyani to'g'ri e'lon qilishga misol – yuqorida keltirilgan faktorialni hisoblash funksiyasi.

Funksiya chaqirish qanday amalga oshadi? Funksiya nomi va yumaloq qavslar ichida uning parametrlari ma'nolari ro'yxati ko'rsatiladi, agar shundaylari mavjud bo'lsa:

### Misol:

```
<?php
    Funksiya_nomi ("1-parametr_uchun_ma'no", "2-parametr_uchun_ma'no ",...);
    /* Funksiyani chaqirishga misol - Funksiyani chaqirish faktorialni
hisoblash Yuqorida bor, 3 soni faktorialini hisoblash uchun u erda biz fact(3)
deb yozganmiz; u erda fact - chaqiriluvchi Funksiya nomi, a 3 -$n nomli uning
parametri ma'nosi *\
?>
```

Funksiyani qachon chaqirish mumkin? Bu halati savol bo'lib tuyulishi mumkin. Funksiyani uni aniqlangandan keyin chaqirish mumkin, ya'ni function f\_name(){...} blokidan pastda istalgan dastur qatorida. PHP3 da bu aynan shunday. Lekin PHP4 da bunday talab yo'q. Hamma gap interpretator olingan kodni qanday qayta ishlashida. Birgina istisno shartli ravishda aniqlanadigan Funksiyadan tashkil topadi (shartli operatorlar yoki boshqa Funksiyalar ichida). Funksiya shu tarzda aniqlangan taqdirda, uni aniqlash uni chaqirishdan oldin bajariladi.

### Misol. Shartli funksiya ichida funksiyaning aniqlash

```
<?
$make = true;
/* bu erda Make_event() ni chaqirish mumkin emas; Chunki u hali mavjud
emas, lekin Save_info() ni chaqirish mumkin *\
Save_info("Sobir","Sodiqov", "Men PHP kursini tanladim");
if ($make){
//Make_event() Funksiyasini aniqlash
function Make_event(){
echo "<p> Python<br> ni o'rganmoqchiman";
}
}
//endi Make_event() ni chaqirish mumkin
Make_event();
//Save_info funksiyasini aniqlanadi
function Save_info($first, $last, $message){
echo "<br>$message<br>";
echo "Ism: ". $first . " ". $last . "<br>";
}
```

```
Save_info("Murod","Yoqubov", "Men Lisp ni tanladim");
//Save_info ni bu erda ham chaqirish mumkin
?>
```

Agar funksiya dastur ichida aniqlangan bo'lsa, uni keyin qayta aniqlash yoki o'chirib tashlash mumkin emas. Funksiya nomlariga registr ta'sir qilmasligiga qaramasdan, yaxshisi funksiyaning aniqlash paytida berilgan nom bilan chaqirish mumkin bo'ladi.

**Misol.** Funksiya ichidagi funksiyaning aniqlash

```
<?php
/* ma'lumotlarni saqlash, ya'ni DataSave() funksiyasini chaqirish mumkin
emas. Uning to'g'riligi tekshirilmasdan oldin, ya'ni DataCheck() funksiyasi
chaqirilmasdan oldin bu mumkin emas.*\
DataCheck();
DataSave();
function DataCheck(){
//ma'lumotlar to'g'riligini tekshirish
function DataSave(){
//ma'lumotlarni saqlaymiz
} } ?>
```

Funksiya argumentlari, ularning ma'nolari va ishlatilishini batafsil ko'rib chiqamiz.

## 2. Funksiyalarning argumentlari

Har bir funksiya, avval aytganimizday, argumentlar ro'yxati bo'lishi mumkin. Bu argumentlar yordamida funksiya har xil ma'lumotlar beriladi (masalan, faktoriali hisoblanishi kerak bo'lganson ma'nosi). Har bir argument o'zgaruvchi va konstantaga ega bo'ladi.

Argumentlar yordamida ma'lumotlar funksiya uchun xil turli usullar bilan o'tkazilishi mumkin. Bu argumentlarni ma'nosiga ko'ra (o'zgarish holatda foydalaniladi), ilovalarga ko'ra va o'zgarish holatda argumentlarga ma'no berishga ko'ra o'tkazish. Bu usullarni atroflicha ko'rib chiqamiz.

Argument funksiya ma'nosiga ko'ra o'tkazilsa, funksiya ichidagi argument ma'nosining o'zgarishi uning funksiya tashqarisidagi ma'nosiga ta'sir qilmaydi. Funksiyaga uning argumentlarini o'zgartirishga yo'l qo'yish uchun ularni havolalarga ko'ra o'tkazish kerak. Buning uchun argument nomi oldidan funksiyaning aniqlashda ampersand "&" belgisini yozish kerak.

**Misol.** Argumentlarni havolasiga ko'ra o'tkazish

```
<?php
//qo'shimcha qilishi mumkin bo'lgan funksiyaning yozamiz checked so'zi
qatoriga
function add_label(&$data_str){
    $data_str .= "checked";
}
$str = "<input type=radio name=article ";
//bunday qator mavjud bo'lsin
echo $str."><br>";
//forma elementini keltiradi - belgilanmagan radio knopkasini
add_label($str);
//Funksiyaning chaqiramiz
echo $str."><br>";
//bu endi belgilangan radio knopkani keltiradi
?>
```

Funksiyada tinch holatda foydalanilayotgan argumentlar ma'nosini aniqlash mumkin. Ayni paytdagi ma'noning o'zi konstant ifoda bo'lishi, o'zgartirish va sinf vakili yoki boshqa funksiya chaqiruvi bo'lmasligi lozim.

Bizda informatsion xabar tuzuvchi funksiya, unga berilgan parametr ma'nosiga muvofiq tarzda o'zgaruvchi imzo bor. Agar parametr ma'nosi berilmagan bo'lsa, "Tashkiliy qo'mita" imzosidan foydalaniladi.

**Misol.** Tinch holatdagi argument ma'nosi

```
<?php
function Message($sign="Tash.qo'mita."){
//bu erda parametr sign ayni paytda "Tash.qo'mita" ma'nosiga ega
    echo "Keyingi yig'ilish ertaga bo'lib o'tadi.<br>";
    echo "$sign<br>";
}
Message();
//Parametersiz Funksiyani chaqiramiz. Bu holda imzo - Bu Tashkiliy
qo'mita
Message("Hurmat bilan Kamoliddin");
//Bu holda imzo "Hurmat bilan Kamoliddin." bo'ladi
?>
```

Bu skript ishining natijasi quyidagicha:

Keyingi yig'ilish ertaga bo'lib o'tadi.

Tashkiliy qo'mita.

Keyingi yig'ilish ertaga bo'lib o'tadi.

Hurmat bilan Kamoliddin.

Agar funksiyaning bir necha parametrlari bo'lsa, tinch holatda ma'no beriluvchi bu argumentlar funksiya aniqlanishida boshqa barcha argumentlardan keyin yozilishi kerak. Aks holda, agar bu argumentlar funksiyaning chiqarish paytida ko'zdan qochirilsa xato yuzaga kelishi ehtimoli bor.

Masalan, biz katalogga maqola tavsifini kiritmoqchimiz. Foydalanuvchi maqolaga uning nomlanishi, muallifi va qisqa tavsif kabi xarakteristikalarini keltirishi lozim bo'ladi. Agar Foydalanuvchi maqola muallifi ismini kiritmadi, u Murod Yoqubov deb olaylik.

```
<?php
function Add_article($title, $description, $author="Murod Yoqubov"){
echo "Maqolani katalogga kiritamiz: $title,";
echo "muallif $author";
echo "<br>qisqa tavsif: ";
echo "$description <br>";
}
Add_article("Informatika va biz","Bu maqola informatikaga oid ...",
"Zayniddin Saidov");
Add_article("Xarakterlar kim", "Bu maqola xarakterlar haqida ...");
?>
```

Skript ishi natijasi sifatida quyidagilarni olamiz:

Katalogga maqola kiritamiz:

Informatika va biz,

Muallif Murod Yoqubov.

Qisqa tavsif:

Bu maqola informatikaga oid...

Katalogga maqola kiritamiz:

Xarakterlar kim,  
Muallif Odil Ziyaev.  
Qisqa tavsif:  
Bu maqola xarakterlar haqida...

Agar biz quyidagicha yozsak:

```
<?php
function Add_article($author="Odil Ziyaev", $title, $description){
//... avvalgi misoldagi kabi amal
}
Add_article("Xarakterlar kim", "Bu maqola xarakterlar haqida...");
?>
```

Natija quyidagicha bo'ladi:

Warning: Missing argument 3 for add\_article() in c:\users\nina\tasks\func\def\_bad.php on line 2

### 3. O'zgaruvchan uzunlik argumentlari ro'yxatlari

PHP4 da argumentlarning o'zgaruvchan soni bilan funksiya tuzish mumkin. Ya'ni biz uni necha argumentlar bilan chaqirilishini bilmasdan, funksiya tuzamiz. Bu kabi funksiya yozish uchun hech qanday maxsus sintaksis kerak bo'lmaydi. Hammasi uning ichiga o'rnatilgan funksiyalar `func_num_args()`, `func_get_arg()`, `func_get_args()` yordami bilan qilinadi.

**func\_num\_args()** funksiyasi argumentlar sonini qaytaradi. Bu funksiya faqat foydalanuvchi funksiyasini aniqlash mobaynida foydalanishi mumkin. Agar u funksiyadan tashqarida paydo bo'lsa, interpretator ogohlantirish beradi.

**Misol.** `func_num_args()` funksiyasidan foydalanish

```
<?php
function DataCheck() {
    $n = func_num_args();
    echo "Funksiya argumentlari soni $n";
}
DataCheck();
//qatorni keltiradi "0 Funksiya argumentlari soni"
DataCheck(1,2,3);
//qatorni keltiradi "3-Funksiya argumentlari soni"
?>
```

**func\_get\_arg** funksiyasi (argument raqami to'raligicha) argumentni o'zgarishlar ro'yxatidan argumentlar funksiyasiga qaytaradi, uning tartib raqami `func_get_arg` parametri bilan beriladi. Funksiya argumentlari noldan boshlab hisoblanadi. `func_num_args()` kabi bu funksiya faqat biron-bir funksiyani aniqlashda foydalaniladi.

Argument raqami funksiyaga o'zgargan argumentlar sonidan ortib ketishi mumkin emas. Aks holda ogohlantirish umumlashtiriladi va `func_num_args()` Funksiyasiga False qiymat qaytadi.

Ma'lumotlarni tekshirish uchun funksiyaga uning argumentlarini tuzamiz. Agar funksiyaning birinchi argumenti – butun son, ikkinchisi – qator bo'lsa, tekshirish muvaffaqiyatli o'tdi, deb hisoblaymiz.

**Misol.** Ma'lumotlar tipini, uning argumentlarini tekshirish

```
<?
function DataCheck(){
    $check =true;
```

```

    $n = func_num_args();
    \* Funksiyaga o'zgargan argumentlar sonini tekshiramiz, birinchi o'zgargan
argument butun sonmi-yo'qmi *\
    if ($n>=1) if (!is_int(func_get_arg(0)))
        $check = false;
    \* tekshiramiz, ikkinchi o'zgargan argument qatormi-yo'qmi *\
    if ($n>=2)
        if (!is_string(func_get_arg(1)))
            $check = false;
return $check;
}
if (DataCheck(123,"text"))
    echo "Tekshiruv to'g'ri o'tdi<br>";
else echo "ma'lumotlar shartlarni qondirmaydi <br>";
if (DataCheck(324))
    echo "Tekshiruv to'g'ri o'tdi<br>";
else echo "ma'lumotlar shartlarni qondirmaydi <br>";
?>

```

Dastur natijasi quyidagicha bo'ladi.

Ma'lumotlar shartlarni qoniqtirmaydi. Tekshiruv to'g'ri o'tdi.

**func\_get\_args()** funksiyasi argumentlar ro'yxatidan tashkil topgan massiv qaytaradi. Massivning har bir elementi argumentga, funksiya o'zgarishiga to'g'ri keladi. Agar funksiya foydalanuvchi funksiyasi aniqligidan tashqarida foydalanilsa ogohlantirish umumlashtiriladi.

Avvalgi misolni ko'chiramiz, bu funksiyadan foydalanamiz. Funksiyani harakatlantiruvchi juft argument butun son ekanligini tekshiramiz:

### Misol:

```

<?
function DataCheck(){
    $check =true;
    $n = func_num_args();
    //Funksiyaga o'zgargan argumentlar soni
    $args = func_get_args();
    //Funksiya argumentlari massivi
    for ($i=0;$i<$n;$i++){
        $v = $args[$i];
        if ($i % 2 == 0){
            if (!is_int($v)) $check = false;
        }
    }
    //tekshiramiz, juft argument butunmi-yo'qmi
    return $check;
}
if (DataCheck(array("text", 324)))
    echo "Tekshiruv to'g'ri o'tdi<br>";
else echo "Ma'lumotlar shartlarni qoniqtirmaydi <br>";
?>

```

Bundan, **func\_num\_args()**, **func\_get\_arg()** va **func\_get\_args()** funksiya kombinatsiyasi funksiyalar o'zgaruvchan argumentlar ro'yxatiga ega bo'la olishi uchun foydalaniladi. Bu funksiyalar faqat PHP4 ga kiritilgan. PHP3 da shunday natijaga erishish uchun, argument sifatida massiv funksiyasidan foydalanish

mumkin bo'lad. Masalan, har bir toq funksiyalar parametri butun sonligini tekshiruvchi skriptni quyidagicha yozish mumkin:

```
<?
function DataCheck($params){
    $check =true;
    $n = count($params);
    //Funksiyaga o'zgargan argumentlar soni
    for ($i=0;$i<$n;$i++){
        $v = $params[$i];
        if ($i % 2 != 0){
            //tekshiramiz, toq argument butunmi-yo'qmi
            if (!is_int($v)) $check = false;
        }
    }
    return $check;
}
if (DataCheck("text", 324))
    echo "Tekshiruv to'g'ri o'tdi<br>";
else echo "Ma'lumotlar shartlarni qoniqtirmaydi<br>";
?>
```

#### 4. Funksiyalar ichida o'zgaruvchilardan foydalanish

Quyidagi o'zgaruvchilarning turlaridan foydalanish mumkin:

1. Global o'zgaruvchilar;
2. Statistik o'zgaruvchilar;
3. Qaytariluvchan ma'nolar;
4. Havolani qaytarish;

##### ***Global o'zgaruvchilar***

Funksiyalar ichida undan tashqarida berilgan o'zgaruvchilardan foydalanish uchun, bu o'zgaruvchilarni global deb e'lon qilish kerak. Buning uchun funksiya tanasida uning nomlarini global kalit so'zidan keyin keltirish lozim bo'ladi:

**Misol.** Global o'zgaruvchilar

```
global $var1, $var2;
<?
$a=1;
function Test_g(){
    global $a;
    $a = $a*2;
    echo ' $a=', $a Funksiya ishida natija;
}
echo 'Funksiyadan tashqarida $a=', $a, ', ' ;
Test_g();
echo "<br>";
echo Funksiyadan tashqarida $a=', $a, ', ' ;
Test_g();
?>
```

Bu skript ishidan quyidagi natijalarni olamiz:



\$a=2 Funksiyadan tashqarida, \$=2 Funksiya ishida natija

\$a=2 Funksiyadan tashqarida, \$=4 Funksiya ishida natija

O'zgaruvchi global deb e'lon qilinganda, aniq global o'zgaruvchi uchun havola tuziladi. Buning uchun bunday yozuv quyidagiga ekvivalent (GLOBALS massivi mavjud ko'rinish sohalariga muvofiq barcha global o'zgaruvchilarni o'z ichiga oladi):

```
$var1 = & $GLOBALS["var1"];
```

```
$var2 = & $GLOBALS["var2"];
```

Bundan kelib chiqadiki, \$var1 o'zgaruvchini o'chirish \$\_GLOBALS["var1"] global o'zgaruvchiini o'chirib tashlamaydi.

### ***Statistik o'zgaruvchilar***

O'zgaruvchilardan faqat funksiya ichida foydalanish uchun bunda uning ma'nosini saqlagan holda va funksiya chiqqandan so'ng, bu o'zgaruvchilarni statistik deb e'lon qilish kerak. Statistik o'zgaruvchilar faqat funksiyalar ichida ko'rinadi va dasturni yuklash funksiya doirasidan tashqariga chiqsa o'z ma'nosini yo'qotmaydi. Bu o'zgaruvchilarni e'lon qilish static kalit so'zi yordamida amalga oshiriladi:

```
static $var1, $var2;
```

Har qanday ma'no statistik o'zgaruvchi sifatida talqin qilinishi mumkin, faqat havola emas.

**Misol.** Statistik o'zgaruvchilardan foydalanish

```
<?
function Test_s(){
static $a = 1;
//ifoda yoki havolani o'zlashtirib bo'lmaydi
    $a = $a*2;
    echo $a;
}
Test_s(); //2 chiqadi
echo $a; //hech narsa chiqmaydi, zero $a faqat funksiya ichida kirish
yo'lagi bor
Test_s(); //$a=2 Funksiya ichida, shuning uchun funksiya ishi natijasi 4
soni bo'ladi
?>
```

### ***Qaytariluvchan ma'nolar***

Yuqorida misol qilib keltirilgan barcha funksiyalar biror-bir amal bajarishgan. Bunday hollardan tashqari, har qanday funksiya o'z ishi natijasi sifatida qandaydir qiymat qaytaradi. Bu return tasdig'i yordamida qilinadi. qaytariluvchan qiymat har qanday turda, shu jumladan, ro'yxat va ob'ektlar bo'lishi mumkin. Interpretator funksiya tanasida return komandasiga uchraganda, u darhol uni bajarishni to'xtatadi va funksiya chaqirilgan qatorga o'tib ketadi.

Masalan, inson yoshini qaytaruvchi funksiya tuzamiz. Agar inson vafot etmagan bo'lsa, yosh joriy yilga muvofiq hisoblanadi.

```
<?php
/* agar ikkinchi parametr true kabi hisoblab chi?ilsa, u vafot etgan
sanaday ko'rib chiqiladi, *\
function Age($birth, $is_dead){
    if ($is_dead) return $is_dead-$birth;
    else return date("Y")-$birth;
}
```

```
echo Age(1971, false); //vo`vedet 33
echo Age(1971, 2001); //vo`vedet 30
?>
```

Bu misolda return funksiyasidan foydalanmasa ham bo'ladi, shunchaki uni chiqarish funksiyasini echo ga almashtiriladi. Aksincha, agar biz funksiya biror-bir qiymat qaytaradigan qilsak (bu misolda inson yoshi), biz dasturda o'zgaruvchini bu funksiya qiymatini istalgan o'zgaruvchiga o'zlashtirishimiz mumkin.

```
$my_age = Age(1981, 2004);
```

Funksiya ishi natijasida faqat bitta qiymat qaytarilishi mumkin. Bir necha qiymatni qiymatlar ro'yxati qaytarilgan taqdirda olish mumkin (bir o'lchamli massiv). Biz inson yoshini kunigacha aniqlikda olmoqchimiz, deb hisoblaylik.

```
<?php
function Full_age($b_day, $b_month, $b_year)
{
    $y = date("Y");
    $m = intval(date("m"));
    $d = intval(date("d"));
    $b_month = intval($b_month);
    $b_day = intval($b_day);
    $b_year = intval($b_year);

    $day = ($b_day > $d ? 30 - $b_day + $d : $d - $b_day);
    $tmpMonth = ($b_day > $d ? -1 : 0);
    $month = ($b_month > $m + $tmpMonth
        ? $b_month + $tmpMonth - $m : $m+$tmpMonth - $b_month);
    $tmpYear = ($b_month > $m + $tmpMonth ? -1 : 0);
    if ($b_year > $y + $tmpYear)
    {
        $year = 0; $month = 0; $day q 0;
    }
    else
    {
        $year = $y + $tmpYear - $b_year;
    }
    return array ($day,$month,$year);
}
$age = Full_age("29","06","1986");
echo "Siz $age[2] yosh, $age[1] oylar va $age[0] kunlar";
?>
```

Funksiya bir necha qiymatlarni ularni dasturda qayta ishlash uchun qaytarganda, bir amal bilan ma'noni birdaniga bir necha o'zgaruvchilarni o'zlashtirishga imkon beruvchi list() til konstruksiyasidan foydalanish qulay bo'ladi. Masalan, yuqoridagi misolda funksiyani, uning qiymatiga o'zgartirish kiritmay qayta ishlash quyidagicha bo'lishi mumkin:

```
<?
//Full_age() Funksiya kiritish
list($day,$month,$year) = Full_age("07",
    "08","1974");
echo "Sizning yoshingiz $year, $month oy va
    $day kun";
?>
```

list() konstruksiyasini umuman o'zgaruvchini o'zlashtirish uchun istalgan massiv elementi qiymatidan foydalanish mumkin.

**Misol.** list() dan foydalanish

```
<?
$arr = array("first","second");
list($a,$b) = $arr;
    //o'zgaruvchi $a o'zlashtiriladi, birinchi massiv qiymati, $b - ikkinchi
echo $a, " ", $b;
//«first second» qatori keltiriladi
?>
```

### ***Havolani qaytarish***

Funksiya o'z ishi natijasida shuningdek havolani biror-bir o'zgaruvchiga qaytarishi mumkin. Bu funksiyani qanday o'zgaruvchi havolaga o'zlashtirish kerakligini aniqlash uchun foydalaniladi. Funksiyadan havola olish uchun, e'lon oldidan ampersand (&) belgisini yozish kerak bo'ladi va har safar funksiya chaqiruvi paytida uning nomi oldidan ham ampersand (&) yozish kerak bo'ladi. Ko'pincha funksiya havolani biror-bir global o'zgaruvchiga (yoki uning qismini – havolani global massiv elementiga), havolani statistik o'zgaruvchiga (yoki uning qismini) yoki havolani argumentlardan biriga qaytaradi, agar u havola bo'yicha berilgan bo'lsa.

**Misol.** Havolani qaytarish

```
<?
$a = 3; $b = 2;
function & ref($par){
    global $a, $b;
    if ($par % 2 ++ 0) return $b;
    else return $a;
}
$var =& ref(4);
echo $var, " i ", $b, "<br>"; //2 va 2 keltiriladi
$b = 10;
echo $var, " i ", $b, "<br>"; //10 va 10 keltiriladi
?>
```

Havola sintaksisidan foydalanishda bizning misoldagi \$var o'zgaruvchi o'zgaruvchining \$b qiymati \$ref qaytarilgan funksiyasiga ko'chirilmaydi, bu o'zgaruvchiga havola tuziladi. Demak, endi \$var va \$b teng kuchli o'zgaruvchilar va ular bir paytda o'zgartiriladi.

## **5. Funksiyaning o'zgaruvchilari**

PHP funksiyalar o'zgaruvchilariga ko'maklashadi. Bu degani, agar o'zgaruvchi nomi oddiy qavslar bilan tugasa, PHP shu kabi nomli funksiyani qidiradi va uni bajarishga harakat qiladi.

**Misol.** Funksiyalar o'zgaruvchilaridan foydalanish

```
<?
/* Ikkita oddiy funksiya tuzamiz: Add_sign – qatorga imzo qo'shadi va Show_text –matn qatorini
chiqarib beradi*\
function Add_sign($string,
    $sign="Hurmat bilan, Murod"){
```

```

        echo $string ." ".$sign;
    }
    function Show_text(){
        echo "Xabarni pochtadan jo'natish<br>";
    }
    $func = "Show_text"; //ma'noga ega o'zgaruvchi tuzamiz, u funksiya nomiga
teng Show_text
    $func(); //u Show_text funksiyani chaqiradi
    $func = "Add_sign"; //ma'noga ega o'zgaruvchi tuzamiz, u funksiya nomiga
teng Add_sign
    $func("hammaga salom <br>");
    //bu funksiyani chaqiradi Add_sign "hammaga salom" parametrli
?>

```

Bu misolda Show text Funksiya shunchaki matn qatorini chiqaradi. Agar echo maxsus funksiyasi mavjud bo'lsa, nega buning uchun alohida Funksiya tuzish kerak, deyish mumkin. Gap shundaki, echo(), print(), unset(), include() kabi funksiyalardan funksiyalar o'zgaruvchilari sifatida foydalanib bo'lmaydi. Ya'ni biz yozsak:

```

<?
$func = "echo ";
$func("TEXT");
?>

```

Interpretator xatoni ko'rsatadi:

```

Fatal error: Call to undefined function:
echo() in c:\users\nina\tasks\func\var_f.php on line 2

```

Shuning uchun yuqorida keltirib o'tilgan istalgan funksiyalardan o'zgaruvchilar funksiyasi sifatida foydalanish uchun yuqoridagi misoldagi yo'lni tutdik.

## 6. Ichki joylashgan (ichma-ich) funksiyalar

Foydalanuvchi tomonidan aniqlanadigan funksiyalar haqida gapirganda ichki joylashgan funksiyalar haqida gap ketmasligi mumkin emas. Yuqorida biz echo(), print(), date(), include() kabi ichki joylashgan funksiyalar bilan tanishdik. Bundan tashqari date() funksiyadan boshqa barcha funksiyalar PHP dasturlash tili konstruktsiyasiga ega. Ular PHP dasturlash tili yadrosiga joylashgan bo'lib, hech qanday modullar va qo'shimcha o'zgartirishlar talab etmaydi. Ammo shunday funksiyalar mavjudki, ularga turli fayl bibliotekalari va mos ravishda modullarni yuklamasdan iloji yo'q. Masalan, MySQL ma'lumotlar bazasi bilan ishlaydigan funksiyalardan foydalanish uchun shunday kengaytmali fayllarni qo'llab quvvatlaydigan komponentalari kerak. Oxirgi vaqtlarda bu funksiyalardan foydalanish uchun qo'shimcha komponentalar kerak emas, chunki ularning barchasi Hozirda PHP dasturlash tili yadrosiga kiritilgan.

## 7. Sinflar va ob'ektlar

Ob'ektga yo'naltirilgan dasturlashning asosiy tushunchalari – sinflar hamda ob'ektlardir. Bu tushunchalarni quyidagicha tushunish mumkin: ob'ekt – bu dasturda qo'llaniladigan tushuncha yoki biror fizik predmet haqida ma'lumot beradigan strukturalangan o'zgaruvchidir, sinflar esa bu ob'ektlarning tavsifi va ular ustida bajariladigan qarakatlardir.

PHP dasturlash tilida sinflar quyidagi sintaksis yordamida aniqlanadi:

```

class Sinf_nomi{
    var $xususiyat_nomi;
    \* xususiyatlar ro'yxati *\

```

```

function metod_nomi( ){
    \* usullarning tanasi *\
}
\*usullar ro'yxati*\
}

```

Sinf ob'ektlari xususiyatlari nomi var kalit so'zi yordamida e'lon qilinadi, berilgan sinf ob'ektlariga qo'llanilgan usullar Funksiya sifatida ishlatiladi. Sinf tanasi ichida this kalit so'zi yordamida taqdim qilinayotgan joriy sinfga murojaatni amalga oshirish mumkin.

Masalan, biz maqola kategoriyasini tasvirlovchi sinf tuzishimiz kerak. har bir maqolaning nomi, muallifi va qisqa mazmuni kabi xususiyatlari bor. Biz maqola bilan qanday amal bajarmoqchimiz? Biz sanab o'tilgan xususiyatlarga ma'no berishimiz, maqolani brauzerda ko'rsatishimiz kerak bo'ladi. Shunda bu sinfning ifodalanishi quyidagicha holatda bo'ladi:

```

<?
class Articles { //Maqola sinfini tuzamiz
    var $title;
    var $author;
    var $description;
    //maqola atributi ma'nosini o'zlashtiruvchi usul
    function make_article($t, $a, $d){
        $this->title =$t;
        $this->author = $a;
        $this->description = $d;
    }
    //sinf nusxasini ifodalash uchun usul
    function show_article(){
        $art =$this->title . "<br>" .
            $this->description .
            "<br>Muallif: " . $this->author;
        echo $art;
    } } ?>

```

Shunday qilib “maqola” turidagi fizik ob'ektlarni tasvirlash uchun biz uch o'zgartuvchidan tashkil topgan, maqola xarakteristikasini o'zida jamlagan Articles nomli sinf va muayyan maqola tuzish va uni tasvirlash uchun ikkita funksiya tuzdik.

Ma'lumki, PHP bilan ishlash davriy holatda HTML rejimida yuklanishi mumkin. Bu holda dastur bir necha kodning bo'laklari(bloklar)dan tashkil topadi. Sinfni ifodalash php-kodning har xil bloklari bo'yicha va qolaversa har xil fayllar bo'yicha tarqatilmasligi kerak. Ya'ni quyidagicha yozsak:

```

<?php
class Articles { //Sinfni tasvirlashning boshi
    var $title;
?>
<?php //sinfni tasvirlashning davomi
    function show_article(){ //usulning tarkibi
    }
} //sinfni tasvirlashning yakuni
?>

```

bunda dastur tartibli ishlaydi.

Sinf nomi masalasida ayrim narsalarni e'tiborda tutish kerak. Sinfning nomi PHP tilidagi ob'ektlar nomlanishi qoidalariga javob berishi lozim, lekin bir qator nomlar borki, texnik mutaxassislar tomonidan o'z maqsadi uchun zaxira qilinadi. Birinchi navbatda bu nomlar “\_” quyi chiziqdan boshlanuvchilardir. Sinflar

va funksiyalar tuzish uchun bu kabi nomlarni ishlatmaslik kerak. Bundan tashqari stdClass nomi zaxira qilingan, zero u PHP surilgichi ichida ishlatiladi.

## 8. O'zgaruvchilarni initsiallashtirish

Ba'zan ayrim sinf atributlariga ma'noni sinf ishtirokchisini tuzish bilanoq o'zlashtirish kerak bo'ladi. Biz maqola sinfini tuzganimizda, sinf atributlari (xususiyatlari) ma'nolarini o'zlashtirish uchun maxsus funksiya make\_article() dan foydalandik. Umuman olganda, biz to'g'ri yo'l tutmadik, chunki "velosiped ixtirosi" bilan shug'ullandik. Sinf atributlarining boshlang'ich ma'nolarini berish uchun maxsus ikkita standart usul mavjud. PHP4da ma'noni var operatori yoki konstruktor funksiyasi yordamida initsiallashtirish mumkin. var yordamida faqat konstant ma'nolarni initsiallashtirish mumkin. Konstant bo'lmagan ma'nolarni berish uchun ob'ekt sinfdan ajrab chiqqanda o'z-o'zidan ishga tushuvchi konstruktor funksiyasidan foydalaniladi. Konstruktor-funksiya u ifodalangan butun sinfga mos keluvchi nomga ega bo'lishi kerak.

Misol. "maqola" deb nomlangan ob'ektni tuzishda uning xususiyatlarini quyidagicha belgilash mumkin: mualliflar – "Kamolov" satriga teng, nomlanish va qisqa mazmun - \$\_POST global massivi elementlariga mos, maqola nashri – mazkur sanada.

```
<?
class Articles { //maqola sinfini tuzish
    var $title;
    var $author = "Kamolov";
    var $description;
    var $published;          //sinf atributi ma'nosini o'zlashtiruvchi usul
    function Articles(){
        $this->title = $_POST["title"];
        $this->description = $_POST["description"];
        $this->published = date("Y-m-d");
    } } ?>
```

PHP3 va PHP4 da konstruktorlar har xil ishlashini hisobga olish kerak. Funksiya PHP3 da, agar u sinfniki kabi nomga ega bo'lsa, konstruktorga aylanadi, PHP4 da esa – agar u ifodalangan sinfniki kabi nomga ega bo'lsa shunday bo'ladi. Bir sinf boshqasini kengaytirganda va xususiyatlarning va baza sinflar usullarining ergashishida usullar orasidagi farq ko'rinib turibdi. Lekin biz bu haqida biroz keyinroq gapiramiz. PHP5da sinf konstruktori \_\_construct deb nomlanadi. Bundan tashqari, PHPda destruktorga – ob'ektni yo'q qilishda o'z-o'zidan ishga tushuvchi Funksiyalar paydo bo'ldi. PHP5 da Funksiya-destruktor destruct deb nomlanishi kerak bo'ladi.

## 9. Ob'ektlar

### Object (ob'ektlar) tipi

Ob'ektlar – ob'ektga yo'naltirilgan dasturlashdan kirib kelgan ma'lumot tipidir. Ob'ektga yo'naltirilgan dasturlash tamoyiliga ko'ra, sinf – aniq xossalarga ega va ular bilan ishlaydigan metodli ob'ektlar to'plami. Ob'ekt esa mos ravishda sinf nusxasidir. Masalan, dasturchilar – bu dasturni tuzuvchi, kompyuter adabiyotlarini o'rganadigan odamlar sinfi va bundan tashqari hamma odamlar qatori ism va familiyasi mavjud. Endi agarda biror aniq dasturchi – Azamat Boboevni olsak, u holda uni shu xossaga ega bo'lgan dasturchi sinfini ob'ekti sifatida qarash mumkin va u ham dastur tuzadi, hamda ismi mavjud va boshqalar.

PHP dasturlash tilida ob'ekt metodiga murojaat -> amalidan foydalaniladi. Ob'ektni initsializatsiya qilishda ob'ektni o'zgaruvchan nusxasini yaratadigan new ifodasidan foydalaniladi.

```
<?php      //odam sinfini yaratamiz.
class Person
{          //PHP dasturlash tilini o'rganadigan odam metodi
    function know_php() {
```

```

        echo "Endi men PHP dasturlash tilini bilaman!";
    }
}
$bob = new Person; //odam sinfini ob'ektini yaratamiz.
$bob -> know_php(); //uni PHP tiliga o'rgatamiz.
?>

```

Sinf – bu ob'ekt tipidagi ma'lumotlarning bir turidagi ifodalanishidir. Sinflar real o'zgaruvchilar uchun shablon vazifasini o'taydi. Kerakli tipdagi o'zgartuvchi new operatori yordamida sinfdan tuziladi. Ob'ektni tuzib, biz barcha usullarni qo'llashimiz va barcha sinf ifodasida ko'rsatib o'tilgan xususiyatlarni olishimiz mumkin bo'ladi. Buning uchun quyidagicha sintaksisdan foydalaniladi: \$ob'ekt\_nomi->xususiyat yoki \_nomi\$ob'ekt\_nomi->usulning\_nomlanishi(argumentlar ro'yxati). Xususiyatlar yoki usular nomlari oldidan \$ belgisi qo'yilmaydi.

**Misol:** Ob'ekt usullari va xususiyatlariga erkin kirish (dostup)

```

<?php
$art = new Articles;      //ob'ekt tuzamiz $art
echo ($art ->title);      //ob'ekga nomlanish beramiz $art
$another_art = new Articles;    //ob'ekt tuzamiz $another_art
$another_art->show_article(); //ob'ektning brauzerdagi ifodasi uchun
usulni chaqiramiz
?>

```

Sinfnig har bir ob'ekti aynan bir xil xususiyatlar va usullarga ega bo'ladi. Demak, \$art ob'ektda va \$another\_art ob'ektda title, description, author xususiyatlari va Articles(), show\_article() usullari mavjud. Lekin bular ikki xil ob'ektlar. Ob'ektni fayllar sistemasidagi direktoriya deb hisoblaymiz, uning xarakteristikasi esa – bu direktoriyadagi fayllar singari bo'lsin. Aniqki, har bir direktoriyada bir xil fayllar yotishi mumkin, lekin shunday bo'lsa-da, ular har xil direktoriyalarda saqlanayotgani uchun har xil hisoblanishi mumkin. Xuddi shuningdek, xususiyatlar va usullar ham, agar ular turli ob'ektlarga qo'llaniladigan bo'lsa, har xil hisoblanadi. Yuqori bosqichdagi direktoriyadan kerakli faylni olish uchun bu faylga yo'lni batafsil yozib chiqamiz. Sinflar bilan ishlash mobaynida biz chaqirishni istagan Funksiyaning nomini to'liq yozishimiz kerak bo'ladi. PHP dagi Yuqori bosqich direktoriyalariga global o'zgaruvchilarning bo'sh o'rnini bo'ladi, yo'l esa -> taqsimlovchisi yordamida ko'rsatiladi. Shu tarzda \$art->title va \$another\_art->title nomlari ikki xil turli o'zgaruvchilarni anglatadi. PHP da o'zgaruvchi nom oldidan faqat bitta dollar belgisiga ega bo'ladi, shuning uchun \$art->\$title ko'rinishida yozish mumkin emas. Bu konstruktsiya \$art ob'ektining title xususiyatiga murojaat sifatida ko'rib chiqilmaydi, \$title o'zgartuvchi ko'rinishida berilgan nomli xususiyat sifatida ko'riladi (masalan, \$art->"").

**Misol:** Xususiyatlar ma'nosini o'rnatish

```

<?php
$art->title = " Internet ga kirish"; //ob'ekt xususiyati ma'nosini shunday
o'rnatish mumkin
$art->$title = "Internet ga kirish"; //ob'ekt xususiyati ma'nosini bunday
o'rnatib bo'lmaydi
$property = "title";
$art->$property = "Internet ga kirish"; //ob'ekt xususiyati ma'nosini
shunday o'rnatish mumkin
?>

```

Sinfini tuzib, bu sinfnig ob'ekti qanday nomga ega bo'lishini bila olmaymiz, qolaversa ob'ektlar juda ko'p bo'lishi va ularning barchasi har xil nomga ega bo'lishi mumkin. Sinfini yuzaga chiqarish ichida ob'ektga qanday munosabatda bo'lishni bilmaymiz. Sinf yuzaga chiqishi ichida funksiyalar va o'zgaruvchilarga erkin kirish uchun, \$this o'rindosh o'zgaruvchisidan foydalanish kerak. Masalan, \$this-

>title shunday sinf ob'ektining title ini qaytaradi. Ba'zan bu o'zgaruvchini "mening xususiy mulkim" (xususiyatga munosabat tariqasida) deb o'qishni taklif qilinadi.

### **Nazorat savollari:**

1. PHP dasturlash tilida funksiyalarni tushuntirib bering.
2. PHP da funksiyalar argumentlari qanday e'lon qilinadi?
3. PHP da o'zgaruvchilar e'lonini tushuntiring.
4. Ichma-ich funksiyalar nima va ularning ahamiyati haqida gapiring?
5. PHP dasturlash tilida sinflar va ob'ektlar nima?
6. Funksiyani chaqirish qanday amalga oshiriladi?
7. PHP dasturlash tilida funksiyalar nima uchun kerak?
8. Funksiyalar ichida qanday turdagi o'zgaruvchilardan foydalanish mumkin?
9. PHP dasturlash tilida klasslar va ob'ektlar qanday e'lon qilinadi?
10. Sinfning ob'ektlari qanday xususiyatlarga ega bo'lishi mumkin?
11. PHP dasturlash tili konstruktsiyasiga ega bo'lmagan funksiyani ayting va u haqida ma'lumot bering?

## **13-ma'ruza.PHP dasturlash tilida ma'lumotlar bazalari bilan ishlash. MySQL ma'lumotlar bazasi. Fayllar bilan ishlash.**

### **Reja:**

1. Ma'lumotlar bazasi haqida tushuncha;
2. Ma'lumotlar bazasi interfeysi;
3. Ma'lumotlar bazasi bilan bog'lanish;
4. Ma'lumotlar bazasi ustida bajariladigan amallar;
5. Fayllar bilan ishlash.

**Kalit so'zlar:** Ma'lumotlar bazasi, MySQL dasturi, PHP da MySQL bilan bog'lanish, fayllar, fayllar bilan ishlash, MB amallari.

**Ishdan maqsad:** PHP dasturlash tilida ma'lumotlar bazasi va fayllar bilan ishlash bilim va malakasiga ega bo'lishni o'rganish va uni amaliyotga tatqiq qilish.

### **1. Ma'lumotlar bazasi haqida tushuncha**

Ushbu bo'lim PHP va MySQL MBBS o'rtasidagi hamkorlik usullari bilan tanishishga mo'ljallangan. Asosiy e'tibor ma'lumotlar bazasi bilan bog'lanishni o'rnatish, so'rovlar jo'natish funksiyalari va javoblarni (mysql\_connect, mysql\_query, mysql\_result, mysql\_num\_rows, mysql\_close) qayta ishlashga qaratiladi. Misol sifatida virtual tarix muzeyi ma'muriyati uchun web-interfeys tuzish masalasini ko'raylik. PHP distributivida MySQL ma'lumotlar bazasi bilan ishlash uchun mo'ljallangan Funksiyalar mavjud. Bunda bu funksiyalarning MySQL dagi ba'zi bir ma'lumotlar bazasini tasvirlash va to'ldirish maqsadida web-interfeyslarni tuzish imkonini beruvchi funksiyalar bilan tanishamiz. Ma'lumotlar bazasiga ma'lumotlarni qo'shish uchun web-interfeys bilan ishlashda bu ma'lumotlarni shunchaki html-formaga kiritish va ularni serverga jo'natish kerak bo'ladi.

Namoyish etishda bu interfeysni virtual muzey eksponatlari haqidagi ma'lumotlar saqlanadigan Artifacts jadvallari uchun tuzamiz. Artifacts kolleksiyasidagi har bir eksponat quyidagi xarakteristika yordamida tasvirlanishini eslatib o'tamiz:



- nom (title);
- muallif (author);
- ifoda (description);
- o'rindosh nom (alternative);
- tasvir (photo).

Nomlanish va o'rindosh nomlanish uzunasiga 255 belgidan kam satr (ya'ni VARCHAR(255)), tasvirlash – matnli maydon (TEXT turiga mansub) hisoblanadi, “muallif” va “tasvir” maydonlarida esa Persons kolleksiyasidan muallifning identifikatorlari va Images kolleksiyasidan eksponat tasvirlariga muvofiq mavjud bo'ladi.

## 2. Ma'lumotlar bazasi interfeysi

Ma'lumotlar bazasidagi mavjud jadval strukturasi (ya'ni uning maydonlari jamlanmasini) html-formada tasvirlash uchun quyidagi tarkibiy topshiriqlarni rejalashtirish mumkin:

- MB bilan ulanishni o'rnatish;
- MB ishini tanlash;
- Jadval maydonlari ro'yxatini olish;
- html-formada maydonlarni tasvirlash.

Bundan keyin formaga kiritilgan ma'lumotlarni ma'lumotlar bazasiga kiritish mumkin.

## 3. Ma'lumotlar bazasi bilan bog'lanish (MySQL dasturi misolida)

### *Aloqa o'rnatish*

Ma'lumotlar bazasi bilan aloqa o'rnatish uchun **mysql\_connect** funksiyasidan foydalaniladi.

mysql\_connect sintaksisi

mysql\_connect resursi (“server qatori”, “username”, “password”)

Bu Funksiya MySQL serveri bilan aloqa o'rnatadi va bu aloqaga ko'rsatkich qaytaradi yoki muvaffaqiyatsiz chiqqanda FALSE ko'rsatadi. Odatda quyidagi parametrlar qiymati e'lon qilinadi:

```
server = 'localhost:3306'
```

```
username = server jarayoni egasidan foydalanuvchi ismi
```

```
password = bo'sh parol
```

Server bilan ulanish, agar u bungacha mysql\_close() yordamida yopilmagan bo'lsa, skriptni amalga oshirish tugallanishida baza bilan aloqa yopiladi.

### **Misol:**

```
<?
$conn = mysql_connect("localhost", "nina","123") or die("Ulanishni amalga
oshirib bo'lmaydi: ". mysql_error());
echo "Ulanish amalga oshdi";
mysql_close($conn);
?>
mysql_connect amali
shell>mysql -u nina -p123 buyrug'i bilan teng kuchli.
```

## 4. Ma'lumotlar bazasi ustida bajariladigan amallar

### *Ma'lumotlar bazalarini tanlash*

MySQL da ma'lumotlar bazasini tanlash use buyrug'i yordamida amalga oshiriladi:  
mysql>use book;

PHP da buning uchun mysql\_select\_db funksiyasi mavjud.  
mysql\_select\_db: sintaksisi  
mantiqiy mysql\_select\_db (database\_name qatori);

Bu funksiya TRUE qiymatni ma'lumotlar bazasini muvaffaqiyatli tanlanganda qaytaradi va FALSE ni esa – aksincha bo'lganda.

**Misol:** Book ma'lumotlar bazasini tanlash

```
<?
$conn = mysql_connect("localhost","user","123") or die("Ulanishni amalga
oshirib bo'lmaydi: ". mysql_error());
echo "Ulanish amalga oshdi";
mysql_select_db("book");
?>
```

### ***Jadval maydonlari ro'yxatini olish***

PHP da ma'lumotlar bazasi bilan bog'lanilgandan so'ng, undagi jadvallar ro'yxatini olish mumkin. Bu funksiya - mysql\_list\_fields.

mysql\_list\_fields sintaksisi  
mysql\_list\_fields (database\_name qatori, table\_name qatori)

mysql\_field\_name Funksiyasi so'rov amalga oshirilishi natijasida olingan maydon nomini qaytaradi. mysql\_field\_len funksiyasi maydon uzunligini qaytaradi. mysql\_field\_type Funksiyasi maydon tipini qaytaradi, mysql\_field\_flags Funksiyasi esa probel bilan yozilgan maydon bayroqlari ro'yxatini qaytaradi. Maydon tiplari int, real, string, blob va h. bo'lishi mumkin. Bayroqlar not\_null, primary\_key, unique\_key, blob, auto\_increment va h. bo'lishi mumkin.

Bu barcha buyruqlar sintaksisi bir xil:

mysql\_field\_name (result qatori, butun field\_offset) resursi;  
mysql\_field\_type (result qatori, butun field\_offset) resursi;  
mysql\_field\_flags (result qatori, butun field\_offset) resursi;  
mysql\_field\_len ( result qatori, butun field\_offset)

Bu erda result – bu so'rov natijasi identifikatori (masalan, mysql\_list\_fields yoki mysql\_query Funksiyalar bilan jo'natilgan so'rov), field\_offset esa – natijadagi maydonning tartib raqami.

mysql\_num\_rows(result resursi) buyrug'i result ning ko'pgina natijalari qatorlari miqdorini qaytaradi.

**Misol:** Artifacts (eksponatlar kolleksiyasi) jadvali maydonlari ro'yxatini olish.

```
<?
$conn = mysql_connect("localhost","user","123") or die("Aloqa o'rnatib
bo'lmaydi: ". mysql_error());
echo "Aloqa o'rnatildi";
mysql_select_db("book");
$list_f = mysql_list_fields ("book","Artifacts",$conn);
$n = mysql_num_fields($list_f);
for($i=0;$i<$n; $i++){
    $type = mysql_field_type($list_f, $i);
    $name_f = mysql_field_name($list_f,$i);
```

```

    $len = mysql_field_len($list_f, $i);
    $flags_str = mysql_field_flags ($list_f, $i);
    echo "<br>Maydon nomi: ". $name_f;
    echo "<br>Maydon turi: ". $type;
    echo "<br>Maydon uzunligi: ". $len;
    echo "<br>Maydon bayroqlari qatori: " . $flags_str . "<br>";
}
?>

```

Natija sifatida taxminan quyidagilarni olish mumkin (albatta, jadvalda ikkita maydon bo'lganda):

- Maydon nomi: id
- Maydon turi: int
- Maydon uzunligi: 11
- Maydon bayroqlari qatori:
- not\_null primary\_key auto\_increment
- Maydon nomi: title
- Maydon turi: string
- Maydon uzunligi: 255
- Maydon bayroqlari qatori:

### ***HTML-formada maydonlar ro'yxatining aks etishi***

Maydon haqidagi ma'lumotni html-forma elementida aks ettiramiz. BLOB turidagi elementlarni textarea ga o'tkazamiz (TEXT turida biz tuzgan description maydoni BLOB tipiga ega), raqamlar va qatorlarni <input type=text> matnli qatorlarida aks ettiramiz, avtoinkrement belgisiga ega elementni esa umuman aks ettirmaymiz, chunki uning ma'nosi o'z-o'zidan o'rnatiladi.

Buning uchun explode funksiyasidan foydalaniladi:

explode: sintaksisi

explode massivi (separator qatori, string qatori , int limit)

Bu funksiya string qatorini separator taqsimlagichi yordamida qismlarga bo'ladi va olingan qatorlar massivini qaytaradi.

Bizning holatda taqsimlagich sifatida probel " " ni olish kera, bo'lish uchun boshlang'ich qator sifatida esa – maydon bayroqlari qatorini.

**Misol.** Artifacts jadvaliga ma'lumot kiritish uchun forma (index.php)

```

<?php
    $dblocation = "localhost";
    $dbname = "book";
    $dbuser = "root";
    $dbpasswd = "";
    $dbcnx = @mysql_connect($dblocation,$dbuser,$dbpasswd);
    @mysql_select_db($dbname,$dbcnx);
    if(isset($_POST['save_hide'])) {
    $title=($_POST['title']);
    $query = "INSERT INTO Artifacts (title) VALUES ('$title')";
    if(@mysql_query($query))
    {
        echo "<HTML><HEAD><META HTTP-EQUIV='Refresh' CONTENT='0';
URL=index.php'><\HEAD><\HTML>";
    } else { print mysql_error(); error("Ma'lumotni bazaga ezishda
xatolik");}
    } else {

```

```
?>
<TABLE width="100%" align=center border=0><TR><TD>
<form method="POST" name="save" action="" enctype="multipart\form-data" >
<input type=hidden name=save_hide value=save_hide>
<TABLE width="100%" align=center border=0><TR>
<TD>Ekspонат nomini kiriting <\TD>
<TD><INPUT class="input" size="35" name="title" value=""><\TD>
<\TR><\TABLE>
<input type="submit" value="Saqlash" class="button" G`>
<\form><\TD><\TR><\TABLE>
<? } ?>
```

### ***Ma'lumotlar bazasiga ma'lumotlar yozish***

Ma'lumki, ma'lumotlarni jadvalga yozish uchun SQL tilidagi INSERT buyrug'i ishlatiladi:

```
mysql> INSERT INTO Artifacts SET title='Ekspонат nomi';
```

PHP skriptda bunday buyruqdan foydalanish uchun mysql\_query() funksiyasi mavjud.

mysql\_query sintaksisi

mysql\_query resursi (query qatori)

mysql\_query() SQL-so'rovni MySQL ma'lumotlar bazasining ma'lumotlar bazasiga jo'natadi. Agar ochiq aloqa bo'lmasa, funksiya parametrsiz mysql\_connect() funksiyasiga o'xshash holda MBBT bilan bog'lanishga urinadi.

So'rov natijasi buferlanadi.

## **5. Fayllar bilan ishlash**

PHP tilida fayllar bilan ishlash jarayonida quyidagi holatlarga duch kelish mumkin:

### ***Fayllarni ochish***

**Fayl** biror ma'lumot saqlash fizik qurilmasidagi baytlar ketma ketligidir. Har bir fayl joylashuvini ko'rsatuvchi absolyut yo'lga ega. Yo'l ajratuvchisi sifatida Windows da to'g'ri slesh (/) yoki teskari slesh (\) ishlatilishi mumkin. Boshqa operatsion tizimlarda faqat to'g'ri slesh ishlatiladi.

Fayllarni serever fayl tizimida ochish **fopen** funksiyasi yordamida amalga oshiriladi:

**int fopen(string filename, string mode [, int use\_include\_path])**

Birinci argument **filename** – fayl nomi yoki unga olib boruvchi absolyut yo'l. Agar absolyut yo'l ko'rsatilmasa, fayl kattalogda joylashgan bo'lishi kerak.

Ikkinchi argument **mode** fayl qaysi amallar uchun ochilganligini ko'rsatadi va quyidagi qiymatlarga ega bo'lishi mumkin:

**r** (faylni faqat o'qish uchun ochish; ochilgandan so'ng fayl ko'rsatkichi fayl boshiga o'rnatiladi);

**r+** (faylni o'qish va yozish uchun ochish; ochilgandan so'ng fayl ko'rsatkichi fayl boshiga o'rnatiladi);

**w** (Yozish uchun Yangi bo'sh fayl yaratish; agar shu nomli fayl mavjud bo'lsa, undagi hamma ma'lumot o'chiriladi);

**w+** (Yozuvlarni o'qish uchun Yangi bo'sh fayl yaratish; agar shu nomli fayl mavjud bo'lsa, undagi hamma ma'lumot o'chiriladi);

**a** (Yozuv qo'shish uchun faylni ochish, ma'lumotlar fayl oxiriga yoziladi);

**a+** (Yozuv qo'shish va o'qish uchun faylni ochish, ma'lumotlar fayl oxiriga yoziladi);

**b** (Ikkilik fayl Bilan ishlash usulini( o'qish va yozish) ko'rsatuvchi bayroq; faqat Windows da ko'rsatiladi).

Uchinchi shart bo'lmagan argument **use\_include\_path** fayllar **include\_path** kattalogida izlash kerakligini belgilaydi. (**include\_path** parametr php.ini faylda o'rnatiladi).

Fayl muvaffaqiyatli ochilganda, **fopen** funksiyasi fayl descriptorini qaytaradi, aks xolda – **false** qaytaradi. **Fayl Descriptori** ochilgan faylga ko'rsatkich bo'lib, operatsion tizim tomonidan shu fayl bilan amallarni qo'llash uchun ishlatiladi. Funksiya tomonidan qaytarilgan fayl descriptorini keyinchalik shu fayl bilan ishlaydigan hamma funksiyalarda ko'rsatish lozim.

Quyida keltirilgan kod, C:/WWW/HTML/file.txt faylini o'qish uchun ochadi:

```
<?
  $file = fopen("c:/www/html/file.txt","r");
  if(!file)
  {
    echo("Fayl ochish xatoligi");
  }
?>
```

Ikkiлик fayl, masalan rasmini ochish shu tariqa **b** bayrog'i bilan bajariladi:

```
<?
  $file = fopen("c:/www/html/river.jpg","rb");
  if(!file)
  {
    echo("Fayl ochish xatoligi");
  }
?>
```

### *Fayllarni akslantirish*

Ochilgan fayldagi ma'lumotlarni brauzerda **fpasssthu** funksiyasi yordamida akslantirish mumkin:

**int fpasssthu (int file)**

Argument **file** fayl descriptori emasdir.

```
<?
  $file = fopen("c:/www/html/pavlovo.jpg","rb");
  if(!file)
  {
    echo("Fayl ochish xatoligi");
  }
  else
  {
    fpasssthu($file);
  }
?>
```

Matnli fayllarni akslantirish uchun Yana bir **readfile** akslantirish funksiyasi mavjuddir:

**readfile (string filename)**

Shuni ta'kidlash lozimki argument sifatida bu funksiya fayl nomini emas, uning descriptorini qabul qiladi:

```
<?
  readfile ("file.txt");
?>
```

### *Fayllarni berkitish*

Fayl bilan ishini tugallagandan so'ng uni yopish kerak, Fayllarni yopish **fclose** funksiyasi yordamida amalga oshiriladi:

```
int fclose (int file)
```

Argument **file** yopish kerak bo'lgan fayl descriptori.

### *Fayllardan o'qish*

Ochiq fayldan qatorni **fread** funksiyasi yordamida o'qish mumkin:

```
string fread ( int file, int length )
```

Bu funksiya fayl simvollaridan iborat **length** uzunlikdagi **file** descriptorli qatorni qaytaradi.

```
<?
  $file = fopen("c:/www/html/file.txt","r");
  if(!file)
  {
    echo("Fayl ochish xatoligi");
  }
  else
  {
    $buff = fread ($file,100);
    print $buff;
  }
?>
```

Fayldan o'qish uchun **fgets** funksiyasidan foydalanish ham mumkin:

```
string fgets ( int file, int length)
```

Bu funksiya **length - 1** bayt uzunlikdagi satrni o'qiydi va qaytaradi. Yangi satr yoki fayl oxiriga etilganda o'qish to'xtatiladi. Fayl oxiriga etilganda funksiya bo'sh satr qaytaradi.

HTML teglarini tashlab yuborgan xolda faylni o'qish uchun **fgetss** funksiyasi qo'llanadi:

```
string fgetss (int file, int length [, string allowable_tags])
```

Shart bo'lmagan uchinchi parametr **allowable\_tags** tashlab yuborilmasligi kerak bo'lgan teglar ro'yxatidan iborat satrni o'z ichiga oladi. Bu satrda teglar vergul bilan ajratiladi.

Agar fayldagi bor ma'lumotlarni massivga yozish kerak bo'lsa, **file** funksiya si qo'llanadi:

```
array file (string filename [, int use_include_path])
```

Funksiya **filename** nomli faylni o'qiydi va har bir elementi o'qilgan fayldagi satrga mos keluvchi massiv qaytaradi. Quyidagi misolda funksiya yordamida fayl o'qiladi va undagi bor ma'lumot brauzerga chiqariladi.

```
<?
  $file_array = file("file.txt");
  if(!$file_array)
  {
    echo("Fayl ochish xatoligi");
  }
  else
  {
    for($i=0; $i < count($file_array); $i++)
    {
      printf("%s<br>", $file_array[$i]);
    }
  }
?>
```

Bu funksiya qulayligi shundaki uning yordamida fayldagi satrlar sonini hisoblash mumkin:

```
<?
$file_array = file ("file.txt");
if(!$file_array)
{
    echo("Fayl ochish xatoligi");
}
else
{
    $num_str = count($file_array);
    echo($num_str);
}
?>
```

Shuni e'tiborga olish kerakki **file** funksiyasini faqat kichkina fayllarni o'qish uchun qo'llash kerak. Kengaytmasi **\*.csv** bo'lgan fayllarni o'qish uchun **fgetcsv** funksiyasi qo'llanadi:

**array fgetcsv ( int file, int length, char delim)**

Funksiya fayldan satr o'qiydi va **delim** simvoli bo'yicha uni ajratadi. Parametr **delim** albatta bir simvoldan iborat satr bo'lishi kerak, aks xolda satrlarning faqat birinchi simvoli inobatga olinadi. Funksiya xosil bo'lgan massivni yoki fayl oxiriga etilgan bo'lsa false qiymatni qaytaradi. Bo'sh satrlar tashlab yuborilmaydi, aksincha bir element bo'sh satrdan iborat massiv qaytaradi. Parametr **length** satrlar maksimal uzunligini qaytaradi, **fgets** funksiyasidagi kabi.

CSV formati MSExcel fayllari saqlanuvchi formatlardan biridir. Quyidagi misolda MSExcel da yaratilgan, foydalanuvchilar parollarini o'z ichiga olgan file.csv o'qiladi.

```
<?
$count = 1;
$file = fopen ("file.csv","r");
while ($data = fgetcsv ($file, 1000, ","))
{
    $num = count ($data);
    $count++;
    for ($i=0; $i < $num; $i++)
    {
        print "$data[$i]<br>";
    }
}
fclose ( $file );
?>
```

### *Fayllarga yozish*

Fayllarga yozish **fputs** va **fwrite** bir xil funksiyalari bilan amalga oshiriladi:

**int fputs ( int file, string string [, int length ])**  
**int fwrite ( int file, string string [, int length ])**

Birinchi argument yozuv amalga oshiralayotgan fayl descriptori. Ikkinchi argument faylga yozilishi kerak bo'lgan satr. Uchinchi shart bo'lmagan argument satrda yozilishi kerak bo'lgan simvollar soni. Uchinchi argument ko'rsatilmasa hamma satr yozilishi kerak.

Bu misolda "file.txt" fayliga "Hello, world!" qator yoziladi.

```
<?
$file = fopen ("file.txt","r+");
$str = "Hello, world!";
if ( !$file )
```

```

{
    echo("Fayl ochish xatoligi");
}
else
{
    fputs ( $file, $str);
}
fclose ($file);
?>

```

### *Fayllardan nusxa olish, nomini o'zgartirish va fayllarni o'chirish*

Fayldan nusxa olish **copy** funksiyasi yordamida amalga oshiriladi:

```
int copy ( string file1, string file2)
```

**copy** funksiyasi **file1** nomli fayldan **file2** nomli fayl nusxa oladi. Agar fayl **file2** mavjud bo'lsa, u qaytadan yoziladi.

Fayl nomini o'zgartirish **rename** funksiyasi yordamida amalga oshiriladi:

```
int rename ( string old, string new)
```

Bu funksiya fayl **old** nomini **new** nomiga almashtiradi.

Agar fayl Yangi nomi boshqa fayl tizimida joylashgan bo'lsa **rename** funksiyasi faylga Yangi nom bermaydi.

Faylni o'chirish **unlink** funksiyasi yordamida amalga oshiriladi:

```
int unlink ( string filename)
```

### *Fayllar atributlari*

Fayl atributlari haqida qo'shimcha ma'lumot olish uchun quyidagi funksiyalardan foydalanishingiz mumkin.

**file\_exists** funksiyasi fayl mavjudligini tekshiradi va fayl mavjud bo'lsa true, aks xolda false qiymat qaytaradi:

```
bool file_exists ( string filename)
```

**filetime** funksiyasi faylga oxirgi murojaat vaqtini qaytaradi:

```
int filetime ( string filename)
```

**filemtime** funksiyasi faylning oxirgi o'zgartirish vaqtini qaytaradi:

```
int filemtime ( string filename)
```

**file\_size** funksiyasi fayl hajmini baytlarda qaytaradi:

```
int file_size ( string filename)
```

**file\_type** funksiyasi fayl tipini qaytaradi:

```
string file_type ( string filename)
```

Bu funksiya qaytaradigan satr quyidagi fayl tiplaridan biriga tegishli bo'ladi:

- char (maxsus simvolli qurilma);
- dir (kattalog);
- fifo (nomlangan kanal);
- link (simvolli ilova);
- block (maxsus blokli qurilma);
- file (oddiy fayl);
- unknown (tip ma'lum emas).

Fayl karakteristikalarini qaytaruvchi funksiyalardan foydalanish ko'p resurslarni talab qilgani uchun, bunday funksiyalarni chaqirishda unumdorlikni yo'qotmaslik uchun, PHP fayl haqidagi ma'lumotni keshlaydi. Bu keshni **clearstatcache** funksiyasi yordamida tozalash mumkin:

```

<?
clearstatcache();

```



?>

### *Fayllar bo'yicha ko'chish*

Fayldan ma'lumot o'qilganda joriy pozitsiya ko'rsatkichi navbatdagi o'qilmagan simvolga suriladi. Ko'rsatkich xolatini boshqarishga imkon beruvchi bir necha funksiyalar mavjuddir.

Joriy pozitsiya ko'rsatkichini fayl boshiga keltirish uchun **rewind** funksiyasi qo'llanadi:

**int rewind ( int file)**

**file** argumenti fayl descriptoridir.

Ko'rsatkich joriy pozitsiyasini **ftell** funksiyasi yordamida aniqlash mumkin:

**int ftell ( int file)**

Ko'rsatkichni fayl ixtiyoriy joyiga **fseek** funksiyasi yordamida o'rnatish mumkin:

**int fseek (int file, int offset [, int whence ])**

Funksiya **fseek** fayl ko'rsatkichini **offset** siljishli baytga o'rnatadi (fayl boshidan, oxiridan yoki joriy pozitsiyadan **whence** **paorometri qiymatiga qarab**). Argument **file** fayl descriptoridir. Argument **whence** qaysi joydan **offset** siljish hisoblanishi kerakligini aniqlaydi va quyidagi qiymatlardan biriga teng bo'lishi mumkin:

**SEEK\_SET** (pozitsiyani fayl boshidan hisoblaydi);

**SEEK\_CUR** (pozitsiyani ko'rsatkich joriy pozitsiyasidan *otschityvaet pozitsiyu otnositelno tekuyego polojeniya ukazatelya*);

**SEEK\_END** (*otschityvaet pozitsiyu otnositelno kontsa fayla*).

Ko'zda tutilgan bo'yicha argument **whence** qiymati **SEEK\_SET**.

Ko'rsatkich fayl oxirida ekanligini **feof** funksiyasi yordamida aniqlash mumkin:

**int feof ( int file)**

Agar ko'rsatkich fayl oxirida joylashgan bo'lsa, funksiya **true** qaytaradi aks xolda **false**.

**feof** funksiyasini fayl o'qishda ishlatish qulaydir:

<?

```
$file = fopen ("file.txt","r");
if ($file)
{
    while(!feof($file))
    {
        $str = fgets($file);
        echo $str;
        echo "<br>";
    }
    fclose ( $file);
}
else
{
    echo("Fayl ochish xatoligi");
}
```

?>

Bu funksiya yordamida faylda satrlar sonini aniqlash qulay:

<?

```
$file = fopen ("file.txt","r");
if ($file)
{
    $counter = 0;
    while(!feof($file))
    {
        $str = fgets ($file);
        $counter++;
    }
}
```

```

        echo($counter);
        fclose ($file);
    }
    else
    {
        echo("Fayl ochish xatoligi");
    }
?>

```

### ***Kattaloglar bilan ishlash***

Joriy kattalogni o'rnatish uchun **chdir** funksiyasi qo'llanadi:

**int chdir ( string directory)**

Bu funksiya bilan quyidagicha ishlash mumkin:

**chdir("/tmp/data");** // absolyut yo'l bo'yicha o'tish

**chdir("./js");** // joriy kattalog ost kattalogiga o'tish

**chdir("../");** // ajdod kattaloga o'tish

**chdir("~/data");** // o'tamiz /home/foydalanuvchi/data (Unix uchun)

Joriy kattalogni aniqlash uchun **getcwd** funksiyasidan foydalanish mumkin:

**string getcwd ( string path)**

Kattalogni ochish uchun path parametri bilan berilgan kattalogni ochuvchi **opendir** funksiyasidan foydalaniladi:

**int opendir ( string path)**

Kattalog ochilgandan so'ng, uni **readdir** funksiyasi bilan o'qish mumkin:

**string readdir ( int dir)**

Bu funksiya kattalogdagi elementlar nomlarini qaytaradi. Undan tashqari kattaloglarda "." va ".." elementlari mavjud. Birinchi element joriy ikkinchisi ajdod kattaloga ko'rsatadi. Joriy kattalogni nomini "." sifatida ko'rsatib ochish mumkin:

**\$dir = opendir (".");**

Kattalog bilan ish tugagandan so'ng uni yopish kerak. Kattalog yopilishi **closedir** funksiyasi bilan amalga oshiriladi:

**void closedir (\$dir)**

Quyida joriy kattalogdagi fayllarni o'qish va chiqarishga misol keltirilgan.

```

<?
    $dir = opendir (".");
    echo "Files:\n";
    while ($file = readdir ($dir))
    {
        echo "$file<br>";
    }
    closedir ($dir);
?>

```

Bu funksiya "." va ".." qiymat ham qaytaradi. Agar bu kerak bo'lmasa, bu qiymatlarni quyidagicha tashlab yuborish mumkin:

```

<?
    $dir = opendir (".");
    while ( $file = readdir ($dir))
    {
        if (( $file != ".") && ($file != ".."))
        {
            echo "$file<br>";
        }
    }
    closedir ($dir);
?>

```

Ko'rilgan funksiyalarga misol sifatida, c:/temp kattalogidagi bir sutka davomida murojaat qilinmagan hamma fayllarni o'chiruvchi script yaratamiz. Fayllarni o'chirish funksiyasi bu holda rekursiv chaqiriladi.

```
<?
function delTemporaryFiles ($directory)
{
$dir = opendir ($directory);
while (( $file = readdir ($dir)))
{
    if( is_file ($directory."/".$file))
    {
        $acc_time = filemtime ($directory."/".$file);
        $time = time();
        if (($time - $acc_time) > 24*60*60)
        {
            if ( unlink ($directory."/".$file))
            {
                echo ("Fayllar muvaffaqiyatli o'chirilgan");
            }
        }
    }
    else if ( is_dir ($directory."/".$file) && ($file != ".") && ($file !=
".."))
    {
        delTemporaryFiles ($directory."/".$file);
    }
}
closedir ($dir);
}
delTemporaryFiles ("c:/temp");
?>
```

Kattaloglarni yaratish **mkdir** funksiyasi yordamida amalga oshiriladi:

**bool mkdir ( string dirname, int mode)**

Bu funksiya dirname nomli i mode.murojaat huquqlari bilan kattalog yaratadi. Kattalog yaratilmasa false qaytaradi. Murojaat xquqlari faqat UNIX kattaloglari uchun beriladi, chunki Windows da bu argument ta'sir ko'rsatmaydi. Quyida c:/temp direktoriyasida test v kattalogini yaratish misoli ko'rilgan.

```
<?
$flag = mkdir ("c:/temp/test", 0700);
if($flag)
{
    echo("Kattalog muvaffaqiyatli yaratilgan");
}
else
{
    echo("Kattalog yaratish xatosi ");
}
?>
```

Kattalogni **rmdir** funksiyasi yordamida o'chirish mumkin:

**bool rmdir ( string dirname)**

Endi yaratilgan /test kattalogini o'chiramiz:

```
<?
```

```

$flag = rmdir ("c:/temp/test");
if($flag)
{
    echo("Kattalog muvaffaqiyatli o'chirilgan");
}
else
{
    echo("Kattalogni o'chirish xatosi ");
}
?>

```

Funksiya **rmdir** faqat bo'sh kattaloglarni o'chiradi. Bo'sh bo'lmagan kattaloglarni o'chirish uchun, funksiya yaratib c:/temp kattalogini undagi papkalar va fayllari bilan birga o'chiramiz:

```

<?
function full_del_dir ($directory)
{
    $dir = opendir($directory);
    while(($file = readdir($dir)))
    {
        if ( is_file ($directory."/".$file))
        {
            unlink ($directory."/".$file);
        }
        else if ( is_dir ($directory."/".$file) &&
            ($file != ".") && ($file != ".."))
        {
            full_del_dir ($directory."/".$file);
        }
    }
    closedir ($dir);
    rmdir ($directory);
    echo("Kattalog muvaffaqiyatli o'chirilgan");
}
full_del_dir ("c:/temp")
?>

```

Funksiyani rekursiv chaqirganda argumentlar sifatida joriy va ajdod kattaloglarga ko'rsatuvchi "." va ".." yozuvlarni uzatmang, chunki bu holda siz ma'lumotlaringizni yo'qotishingiz mumkin. Bu yozuvlarni shartli operator yordamida o'tkazib yuboring.

### ***PUT va POST usullari***

HTTP PUT va HTTP POST usullari fayllarni serverga yuklash uchun mo'ljallangan.

Web-serverdagi ma'lumot bilan ishlash uchun HTTP Protokolda uchta usul mavjud: GET, PUT va POST. GET usuli Web-sahifalarni olish uchun qo'llanadi, bunda forma hamma o'zgaruvchilari URL ga uzatiladi. Ko'pgina Web-serverlarda URL maksimal uzunligi cheklanganligi uchun (odatda 1024 dan oshmaydi), katta xajmdagi ma'lumotlarni uzatishda GET usulini qo'llab bo'lmaydi.

PUT usuli serverdagi ma'lumotni Yangilash uchun qo'llanadi va HTTP PUT so'rov natijasi serverda saqlanishini talab qiladi. So'rov quyidagi ko'rinishga ega:

PUT /path/filename.html HTTP/1.1

Bu holda Web-server bu so'rov natijasini Web-serverning URL nomlar fazosida /path/filename.html ko'rinishda saqlashi lozim. Ko'zda tutilgan bo'yicha Web-server o'zi bunday so'rovlarni bajarmaydi, balkim ularni qayta ishlash uchun CGI-stsenariy beradi. Apache da PUT so'rovlarni qayta ishlovchi stsenariy tayinlash uchun httpd.conf fayldagi Script direktivasini o'zgartirish mumkin, masalan quyidagicha:

Script PUT /cgi-bin/put.cgi

Bu shuni bildiradiki PUT-so'rovlarni CGI-script put.cgi qayta ishlaydi.

Odatda fayllarni serverga yuklash uchun HTTP POST usulidan foydalaniladi. Bu usul formadan ma'lumotlar katta xajmini uzatishga imkon beradi va so'rov tanasidagi hamma forma o'zgaruvchilarni saqlaydi.

### Nazorat savollari:

1. Ma'lumotlar bazasi deganda nimani tushunasiz?
2. Qanaqa ma'lumotlar bazasi dasturlarini bilasiz va ularni imkoniyatlari haqida gapiring?
3. PHP da MySQL bilan bog'lanish funksiyasini misollar yordamida tushuntirib bering?
4. PHP da MySQL so'rovlarni amalga oshirishga misol keltiring?
5. Turli ma'lumotlar bazasi dasturlarida ishlash jarayonida har biriga tegishli bo'lgan kamchilik va afzalliklarni ayting?
6. Fayllar bilan ishlash jarayonlarini tushuntiring?
7. Fayllarni serverga yuklash uchun mo'ljallangan qanday usullar mavjud?
8. Web-serverdagi ma'lumot bilan ishlash uchun qanday usullar mavjud va ularning ishlash mohiyatini tushuntiring?
9. PUT va POST usullari usullarining mohiyatini tushuntiring?
10. PHP dasturlash tilida fayllarni yozish, ochish, berkitish, nusxa ko'chirish qanday amalga oshiriladi?

## 14-ma'ruza.Cookie, seans, FTP va e-mail texnologiyalari.

### Reja:

1. Cookie texnologiyasini o'rnatish;
2. Cookie texnologiyasini o'qish;
3. FTP texnologiyasi;
4. E-mail texnologiyasi.

**Kalit so'zlar:** Cookie, seans, FTP, E-mail texnologiyalari, FTP server, FTP klient, anonym FTP, FTP Proxy, FTP host, FTP sayt.

**Ishdan maqsad:** Ishlash jarayonida Cookie, seans, FTP, E-mail texnologiyalari bilan ishlash ko'nikmalariga ega bo'lish va talabalarga bu haqida bilim berish.

### 1. Cookie texnologiyasini o'rnatish

Cookie — klient kompyuterida saqlanuvchi va u har safar serverga murojaat qilayotganda web-serverga yuboriladigan matn satridir. Shu tariqa ma'lumotlar turli skriptlar aro formalar yoki URL adreslarsiz uzatilishi mumkin. Cookie qiymatini o'rnatish uchun quyidagi sintaksisdan iborat bo'lgan setcookie funksiyasidan foydalaniladi:

```
bool setcookie (string name [, string value [, int expire [, string  
path [, string domain [, bool secure]]]])
```

Funksiya klient kompyuterida saqlanuvchi cookie larni tasnif etadi. Quyida ushbu funksiya parametrlarining tasnifi keltirilgan:

- name. Cookie ning nomi.
- value. Cookie ning qiymati.

- expire. Cookie ning amal qilish muddati. Agar uberilsa, u tugagach cookie o'chirib tashlanadi. Agar ko'rsatilmasa, cookie brauzer oynasi yopilgandan so'ng o'chirib tashlanadi
- path. Serverdagi cookie ga ruxsat etilgan adres.
- domain. cookie ga ruxsat etilgan domen.
- secure. HTTPS protokol orhali bog'lanishda cookie ning havfsizlik belgisi. Standart holatda cookie HTTPS uchun ham HTTP dagi kabi ishlaydi.

Ko'rinib turibdiki, cookie HTTP-so'rovning qismi hisoblanadi va u brauzerga yuboriladi. Shu sabab HTML-kod formalashtirishdan oldin uning qiymati o'rnatilishi kerak. Bu shuni anglatadiki, setcookie funksiyasining qiymati HTML-tegdan oldin va echo operatorigacha o'rnatiladi.

Agar bu qoida saqlanmasa setcookie funksiyasini chaqirish FALSE qiymat qaytaradi va bu cookie formirovka qilinishida xatolik bo'lganligini anglatadi. cookie ning to'g'ri formirovka qilinishi funksiyaning TRUE qiymat qaytarishiga olib keladi. Bu cookie klientga qabul qilindi degani emas albatta. Chunki brauzerni sozlashda cookie o'chirib tashlash yoki serverga jo'natmaslik parametrlari o'rnatilgan bo'lishi mumkin.

## 2. Cookie texnologiyasini o'qish

Cookie qiymati o'rnatilganidan so'ng, sahifa qayta yuklanmagunicha darhol undagi skript uchun cookie ga ruxsat mavjud bo'lmaydi. Chunki cookie foydalanuvchi kompyuterida saqlanadi va brauzer orhali web-serverga jo'natiladi. Bundan tashqari, cookie qachonki uning domeni server domeni bilan mos tushgandagina jo'natiladi.

Cookie ga ruxsatni olish uchun maxsus \$\_COOKIE superglobal massividan foydalaniladi. Massivning qiymati sifatida oldin ishlatilgan cookie nomi olinadi. Massiv odatda skript yuklanayotgan vaqtda \$\_GET, \$\_POST va \$\_REQUEST massivlari bilan avtomatik tarzda to'ldiriladi

Cookie dan foydalanishdan oldin uning qiymati o'rnatilganligiga ishonch hosil qiling. Buning uchun isset funksiyasidan foydalanish juda qulay. Quyidagi misolda message nomli cookie ning qiymati tekshirilishi va ko'rsatilishi keltirilgan.

### Misol:

```
<HTML>
<HEAD> <TITLE> Cookie ning qiymatini o'qish <\TTITLE> <\HEAD>
<BODY>
<CENTER>
<N1> Cookie ning qiymatini o'qish <\H1>
Cookie ning qiymati:
<?php
if (isset ($_COOKIE ['message']))
{
echo Cookie ning qiymati:' . $_COOKIE ['message' ] ;
}
else
{
echo 'Cookie o'rnatilmagan' ;
}
?>
<\CENTER; .
<BODY>
<\HTML>
```

Cookie massivlarda ham tashkil etilgan bo'lishi mumkin. Masalan quyida uchta cookie o'rnatilgan:

```
setcookie ("cookie[one]" , "Bugun");
setcookie ("cookie[two]" , "Hayot");
```

```
setcookie ("cookie[three] ", "go'zal!") ;
```

Natijada \$\_COOKIE['cookie'] massiv qiymatlari quyidagi tarzda bosmaga chiqarilishi mumkin:

```
if (isset ($_COOKIE ['cookie'])) {  
    foreach ($_COOKIE ['cookie'] as $data)  
    {  
        echo "$data <BR>";  
    }  
    ...  
}
```

### 3. FTP texnologiyasi

Quyida FTP ning ba'zi atamalarini ko'rib chiqamiz:

#### 1. FTP

File Transfer Protocol ( fayllarni almashish protokoli); tarmoq orqali (xususan Internet) fayllarni uzatish protokoli. Buni bir kompyuterdan boshkasiga fayl nusxasini ko'chirish kabi tushunish mumkin.

#### 2. FTP Server

Bu faylni junatilishi so'rovini kutuvchi kompyuter yoki server.

#### 3. FTP Client

Bu FTP serverga so'rov jo'natuvchi kompyuter. So'rov tekshiruvi yoki tasdiqlanishidan so'ng FTP klient kompyuteri ma'lumotlarni serverga yuklashi yoki serverdan yuklab olishi mumkin.

#### 4. Anonim FTP

Ushbu serverga FTP klient kompyuteri orhali avtorizatsiyadan o'tmasdan bog'lanishi mumkin. Bunday imkoniyatni bir nechta web saytlarda ko'rishimiz mumkin, qaysiki ro'yxatdan o'tmasdan ma'lum fayllarni yuklab olish imkoniyatini beradi.

#### 5. FTP Host

Ushbu xizmat FTP sayt sifatida, ya'ni saytda ro'yxatdan o'tib, saytda keltirilgan ma'lumotlar fayllarni yuklab olish imkoniyatini beruvchi kompyuter. Bu foydalanuvchilar soni cheklangan pullik xizmat hisoblanadi.

#### 6. FTP sayt

FTP xost kompyuteri tasarrufidagi foydalanuvchi login paroli talab qilinuvchi web sahifa. Xost serverda bir nechta web-sahifa joylanishi mumkin, bunda har bir sayt uchun alohida foydalanuvchi avtorizatsiyasi mavjud.

#### 7. FTP Proxy

FTP Proxy bu shunaqa server kompyuterki, bunda so'rov FTP serverga jo'natiladi, ushbu so'rov proksi orhali o'tib, keyin zarur manzilga yo'naltiriladi.

### 4. E-mail texnologiyasi

Bu xizmat internet tizimi orhali o'zaro pochta xabarlar almashinuvi usulidir. Bunday tizim oraliq saqlanish metodi asosida yo'lga qo'yilgan. Zamonaviy internet protokollar xabarlar jo'natish imkoniyati bo'yicha Yuqori darajaga ko'tarildi. Pochta jo'natishning eng keng tarqalgan protokoli SMTP (Simple Mail Transport Protocol) hisoblanadi.

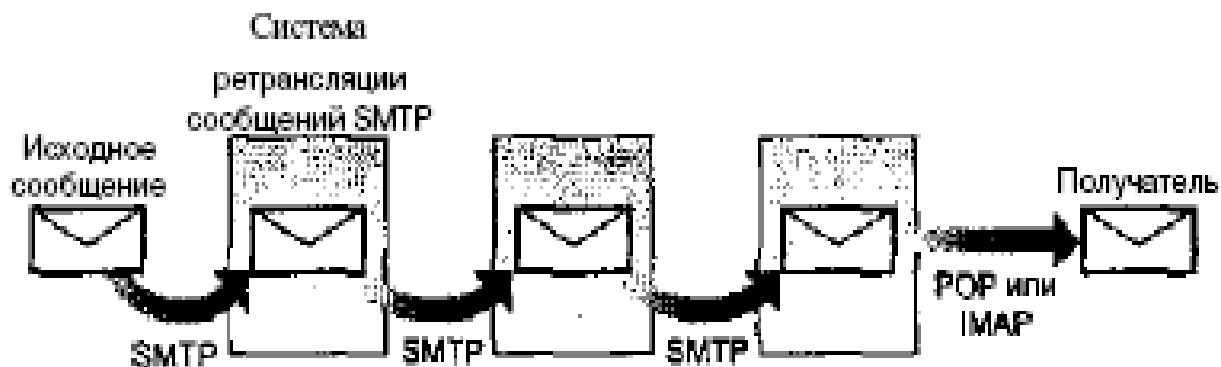


Рис. 22.2 Для пересылки и получения почты используются различные протоколы

Pochtani qabul qilish pochta protokoli POP (Post Office Protocol), yoki Internet xabarlarini qabul qiluvchi protokol IMAP (Internet Message Access Protocol) dan foydalaniladi.

#### Nazorat savollari:

1. Cookie texnologiyasi nima?
2. Seans texnologiyasi nima va uning mohiyatini tushuntiring?
3. FTP protokol va uning vazifasi nimadan iborat?
4. E-mail texnologiyasi va uning vazifasi haqida gapiring?
5. FTP host xizmati ishlash mohiyatini tushuntiring?
6. FTP texnologiyasida mavjud bo'lgan yana qanday xizmatlarni bilasiz?