

Boymurodov A.X., Allamova Sh.Sh.

WEB TEKNOLOGIYALARI

**O'ZBEKISTON RESPUBLIKASI OLIY TA'LIM, FAN VA
INNOVATSIYALAR VAZIRLIGI**

CHIRCHIQ DAVLAT PEDOGOGIKA UNIVERSITETI

Sh.Sh.Allamova

WEB TEKNOLOGIYALARI

Darslik

Chirchiq-2024

Sh.Sh. Allamova
Web texnologiyalari

Darslik–T., 2024 y. 213 b.

Darslik web texnologiyalari fanining asosiy tushunchalari, html5 xususiyatlari, htmlda matnlar, ro‘yxatlar va jadvallar bilan ishlovchi teglar va atributlar, multimedia ob’ektlari va gipermatnli hujjatlar, shakllar, ularning vazifalari va turlari, html elementlariga ishlov berish uchun CSSga ulanish usullari, selektor turlari, css da matnlar, rasm va xavolalarga ishlov berish, cssda ro‘yxatlar, jadvallar va shakllarga ishlov berish, JavaScript asoslari va web saytlarni yaratishda cms texnologiyasidan foydalanish ko‘nikmalarini hosil qilishga qaratilgan.

Учебное пособие охватывает основные понятия веб-технологий, возможности html5, теги и атрибуты работы с текстами, списки и таблицы в html, мультимедийные объекты и гипертекстовые документы, формы, их функции и типы, методы подключения к CSS для обработки html-элементов, селектор типы, направлен на формирование навыков обработки текстов, изображений и ссылок в css, обработки списков, таблиц и форм в css, работы с основами JavaScript и использования технологии cms при создании веб-сайтов.

The training manual includes the basic concepts of web technologies, html5 features, tags and attributes for working with texts, lists and tables in html, multimedia objects and hypertext documents, forms, their functions and types, methods for connecting to CSS for processing html elements , a type selector, is aimed at developing skills in processing texts, images and links in css, processing lists, tables and forms in css, working with the basics of JavaScript and using cms technology when creating websites.

Taqrizchilar: p.f.f.d(phd), CHDPU, **Xurramov A. J**
t.f.d., TATU, **Beknazarova S.S.**

O‘zbekiston respublikasi oliy ta’lim, fan va innovatsiyalar vazirligi Chirchiq davlat pedagogika universiteti kengashining 20__ yil ____-sonli qaroriga asosan 60110600 – Matematika va informatika ta’lim yo‘nalishlari bo‘yicha tahsil olayotgan talabalar va professor o‘qituvchilar uchun darslik sifatida nashr qilishga tavsiya etilgan.

KIRISH

Ayni vaqtida axborot texnologiyalari jadal suratlar bilan rivojlanmoqda va inson faoliyatining deyarli barcha sohalariga kirib bormoqda. Internet zamonaviy insonning kundalik hayotining ajralmas qismiga aylandi. Internet rivojlanib borayapti. Bu esa ko‘p odamlar o‘z hayat tarzini tubdan o‘zgartirib yuborishiga sabab bo‘lmoqda. Hozirgi kunda dunyoning qariyb 3/2 qismi bu tarmoqdan foylanmoqda va bu ko‘rsatkich kundan-kunga o‘sib borayapti. Internet tufayli hayotning barcha sohalarida deyarli cheksiz imkoniyatlar mavjud. Bugun axborotlarni elektron ko‘rinishda saqlashni va jamlashni tashkil qilish unchalik murakkab ish emas. Xatto Internet texnologiyalarini qo‘llab yer sharining ixtiyoriy nuqtasidagi elektron arxivdan foydalanishni tashkil qilish mumkin.

Axborot texnologiyalarida web texnologiyalar vositalari Web-sahifa va ma’lumotlar bazasi, Web-interfeyslarni dasturlash uchun PHP vositalarini amalga oshirish eng dolzarb masalalardan hisoblanadi.

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

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

“Web texnologiyalari” fanidan darslik bakalaviat ta’lim yo‘nalishlarining 60110600 - Matematika va informatika ta’lim yo‘nalishining 3-bosqich talabalari uchun mo‘ljallangan.

Mazkur darslikda fan dasturida keltirilgan barcha mavzularning uzviy ketma-ketlikdagi batafsil bayoni, o‘quv kursi bo‘yicha test savollari, qisqartmalar, atamalar va foydalilanigan adabiyotlar ro‘yxatini o‘z ichiga oladi.

Darslik 13 ta bobdan iborat bo‘lib, har bir bobda mavzular misollar bilan yoritib berilgan va har bir bob so‘ngida mavzuga doir nazorat savollari berilgan.

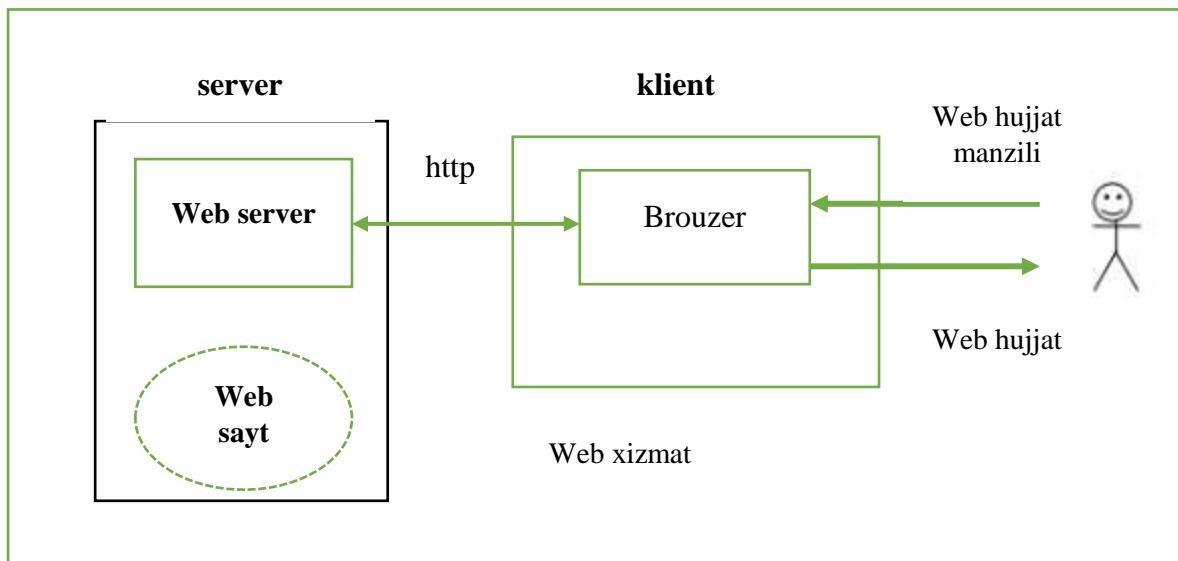
Mazkur darslikda talabalarga HTML, HTML5 gipermatn tili asoslari, CSS, CSS3 stillar majmuasi, web-sahifa va java Script dasturlash tillarini o‘rganishda yordamchi qo‘llanma bo‘lib xizmat qiladi.

Ushbu darslikni yaratishda ayrim xato va kamchiliklar bo‘lishi mumkin, bu kamchiliklarni bartaraf etishda o‘z takliflari bilan o‘rtoqlashgan talabalarga minnaddorchilik bildiramiz. Barcha fikr va takliflarni mammuniyat bilan qabul qilamiz.

**I-BOB. WEB-TEXNOLOGIYA HAQIDA TUSHUNCHA. BRAUZER
DASTURLAR. WEB-SAHIFANING TUZILISHI VA YARATISH
IMKONIYATLARI, ASOSIY TUSHUNCHALARI. HTML TILIGA
KIRISH, UNING ASOSIY ELEMENTLARI. HTML DA MATNLAR BILAN
ISHLOVCHI TEGLAR**

1.1. Web-sahifa, Web-sayt, Web-server tushunchalari

World Wide Web (WWW, Butun dunyo o‘rgimchak to‘ri) – bu mijoz-server texnologiyasi asosida tashkil etilgan, keng tarqalgan Internet xizmatidir. WWW xizmati gipermatn tushunchasi, HTTP protokoli va HTML tiliga asoslangan.



1.1-rasm. Web xizmat ko‘rsatish

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

Web-sahifa (inglizcha **Web page**) – web-brauzer yordamida kirish mumkin bo‘lgan World Wide Web hujjati yoki axborot resursi.

Web-sahifalar odatda HTML yoki XHTML belgilash tillarida yoziladi va boshqa sahifalarga tezda o‘tish uchun giperhavolalarni o‘z ichiga olishi mumkin.

Web-sahifadagi ma’lumotlar turli shakllarda taqdim etilishi mumkin:

- Matn;
- statik va animatsion grafiklar;
- audio;

- video;
- appletlar.

Web-sahifaning axborot tarkibi odatda kontent deb ataladi.

Umumiy mavzu, dizayn va havolalarga ega bo‘lgan va odatda bitta web-serverda joylashgan bir nechta web-sahifalar web-saytni tashkil qiladi.

Sayt (inglizcha website: web – “tarmoq” va site – “joy”, so‘zma-so‘z “tarmoqdagi joy”) – bitta manzil (domen nomi yoki IP manzil) ostida birlashtirilgan shaxsiy yoki tashkilotning kompyuter tarmog‘idagi elektron hujjatlari (fayllari) to‘plami. Odatda, sayt Internet tarmog‘ida joylashgan deb taxmin qilinadi.

Web-server – bu mijozlardan, odatda web-brauzerlardan HTTP so‘rovlarni qabul qiladigan va odatda HTML sahifasi, rasm, fayl, media oqimi yoki boshqa ma’lumotlar bilan birga HTTP javoblarini chiqaradigan server. Web-serverlar World Wide Web tarmog‘ining asosidir. Web-server – tarmoqqa ulangan kompyuter yoki undagi dastur hisoblanib, umumiy resurslarni mijozga taqdim etish yoki ularni boshqarish vazifalarini bajaradi. Web-serverlar ma’lumotlar bazalari va multimediyali ma’lumotlarni bir-biriga moslashtiradi. Web-serverda Web-sahifa va Web-saytlar saqlanadi. Web-server deganda web-server funksiyalarini bajaradigan dasturiy ta’midot ham, ushbu dastur ishlaydigan kompyutering o‘zi ham tushuniladi.

Web-saytlarni saqlash uchun apparat serverlari web-serverlar deb ataladi. Saqlash xizmatining o‘zi hosting deb ataladi.

Hosting (ingl. “web-hosting”) – Internet hosting xizmatining bir turi bo‘lib, jismoniy va yuridik shaxslarga o‘zlarining web-sahifalarini internetga uzlusiz ulangan web-serverlarga joylashtirish xizmatini nazarda tutadi. Xosting, shuningdek, mijoz uskunasini provayder hududida joylashtirish, uning yuqori o‘tkazuvchanlikdagi aloqa kanallariga ulanishini ta’minalash xizmati (kolokatsiya) deb ataladi. Saytni Internetga joylashtirish uchun birinchi navbatda unga joy kerak. Web-sahifalarni Internet provayderi serverida joylashtirish va qo‘llab-quvvatlash hosting deb ataladi. Joy beruvchi kompaniya esa hoster deyiladi.

Domen nomi – foydalanuvchi brauzerning manzil maydonida yozadigan belgilar jamlanmasi. Shuni takidlash kerakki, brauzerning manzil maydonidagi belgilar jamlanmasi URL deyiladi. URL (*Uniform Resource Locator – Resursning yagona ko‘rsatkichi*) va domen nomi o‘rtasida farq bor. Yani, URLda domen nomiga qo‘sishimcha biron sahifa ham yozilgan bo‘lishi mumkin. Domen – bu cheksiz internet ummonidagi serverlardan birida joylashgan qaysidir saytga olib boradigan manzil hisoblanadi. Aslida Siz hosting xizmatidan foydalanib, saytingizni qaysidir serverga joylashtirganingizda, saytingiz manzili (server nuqta’i nazaridan) qandaydir IP manzilga teng bo‘ladi. Masalan, 192.168.162.134. Bunday IP manzilni eslab qolish qiyin. Agar domen bo‘limganida saytingizga tashrif buyurish uchun foydalanuvchilar saytingiz IP manzilini eslab qolishlari va brauzerlariga 192.168.162.134 deb yozish orqali saytingizga kirishlari kerak bo‘lar edi. Domen afzalligi shundaki, u tushunarsiz va eslab qolish qiyin bo‘lgan IP manzillarni odamlarga tushunarli, eslab qolish oson bo‘lgan chiroyli sayt nomlari aylantiradi. Shuning uchun ham domen sotib olganingizda eng avvalo bu domenni hostingingiz IP manziliga ulab qo‘yasiz. Shunda domen nomi saytingiz IP manzilining tarjimasi bo‘lib qoladi. Har safar foydalanuvchilar o‘z brauzerlarida domen nomingizni yozishsa, bu domen hostingingiz serverining IP manzilini chaqiradi va foydalanuvchiga saytingiz ko‘rsatiladi.

IP manzil (o‘qilishi (ay-pi adres) ingilizcha **Internet Protocol**) – qurilmaning tarmoqdagi takrorlanmas virtual adresidir. Internet va lokal tarmog‘idagi qurilmalar bir-biri bilan IP protokoli orqali bir-birini IP adreslariga ma’lumot jo‘natish orqali aloqa qilishadi. Bu aloqaning o‘ziga yarasha qonun qoidalari bor va shu qonun qoida asosida bir IP manzil ikkinchi IP adresga xabar(paket) jo‘natadi va shu qonun qoida **protokol** deyiladi. Bu takrorlanmas ip manzil qurilmada (komputer/router(marshrut)/..) emas balki tarmoq interfeysida bo‘ladi.

Web-brauzer, brauzer – web-saytlarni ko‘rish, ya’ni web-sahifalarni so‘rash (asosan Internetdan), ularni qayta ishslash, ularni ko‘rsatish va bir sahifadan ikkinchisiga o‘tish uchun dasturiy ta’minot.

Ko‘pgina brauzerlar FTP serverlari tarkibini ham ko‘rsatishi mumkin.

Brauzerlar World Wide Web paydo bo‘lganidan beri doimiy ravishda rivojlanib bordi va o‘sib borishi bilan tobora ko‘proq talab qilinadigan dasturlarga aylandi. Bugungi kunda brauzer web-sahifaning turli komponentlarini qayta ishlash va ko‘rsatish, web-sayt va uning tashrif buyuruvchisi o‘rtasidagi interfeysni ta’minlash uchun murakkab dasturdir. Deyarli barcha mashhur brauzerlar bepul tarqatiladi yoki boshqa ilovalar bilan bir “to‘plamda” mavjud: Internet Explorer (Microsoft Windows bilan birgalikda), Mozilla Firefox (bepul dasturiy ta’milot), Safari (Mac OS bilan birga yoki Windows uchun bepul), Opera (8.50 versiyasidan boshlab bepul), Google Chrome (bepul dasturiy ta’milot), Avant (bepul dasturiy ta’milot).

1.2. Belgilash tillarining rivojlanish bosqichlari

Foydalanuvchi sahifaga so‘rov yuborganda, chaqirilgan sahifa avval serverda qayta ishlanadi va shundan keyingina tarmoq orqali tashrif buyuruvchiga fayl sifatida qaytariladi. Bu fayl kengaytmalari quyidagicha bo‘lishi mumkin: HTML, PHP, ASP, Perl, SSI, XML, DHTML, XHTML.

GML (Generalized Markup Language) – bu makroslar to‘plami bo‘lib, ularning asosiy maqsadi matnni bezash uchun teglardan foydalanadigan belgilashni amalga oshirishdir.

GML dan foydalanilganda hujjat matn xarakterini (paragraflar, sarlavhalar, ro‘yxatlar, jadvallar) belgilovchi teglar bilan belgilanadi. Bunday hujjat turli xil qurilmalar uchun avtomatik ravishda formatlanishi mumkin, faqat qurilma turini ko‘rsatish kerak. Masalan, siz hujjatni lazer yoki matritsali printer uchun formatlashingiz yoki uni faqat apparat profilini ko‘rsatish orqali ekranda ko‘rsatishingiz mumkin.

Bitta hujjatni boshqa hujjatga qo‘shish imkoniyatiga ega bo‘lish uchun loyiha mualliflari ma’lumotlarni tartibga solishning gipermatn usuliga e’tibor qaratishga qaror qilishdi.

GML shaklida ishlab chiqilgan talablarni amalga oshirish uchun asos allaqachon mavjud edi, lekin aniq texnologik yechim ustida to‘xtalib o‘tish qiyin edi.

1978-yilga kelib, Amerika Milliy Standartlar Institutining (ANSI) axborotni qayta ishslash qo‘mitasi gipermatnli ma’lumotlarni tayyorlash uchun tillarga jiddiy qiziqish uyg‘otdi. Charlz Goldfarb SGML (Standard Generalized Markup Language) deb nomlangan kuchli hujjat belgilash tilini yaratish bo‘yicha qo‘mitaga rahbarlik qildi. U GMLga asoslangan edi. SGML matnga belgilashni kiritish sintaksisini aniq belgilab berdi, shuningdek, qaysi teglarga va qayerda ruxsat berilganligini aniq tasvirlab berdi. Bu mualliflarga teglarni tanlash va ularga oddiy tilda nom berish orqali istalgan belgilashni yaratish va ishlatish imkonini berdi.

SGML yirik davlat va aerokosmik loyihalarda mashina tomonidan o‘qiladigan hujjatlarni almashish uchun ishlab chiqilgan. U matbaa va nashriyot sanoatida keng qo‘llanilgan.

1980-yilda spetsifikatsiyaning birinchi ishchi versiyasi paydo bo‘ldi.

1983-yilda SGML Mudofaa vazirligi va IRS tomonidan qo‘llab-quvvatlanadigan sanoat standarti sifatida qabul qilindi. Ikki yil o‘tgach, SGML foydalanuvchilarining xalqaro guruhi tuzildi. Shu tariqa gipermatnli texnologiyalarning gullagan davri boshlandi. 1986-yilda SGML ISO 8879 standarti sifatida qabul qilindi.

SGML (Standard Generalized Markup Language) – bu hujjat belgilash tillarini aniqlash uchun meta til. SGML turli ilovalar tomonidan foydalanish uchun ko‘plab sintaktik belgilash variantlarini taqdim etadi. GML tili katta va murakkab umumiyligi muammolarni hal qilish uchun yaratilgan. Til mualliflari uni juda moslashuvchan va keng qamrovli qilishga intilishgan, natijada til nihoyatda murakkab bo‘lib chiqdi.

HTML (HyperText Markup Language) – bu web-sahifalarni yaratish va nashr qilish uchun World Wide Web tomonidan qabul qilingan til.

- ✓ HTML 2.0, standart sifatida 1995-yil 22-sentyabrda tasdiqlangan;
- ✓ HTML 3.2 - 1997 yil 14-yanvar;

- ✓ HTML 4.0 - 1997 yil 18-dekabr;
- ✓ HTML 4.01 - 1999 yil 24-dekabr;
- ✓ HTML 5 - 2014 yil 28-oktyabr;
- ✓ HTML 5.1 2014-yil 19-dekabrdan ishlab chiqila boshlandi

Web-sahifalar sahifa mazmunining haqiqiy matnini va teglar yoki HTML belgilash teglari deb ataladigan formatlash buyruqlarini o‘z ichiga olgan matnli fayllardir. HTML kod brauzer (web-brauzer) tomonidan talqin qilinadi, u web-sahifaning formatlangan mazmunini ekranda aks ettiradi.

1995-yilgacha HTMLning ko‘plab norasmiy variantlari mavjud edi. Standart versiyani ulardan farq qilish uchun unga darhol ikkinchi raqam berildi. 3-versiya 1995-yil mart oyida World Wide Web Consortium (W3C) tomonidan taklif qilingan va jadvallar yaratish, murakkab matematik formulalarni ko‘rsatish kabi ko‘plab yangi funksiyalarni taqdim etgan. Ushbu standart ikkinchi versiyaga mos kelgan bo‘lsa ham, uni amalga oshirish o‘sha davr brauzerlari uchun qiyin edi. 3.1 versiyasi hech qachon rasman taklif qilinmagan va HTML standartining keyingi versiyasi 3.2 bo‘lib, u 3.0 versiyasining ko‘pgina yangiliklarini o‘tkazib yuborgan, lekin o‘sha paytda mashhur Netscape va Mosaic brauzerlari tomonidan qo‘llab-quvvatlanadigan nostandard elementlarni qo‘shgan.

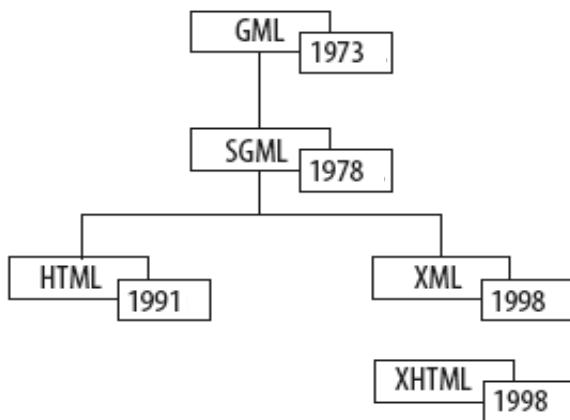
HTML 4.0 versiyasida alohida brauzerlar uchun xos bo‘lgan ko‘plab elementlar mavjud edi, biroq shu bilan birga standartni biroz “tozalash” amalga oshirildi. Ko‘pgina teglar, shrift xususiyatlarini o‘zgartirish uchun foydalaniladigan shrift kabi teglar, matn belgilarining tagiga chizishni ko‘rsatuvchi teg va h.k teglar eskirgan deb belgilandi (o‘rniga CSS dan foydalanish tavsiya etiladi). Boshqa tomonidan, HTML 4.0 ko‘proq multimedia variantlarini, skript tillarini, kaskadli uslublar jadvallarini qo‘llab-quvvatlay boshladi. Til versiyasi HTML hujjatida hujjat turi ta’rifi (doctype) yordamida ko‘rsatilgan.

1998-yilda bo‘lib o‘tgan seminarda W3C belgilash tillari nuqtai nazaridan Internetning kelajagi HTMLni XML bilan moslashishga qarab rivojlantirishga qaror qildi. W3C 2000 yil boshida XHTML 1.0 (Kengaytiriladigan gipermatn belgilash tili) spetsifikatsiyasiga e’tibor qaratdi.

2004-yildan boshlab W3C (World Wide Web Consortium) HTML 5-versiyasini ishlab chiqishga rahbarlik qildi. 2007-yil 20-noyabrda Internetda til spetsifikatsiyasi loyihasi paydo bo‘ldi. HTML5 ko‘plab sintaktik xususiyatlarni taqdim etadi. HTML5 bir nechta yangi elementlar va atributlarni taqdim etadi, HTML 4.01 da foydalanish mumkin bo‘lgan ba’zi eskirgan elementlarni va freymlarni qo‘llab-quvvatlash to‘xtatilgan.

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

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



1.2-rasm. Belgilash tillarining rivojlanish bosqichlari

Bugungi kunda mobil aloqa vositalaridan foydalanuvchilar uchun yangi til ishlab chiqilgan bo‘lib, u **WML** (Wireless Markup Language) deb ataladi;

WML (English Wireless Markup Language – “simsiz qurilmalar uchun belgilash tili”) WAP standartiga muvofiq uyali telefonlar va boshqa mobil qurilmalarda foydalanish uchun hujjatlarni belgilash tilidir.

Tuzilishi biroz soddalashtirilgan HTMLga o‘xshaydi, ammo asosiy farqlar mavjud, chunki WML shaxsiy kompyuterlarning imkoniyatlariga ega bo‘lmagan qurilmalarga qaratilgan (kichik ekran, barcha qurilmalar grafiklarni ko‘rsata olmaydi, kichik xotira hajmi va boshqalar):

- ✓ WML dagi barcha ma’lumotlar “palubalar”da mavjud. Paluba(dek, inglizcha deck) – server tomonidan uzatilishi mumkin bo‘lgan ma’lumotlarning minimal blokidir. Deklarda “kartalar” (karta) mavjud (har bir karta <card> va </card> teglari bilan ajratilgan). Bitta palubada har doim kamida bitta karta bo‘lishi kerak, lekin bir nechta bo‘lishi ham mumkin. Shu bilan birga, qurilma ekranida bir vaqtning o‘zida faqat bitta karta ko‘rsatiladi va foydalanuvchi havolalarini bosish orqali ular o‘rtasida almashishi mumkin – bu serverga ma’lumot so‘rovlarni kamaytirish uchun amalga oshiriladi;
- ✓ shu bilan birga, WML sahifalarining hajmi 1-4 kilobaytdan oshmasligi kerak.

Biroq, ko‘plab mobil qurilmalar WML ni “haqiqiy” shaklida ham tushunmaydi, ular uchun saytning WBXML formati kerak.

1.3. “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 sstenariy deb ataladi. Bunday atama dasturning natijasiga bog‘liq holda vujudga kelgan. Uning asosiy vazifasi Web-sahifasida foydalanuvchi holatiga, harakatiga “reakstiya” berishdir. Shu tariqa sstenariylar klient tomonidan bajariluvchi va server tomonidan bajariluvchi sstenariylarga bo‘linadi. Klient tomonidan bajariluvchi sstenariylar

brauzer yordamida bajariladi. Server tomonida bajariluvchi sstenariylar esa Web-server yordamida bajariladi.

1.4. HTML tiliga kirish, uning asosiy elementlari

HTML5 tilida HTMLni yanada kuchliroq va web-ilovalarni yaratish uchun qulayroq qiladigan ko‘plab yangi xususiyatlar mavjud. Quyidagi ro‘yxat siz bilishingiz kerak bo‘lgan asosiy xususiyatlarni umumlashtiradi.

Yangi semantik elementlar: HTMLda semantika juda muhim – biz ishslash uchun doimo to‘g‘ri elementdan foydalanishimiz kerak. HTML 4.01 da jadvallar, ro‘yxatlar, sarlavhalar va boshqalar kabi maxsus xususiyatlarni aniqlash uchun juda ko‘p elementlar mavjud, ammo ularni aniqlash uchun elementga ega bo‘lmagan ko‘plab umumiylar web-sahifa xususiyatlari ham mavjud. Sayt sarlavhalari, navigatsiya menyulari va boshqa elementlarni biz <div id= “ ”> </div> bilan aniqladik, buni biz tushuna olamiz, lekin mashinalar tushuna olmaydi, bundan tashqari, turli web-ishlab chiquvchilar turli identifikatorlar va sinflardan foydalanadilar. HTML5 yangi semantik elementlarni o‘z ichiga oladi, masalan, <nav>, <header>, <footer> va <article>.

Formaning yangi xususiyatlari: HTML 4.01 allaqachon foydalanish mumkin bo‘lgan web-shakllarni yaratishga imkon beradi, ammo ba’zi umumiylar shakl xususiyatlari unchalik qulay emas va amalga oshirish uchun alohida harakatlar talab etiladi. HTML5 sana tanlash, slayderlar va mijoz tomonidan tekshirish kabi xususiyatlarni amalga oshirishning standartlashtirilgan, oson usulini taqdim etadi.

Mahalliy video va audioni qo‘llab-quvvatlash: ko‘p yillar davomida Internetda video va audio flash yordamida amalga oshirildi. XXI asr boshida flashning bu qadar mashhur bo‘lishining asl sababi shundaki, ochiq standartlar turli brauzerlar uchun shunga o‘xshash ishlarni bajarish uchun izchil mexanizmni ta’minlay olmadidi, shuning uchun turli brauzerlar bir xil ishni bajarishning turli xil raqobatdosh usullarini qo‘lladilar (masalan, <object>) va <embed>. Flash videoni brauzerlarda ishslashning yuqori sifatlari va yengil usulini taqdim etdi.

HTML5 o‘z video va audio pleyerlaringizni faqat ochiq standartlar yordamida osongina amalga oshirish uchun <video> va <audio> elementlarini, shuningdek, maxsus pleyer boshqaruvlarini osongina amalga oshirish imkonini beruvchi APIni ham o‘z ichiga oladi.

Oflayn web-ilovalar: HTML5 web-ilovalarni oflayn rejimda ishlashiga imkon beruvchi bir qator xususiyatlarni taqdim etadi. Ilova keshlari web-ilovani mahalliy sifatida ishga tushirish uchun zarur bo‘lgan barcha resurslar, boshqa fayllarning nusxasini saqlashga va Web SQL ma’lumotlar bazalari web-ilova ma’lumotlarining mahalliy nusxasini saqlashga imkon beradi. Ular birgalikda tarmoqqa ulanish bo‘limganda ilovadan foydalanishni davom ettirishga imkon beradi va tarmoq yana mavjud bo‘lganda serverdagи asosiy versiya bilan o‘zgarishlarni sinxronlashtirishga imkon beradi.

1.5. Web sahifaning tuzilishi va yaratish imkoniyatlari

HTML4 Web-sahifaning shakllar, ro‘yxatlar, paragraflar, jadvallar va boshqalar kabi turli xususiyatlarini aniq belgilash imkonini beruvchi ko‘plab semantik elementlarga ega. Biroq, uning kamchiliklari ham bor. Turli xil bo‘lgan <div> va elementlari sezilarli darajada ishlatiladi.

id va class atributlari navigatsiya menyulari, sarlavhalar va boshqalar kabi boshqa xususiyatlarni aniqlash uchun ishlatiladi.

Yangi HTML5 elementlari:

- ✓ <header>: Sayt sarlavhasi uchun ishlatiladi;
- ✓ <footer>: Saytning pastki qismi uchun ishlatiladi;
- ✓ <nav>: Sahifaning navigatsiya xususiyatlarini o‘z ichiga oladi;
- ✓ <article>: Quyidagi kontentni o‘z ichiga oladi, masalan,
- ✓ yangilik xabari;
- ✓ <section>: Turli maqsadlardagi yoki turli mavzulardagi turli maqolalarni guruqlash yoki bitta maqolaning turli bo‘limlarini belgilash uchun ishlatiladi;
- ✓ <time>: vaqt va sanani belgilash uchun ishlatiladi;

- ✓ <aside>: Asosiy content bilan bog‘langan, lekin uning asosiy oqimining bir qismi bo‘lidan kontent blokini aniqlaydi;
- ✓ <hgroup>: Agar sahifaning sarlavha tuzilishida faqat bitta sarlavha ko‘rib chiqilsa, bir nechta sarlavhalarni yashirish uchun o‘ram sifatida ishlataladi;
- ✓ <figure> va <figcaption>: formani bitta element sifatida inkapsulyatsiya qilish uchun ishlataladi va mos ravishda rasm uchun sarlavhani o‘z ichiga oladi.

Internetning eng ma’lum va keng tarqalgan qismi bu WWW (World Wide Web) deb nomlanuvchi to‘rdir. Bu to‘r Web saytlar va alohida Web sahifalardan tashkil topgan. Web sahifalar esa HTML tili yordamida yaratiladi.

HTML 4.0 versiyasiga qadar VERSION parametri mavjud edi. HTML4.0da 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.

Har qanday Web sahifa ikkita qismdan tashkil topadi. Bular sarlavha qismi va asosiy qism. Sarlavha qismida Web sahifa haqidagi malumot joylashadi, asosiy qismda esa Web sahifaning mazmuni bilan tasvirlanish qoidalari joylashadi. Sarlavha qismi quyidagi ochiluvchi <head> va yopiluvchi </head> teglari orasida joylashadi. Asosiy qism esa <body> va </body> teglari orasida joylashadi. Odatda sarlavha qismi oldidan qo‘llanilayotgan HTML standartlari haqida malumot yoziladi.

Web sahifa yaratish uchun turli dasturlar mavjud. Jumladan sublime, dreanweaver, phpstorm, notepad va boshqalar. Ushbu qo‘llanmadagi kodlar sublime dasturida yozilgan. Web sahifa yaratish uchun:

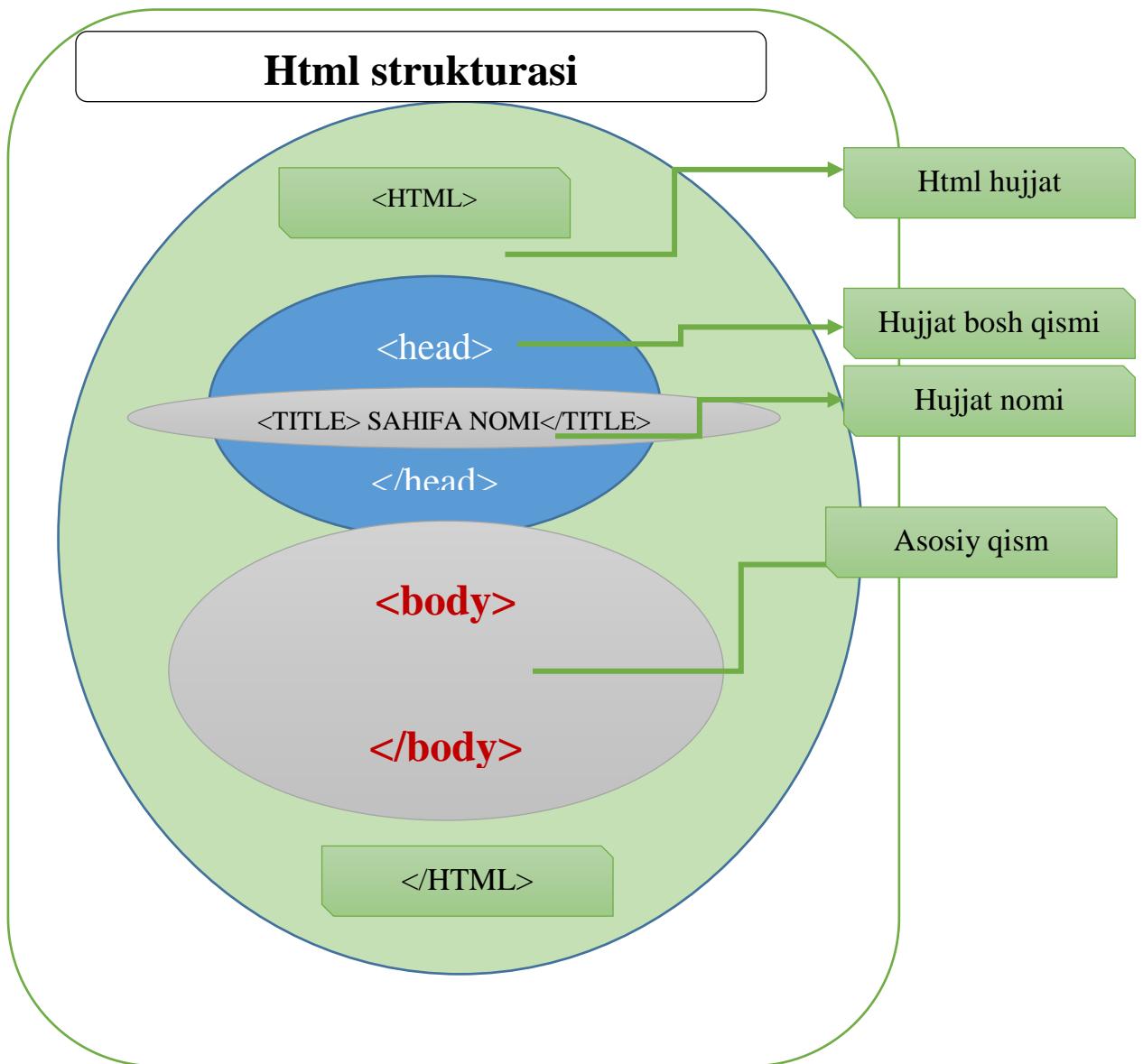
1. Sublime dasturini shaxsiy kompyuterga o‘rnatish va dasturga kirish kerak
2. Hujjatni .html kengaytmasi bilan saqlash
3. Quyida berilgan matnni terish (teglarni terish bilan teglar ro‘yxati chiqadi(chiqmasa ctrl+Probel). Kerakli teg yoki element ustiga bosilganda ochib yopiluvchi teg hosil bo‘ladi(1.3-rasm).

```

C:\Users\shohista\Desktop\index.html - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.html
1 <html>
2   <head>
3     <title></title>
4   </head>
5   <body>
6
7 </body>
8 </html>

```

1.3-rasm. Html kodini yozish



<head> – bu element hujjatning boshi hisoblanadi (inglizcha *HEAD* so‘zidan olingan). Bu qismga asosan yordamchi axborotlar kiritiladi.

<title> – ushbu element **<head>** elementi tarkibida bo‘lib, sahifaga sarlavha

berish uchun ishlataladi

<body> – hujjatning tana qismi (inglizcha BODY so‘zidan olingan.) Bu yerda o‘qiladigan yoki brouzerda ko‘rinadigan matnning asosiy qismi joylashadi.

Teglar ochilib yopiluvchi bo‘ladi:



Gipermatndagi xohlagan sahifani ochib, uning HTML kodini ko‘rishingiz mumkin. Buning uchun Web sahifasini ustida o‘ng tugmani bosib “просмотреть код страницы” punktini tanlash kerak. Bunda siz ko‘rib turgan sahifangizning HTML kodini ko‘rasiz.

HTML ning konstrukstiyasi TEG lar deyiladi. Brauzer TEGlar oddiy matnlardan farqlashi uchun burchak qavslarga olinadi. TEG tasvirlash jarayoni xatti xarakatlarining boshlanishini bildiradi. Agar bu harakat butun xujjatga talluqli bo‘lsa, bunday teg o‘zining yopiluvchi juftiga ega bo‘lmaydi. Juft teqlarning ikkinchisi birinchisining harakatini yakunlaydi. Masalan, har bir Web sahifa **<html>** tegi bilan boshlanib **</html>** tegi bilan yopilishi kerak. Etibor bergan bo‘lsangiz yopiluvchi teg ochiluvchidan “ / ” belgisi bilan farq qiladi. Teg nomlari katta yoki kichik harflar bilan yozilishi mumkin, bularni brauzer bir xil qabul qiladi.

HTML5 META tegi

META tegi hujjatning ba’zi xossalari (masalan: muallif, kalit so‘zlar ro‘yxati, kodlash va h.k.) o‘rnatish uchun ishlataladi, bu esa HTML hujjatiga ishlov berishni boshqarish imkonini beradi. Ushbu teg, TITLE tegi bilan birga, sarlavha o‘rnatishda eng ko‘p qo‘llaniladi.

META ni belgilashda boshlang‘ich yorlig‘i talab qilinadi, tugatish tegi taqiqланади.

Atributlar:

- ✓ **name** – xususiyat nomini ko‘rsatadi;

- ✓ **content** – xususiyat qiymatini belgilaydi;
- ✓ **scheme** – xususiyat qiymatini qayta ishlash uchun foydalaniladigan sxema nomini belgilaydi;
- ✓ **http-equiv** – http xabar nomini ko‘rsatish uchun path atributi o‘rniga ishlataladi;
- ✓ **lang** – til haqida ma’lumot. Majburiy bo‘lmagan atribut;
- ✓ **dir** – matnning yo‘nalishini ko‘rsatadi.

Har bir META tegida xususiyat-qiyamat juftligi mavjud. Name atributi (http-equiv) xususiyatni, kontent atributi qiymatini belgilaydi.

HTML hujjatining xususiyatlarini http sarlavhasi darajasida o‘rnatish uchun name atributi o‘rniga http-equiv atributidan foydalanish mumkin. http-equiv atributi HTTP sarlavhalari maydonlariga kiradi.

Ushbu elementdan birinchi foydali foydalanish brauzerni hujjatni qayta yuklashga majbur qilish edi. Qayta yuklash kerak bo‘lgan vaqt kontent atributi orqali, yuklangan hujjatning manzili esa yangilash operatorining url atributi orqali ko‘rsatiladi.

Misol:

```
<META http-equiv="refresh" content="5"; artel.html">
```

Ushbu vazifa bilan, joriy hujjat yuklangandan 5 soniya o‘tgach, brauzer avtomatik ravishda artel.html hujjatini yuklashni boshlaydi.

Agar url o‘rnatilmagan bo‘lsa, hujjat shunchaki yangilanadi.

META tegidan bunday foydalanish sizga avtomatik ravishda qayta yuklanadigan hujjatlar ketma-ketligini yaratish imkonini beradi.

Deyarli barcha HTML hujjatlari kalit so‘zlar ro‘yxatini va hujjat tarkibi haqida qisqacha ma’lumotni ko‘rsatish orqali ularni tavsiflash uchun META tegidan foydalanadi. Kalit so‘zlar hujjatning sarlavhasi bilan birga qidiruv tizimlariga hujjatni topishga yordam beradi. O‘z hisobotlarida ular TITLE elementida yozilgan hujjat nomini va META tegi orqali ko‘rsatilgan uning qisqacha tavsifini beradi.

Sarlavhada kalit so‘zlar ro‘yxatini va hujjat haqida qisqacha ma’lumotni ko‘rsatish uchun ikkita META tegidan foydalaniladi:

<META name= “description” content= “Arteldan maishiy texnikalar, kir moshina, changyutgich,sovutgich, kondensaner”>

<META name= “keywords” content= “Artel telefon, kir moshina, changyutgich,sovutgich, kondensaner”>

Yuqoridagi misollardan birida ko‘rsatilgandek, yanada samarali qidiruvlar uchun abstrakt va kalit so‘zlar odatda bir xil bo‘ladi.

META tegi orqali siz hujjat mazmunini kodlashni belgilashingiz mumkin. Keyin META tegi quyidagicha bo‘ladi:

<meta charset = “UTF-8”>

META yorlig‘i hujjatni keshlashni o‘chirish uchun ishlatilishi mumkin, bu hujjat tez-tez yangilanib tursa foydali bo‘ladi. Ushbu operatsiyani bajarish uchun META tegiga pragma operatori kiritiladi. Operator keshsiz rejimga o‘rnataladi. Keyin META tegi quyidagicha ko‘rinadi:

<META http-equiv= “pragma” content= “no-kesh”>

Hujjatni keshlashda siz keshlangan hujjat serverdagi asl nusxasiga mos keladigan vaqtini belgilashingiz mumkin. Bunday holda, muddati tugagan HTTP operatori ishlatiladi va META tegi quyidagi shaklni oladi:

<META http-equiv=“expires” content=“Dushanba, 18-May-2018 00:00:01”>

Eslatma: bu holda, sana HTTP sarlavhasini tuzatish uchun o‘rnataladi va misolda ko‘rsatilganidek formatga ega bo‘lishi kerak.

BODY teg atributlaridan foydalanish

Hujjatning kontent qismi hujjatning asosiy qismni tashkil etadi. BODY konteyner tegi hujjatning asosiy qismi sifatida aniqlanadi.

Atributlar:

- ✓ **alink**=color – faol (foydalanuvchi tanlagan) havolaning rangini aniqlaydi;
- ✓ **link**=color – hali ko‘rilmagan havola rangini belgilaydi;
- ✓ **vlink**=color – allaqachon ko‘rilgan havola rangini o‘rnataladi;
- ✓ **bgcolor**=color – hujjat foni qanday rangda bo‘lishi kerakligini belgilaydi;

- ✓ ***background***=URL – fon sifatida foydalaniladigan tasvirning manzilini (URL) belgilaydi;
- ✓ ***bgproperties*** – agar FIXED qiymati o‘rnatilmagan bo‘lsa, fon tasviri aylantirilmaydi;
- ✓ ***text***=color – hujjat matnining rangini belgilaydi;
- ✓ ***leftmargin, rightmargin, topmargin***=pixels – hujjatning mos ravishda chap, o‘ng va yuqori chetlari hajmini o‘rnatish. Piksellarda ko‘rsatilgan;
- ✓ ***scroll*** – mantiqiy atributdir: brauzer darchalari harakatlantirish yo‘laklarini o‘rnatadi.

Misol.

```
<body leftmargin=100 rightmargin=100 text=#78bed0 bgcolor=#112232
scroll=no Link=#fe4918 vlink=#ffffff alink=#ffffff>
</body>
```

1.6. Matnlar bilan ishslash

➤ ***Sarlavha elementi teglari.*** HTML hujjatida sarlavhaning o‘z teglari mavjud bo‘lib, ular oltitadir. Eng yuqori darjasini bu birinchidir. Har bir sarlavhaning o‘z tegi va o‘z tasvirlanish qoidasi mavjud. Eng katta ya’ni birinchi darajali sarlavha **<h1>** va **</h1>** teglari orqali, ikkinchi darajali sarlavha **<h2>** va **</h2>** teglari orqali va oxirgi oltinchi darajali sarlavha **<h6>** va **</h6>** teglari orqali ifodalanadi.

<h1> elementining o‘lchami eng kattasi bo‘lib, **<h6>** element eng kichikdir(1.4-rasm).

Atributlar, barchasi ixtiyoriy:

- ✓ ***align*** – tekislash, quyidagi qiymatlarni qabul qilishi mumkin:
 - *left(chapga tekislash);*
 - *right(o‘ngga tekislash);*
 - *center(markazga tekislash);*
 - *justify(kenglik bo‘yicha tekislash).*
- ✓ ***id, class*** – hujjat ichidagi identifikatorlar;
- ✓ ***lang, dir*** – matnning tili va yo‘nalishi haqidagi ma’lumotlar;

- ✓ ***title*** – elementning sarlavhasi (element mazmuni ustiga kursov olib kelganda brauzer tomonidan izoh sifatida ko‘rsatiladi);
- ✓ ***style*** – o‘rnatilgan uslub haqida ma’lumot.

Har bir sarlovaning aniq o‘lchamini brauzer orqali ko‘rish mumkin. Ularning o‘lchami bir-biridan biroz farq qiladi.

1.4-rasm. Sarlovhalar bilan ishslash

➤ **Paragraflar bilan ishslash.** HTML paragraflarga bo‘linish uchun o‘z vositalariga ega. Bu funksiya uchun blok darajasidagi P tegidan foydalilanadi. Boshlash va tugatish teqlari orasidagi matn brauzerlar tomonidan paragraf sifatida qaraladi.

<p> tegi – Paragraf yaratish uchun ishlatiladi. <p> ochiluvchi paragraph tegi va </p> tegiluvchi paragraph tegi.

Ushbu <p> </p> teqlari ichida yozilgan matn xatboshidan, yangi qatordan yoziladi. Paragraf bir yoki bir nechta gaplardan tashkil topgan bo‘ladi.

Atributlar, barchasi ixtiyoriy:

- ✓ ***align*** – tekislash, quyidagi qiymatlarni qabul qilishi mumkin:
 - *left(chapga tekislash);*
 - *right(o ‘ngga tekislash);*
 - *center(markazga tekislash);*
 - *justify(kenglik bo ‘yicha tekislash.*
- ✓ ***id, class*** – hujjat ichidagi identifikatorlar;
- ✓ ***lang, dir*** – matnning tili va yo‘nalishi haqidagi ma’lumotlar;

✓ **title** – elementning sarlavhasi (element mazmuni ustiga kursov olib kelganda brauzer tomonidan izoh sifatida ko'rsatiladi);

✓ **style** – o'rnatilgan uslub haqida ma'lumot;

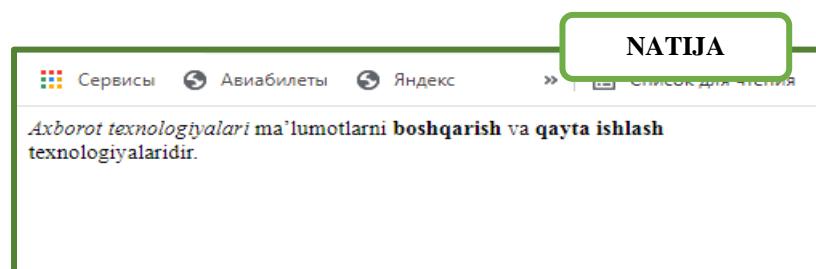
✓ onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup – ichki hodisalar.

➤ **Bold(so'zlarni qoraytirish) & Italic(so'zlarni og'dirib yozish)** (1.5-rasm).

HTML

```
1 <html>
2 <head>
3   <title></title>
4 </head>
5 <body>
6 <p><i>Axborot texnologiyalari </i>
7 ma'lumotlarni <b>boshqarish </b> va <b>
8 qayta ishlash </b>texnologiyalaridir. </
p>
9 </body>
10 </html>
```

HTML



NATIJA

** ** – So'zlarni ushbu teglar ichida yozish orqali so'zlar qalin ko'rinadi.

<i> </i> – So'zlarni ushbu teglar ichida yozish orqali so'zlar og'dirib yoziladi.

1.5-rasm. Qoraytirib va og'dirib yozish

➤ **Superscript(darajaga yozish) & Subscript(indeksga yozish)** (1.6-rasm).

HTML

```
<html>
<head>
  <title></title>
</head>
<body>
  <p> Suvning formulasi  $H<sub>2</sub>O$  dir. </p>
  <p>Kvadrat tenglamining formulasi esa  $y = ax^2 + bx + c$  </p>
</body>
</html>
```

<sup> elementi matematikada yoki sonlarni darajada yozish uchun ishlatiladi. $x=x^4+y^4$



NATIJA

<sub> elementi esa ko‘proq kimyoviy va matematik formularni yozishda ishlataladi. Ushbu element yordamida indekslar yoziladi. H₂O

1.6-rasm. Daraja va indeksga yozish

➤ **Line Breaks(keying qatorga o‘tish) &Horizontal Rules(gorizontall chiziq chizish** (1.7-rasm).

HTML eng oddiy gorizontal chiziqlarni kiritish qobiliyatini amalga oshiradi. HR tegi yordamida eng oddiy chiziq chizish mumkin.

Ushbu qatorning vizual parametrlari HR tegining atributlari yordamida o‘rnatiladi:

✓ **align** – tekislash, quyidagi qiymatlarni qabul qilishi mumkin:

- *left: chapga tekislash;*
- *right, o‘ngga tekislash;*
- *center, markazda tekislash.*

✓ **size** – bu atribut chiziq qalinligini piksellarda o‘rnatadi;

✓ **width** – piksellardagi chiziq uzunligi yoki brauzer oynasi kengligining foizi;

✓ **noshade** – mantiqiy atribut, uning o‘rnatilishi chiziq relyefini bekor qiladi.

Odatiy bo‘lib, chiziq relyefda chizilgan;

✓ **color** – bu atribut chiziq rangini rangni bildiruvchi so‘z(inlistilida) ko‘rinishida yoki RGB kod ko‘rinishida ko‘rsatadi.

```
1 <html>
2 <head>
3   <title></title>
4 </head>
5 <body>
6 <p> Axborot texnologiyalari<br>
7   ma'lumotlarni boshqarish va <br> qayta
8   ishlash texnologiyalaridir
9 <hr>
10 </body>
11 </html>
```

HTML

NATIJA

Axborot texnologiyalari
ma'lumotlarni boshqarish va
qayta ishlash texnologiyalaridir

 browser sarlavhalar yoki paragraflarni avtomatik holda yangi qatordan yozadi. Ammo paragraph ichiga yangi qator qo‘shmoqchi bo‘lsangiz yangi qator qo‘shish tegi
 dan foydalanishingiz mumkin.

<hr> tegi esa gorizontal to‘g‘ri chiziq chizish uchun ishlataladi

1.7-rasm. Keying qatorga o‘tish va gorizontal chiziq chizish

➤ Abbreviations & Acronyms (So‘zlarni qisqartma ko‘rinishida yozish (1.8-rasm).

```

1 <html>
2 <head>
3   <title></title>
4 </head>
5 <body>
6 TVCHDPI<abbr title="Toshkent
viloyati Chirchiq davlat pedagogika
instituti"></abbr>
7 </body>
8 </html>

```

<abbr> yoki <acronym>. Agar siz qisqartma so‘zlarni ishlatsangiz <abbr> yoki <acronym> teglaridan foydalanish mumkin.

1.8-rasm. So‘zlarni qisqartma ko‘rinishida yozish

➤ <ins>(So‘zlarning tagiga chizish)& (So‘zlarni ustidan o‘chirish)

<ins> tegidan foydalanish orqali so‘zlarni tagiga chiziq chiziladi. Buning uchun <ins> </ins> ochib yopiluvchi teg ichida so‘zlar yozilishi kerak.

 tegi so‘zlarni ustidan chizish uchun ishlataladi(1.9-rasm).

```

<html>
<head>
  <title></title>
</head>
<body>
  Sale 50000 <del>100000</del> <br>
  <ins>So‘zlarni tagiga chizish</ins>
</body>
</html>

```

NATIJA
Sale 50000 ~~100000~~
So‘zlarni tagiga chizish

1.9-rasm. So‘zlarni tagiga chizish va so‘zlarni ustidan o‘chirish

➤ **PRE tegi.** Ushbu tegning mavjudligi brauzerga ilova qilingan matn formatlangan matn ekanligini bildiradi. Bunday matn brauzerlar tomonidan hujjatning HTML matnida mavjud bo‘lgan shaklda ko‘rsatiladi(1.10-rasm).

NATIJA
PRE tegidan foydalanish

1.10-rasm. Pre tegidan foydalanish

NAZORAT SAVOLLARI

- 1.1. Web-sahifa, Web-sayt, Web-server tushunchalariga ta’rif bering
- 1.2. Domen va xosting nima? Misollar keltiring
- 1.3. Belgilash tillari va ularning rivojlanish bosqichlari haqida ma’lumot bering?
- 1.4. Html 5 va html 4 qanday xususiyatlari bilan ajralib turadi?
- 1.5. Html tili strukturasini ifodalang
- 1.6. Meta va Body tegi atributlari va vazifalarini aniqlang?
- 1.7. Matnlar bilan ishlovchi teglarni va ularni xususiyatlarini aytинг.

II-BOB. HTML DA RO‘YXATLAR, JADVALLAR VA XAVOLALAR BILAN ISHLASH. HTMLDA MULTIMEDIA OBEKTLARI

2.1. Htmlda ro‘yxatlar bilan ishlash

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

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

HTML tilida ro‘yxatlarning quyidagi asosiy turlari mavjud:

- markerlangan(tartibsiz);
- raqamlangan(tartiblangan);
- tariflar ro‘yxati.

Yana boshqa xillar ham bor, lekin yuqoridagi uchta ro‘yxat turlari eng ko‘p ishlatiladi. Ularning o‘zaro o‘xshashligi shundaki, ularning barchasi xat boshining har bir punktiga joylashgan bo‘ladi va siz bemalol, har bir matnni ajratishingiz mumkin. Yuqorida keltirilgan ro‘yxat turlarining asosiy farqi raqamlashtirish usulida va uning strukturasidadir.

Har qanday ro‘yxat kamida bitta elementni o‘z ichiga olishi kerak, aks holda u e’tiborga olinmaydi.

Tartibsiz ro‘yxatlar UL tegi bilan, tartiblangan ro‘yxatlar OL tegi bilan tuziladi. Ikkala turdagи ro‘yxatlar LI tegi bilan belgilanadigan ro‘yxat elementlari ketma-ketligidan iborat. LI tegi, aniqrog‘i uning tarkibi (masalan, kafedralalar nomi) ro‘yxatning alohida qismidir. Tartibsiz ro‘yxatlar brauzerlar tomonidan markirovka qilingan ro‘yxatlar ko‘rinishida, tartiblangan ro‘yxatlar esa raqamlangan holda ko‘rsatiladi. tegida ikkita parametr ko‘rsatilishi mumkin, COMPACT va TYPE. COMPACT parametri ma‘nosiz yoziladi va mazkur ro‘yxatni kompakt ko‘rinishida chiqarish zarurligini brauzerga biriktirib qo‘yish uchun ishlatiladi. Masalan, shriftni yoki ro‘yxat satrlari orasidagi masofani kichraytirish mumkin va b.q.

TYPE parametri ro‘yxat satri belgilanadigan nishonni berish (yusinfh) imkonini yaratadi; uchtaidan bittasiga ruxsat berilgan: diss, circle va square.

Tartibsiz (markerli) ro‘yxatni qanday yaratish mumkin?

Ushbu tegni ochish va yopish majburiy.

Atribut (ixtiyoriy):

✓ ***type*** – ishlataladigan markerlar turi haqidagi ma’lumotlarni belgilaydi. U quyidagi qiymatlarni qabul qilishi mumkin:

- *circle* – markerlar to ‘ldirilmagan doiralar sifatida ko ‘rsatiladi;
- *disc* – markerlar to ‘ldirilgan doiralar sifatida ko ‘rsatiladi;
- *square* – markerlar to ‘ldirilgan kvadratchalar sifatida ko ‘rsatiladi.

!!!Eslatma: Agar siz *type* atributiga qiymat qo ‘ymasangiz, brauzer o‘zi xohlagancha markerlardan foydalanadi (odatda to ‘ldirilgan doiralar).

Standart ixtiyoriy atributlar:

- ✓ ***id, class*** – hujjat ichidagi identifikatorlar;
- ✓ ***lang, dir*** – matnning tili va yo‘nalishi haqidagi ma’lumotlar;
- ✓ ***title*** – elementning sarlavhasi (element mazmuni ustiga kursov olib kelganda brauzer tomonidan izoh sifatida ko ‘rsatiladi);
- ✓ ***style*** – o‘rnatilgan uslub haqida ma’lumot;
- ✓ onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup – ichki hodisalar.

UL tegiga kiritilgan ro‘yxat kontekstdan yuqoridagi (ro‘yxat boshidan oldin) va pastdagi (ro‘yxatning oxirgi elementidan keyin) bo‘sh satrlar bilan ajratiladi. UL tegiga kiritilgan barcha ma’lumotlar o‘ngga siljigan holda ko ‘rsatiladi.

Ro‘yxatning har bir elementi (bu tartiblangan ro‘yxatga ham tegishli) HTML LI tegidir. Ro‘yxatning har bir elementi teg (/LI–List Item–ro‘yxati elementi) bilan boshlanishi kerak. Teg mos yopuvchi tegga muhtoj emas, lekin uning mavjudligi xato hisoblanmaydi. Bunday holda, ro‘yxatning elementi keyingi ochilishgacha joylashgan barcha matn hisoblanadi. Brauzerlar hujjatni aks ettirishda har bir yangi ro‘yxat elementini xat boshidan boshlaydi(2.1-rasm).

HTML tili bo'yicha ba'zi darsliklarda ro'yxat sarlavhasi uchun teg – konteyner <LH> (LH–List Header ro'yxat sarlavhasini) ishlab chiqish kerakligi to'g'risidagi ko'rsatmalar uchrab turadi. Hozirgi paytda nomi chiqqan brauzerlardan

```
1 <html>
2 <head>
3   <title></title>
4 </head>
5 <body>
6   <ul> <b>Matematika va informatika</b>
7     fakultetida quyidagi kafedralar mavjud:</b>
8     <li>Informatika va axborot texnologiyalari</li>
9     <li> Informatika o'qitish metodikasi</li>
10    <li>Matematika o'qitish metodikasi va geometriya</li>
11    <li>Algebra va sonlar nazariyasi</li>
12  </ul>
13 </body>
```

HTML

NATIJA

Matematika va informatika fakultetida quyidagi kafedralar mavjud:

- Informatika va axborot texnologiyalari
- Informatika o'qitish metodikasi
- Matematika o'qitish metodikasi va geometriya
- Algebra va sonlar nazariyasi

birortasi ham bu tegni tan olmaydi va u HTML spetsifikasiga kirmaydi. Shunday qilib bu teg bitta – yarimta xatoliklarni keltirib chiqarmasa ham, uni qo'llash hech qanaqa ma'no kasb etmaydi.

! Tartiblanmagan ro'yxatlar

 tegi ichida beriladi. Ro'yxat ichidagi xar bir element esa teglari yordamida hosil qilinadi(2.1-rasm).

2.1-rasm. Tartiblanmagan ro'yxatlar

Tartiblangan (raqamlangan) ro'yxat qanday tuziladi?

 tegi HTML hujjalarda tartiblangan (raqamlangan) ro'yxatlarni amalga oshirish uchun ishlatiladi. Tartibsiz ro'yxatlarda bo'lgani kabi, tartiblangan ro'yxatning elementi tegi bilan beriladi.

Ushbu tegni ochish va yopish majburiy.

Atributlar (barchasi ixtiyoriy):

✓ ***type*** – tartiblangan ro'yxat elementlarini raqamlash turini ko'rsatadi. HTML spetsifikatsiyasida eskirgan. **U quyidagi qiymatlarni olishi mumkin:**

- *type =I* – arab raqamlari bo'yicha raqamlashni o'rnatadi (sukut bo'yicha brauzerlar tomonidan qo'llaniladi);
- *type=A* – bosh lotin harflarida raqamlashni o'rnatadi;
- *type=a* – kichik lotin harflarida raqamlashni o'rnatadi;
- *type=I* – katta rim raqamlari bilan raqamlashni belgilaydi;
- *type=i* – kichik lotin raqamlarida raqamlashni o'rnatadi.

✓ ***start*** – tartiblangan ro‘yxatdagi birinchi elementning boshlang‘ich raqamini belgilaydi. Qiymat sifatida u raqamlash turidan qat’iy nazar faqat natural sonni qabul qilishi mumkin.

Standart ixtiyoriy atributlar:

- ✓ ***id, class*** – hujjat ichidagi identifikatorlar;
- ✓ ***lang, dir*** – matnning tili va yo‘nalishi haqidagi ma’lumotlar;
- ✓ ***title*** – elementning sarlavhasi (element mazmuni ustiga kursov olib kelganda brauzer tomonidan izoh sifatida ko‘rsatiladi);
- ✓ ***style*** – o‘rnatilgan uslub haqida ma’lumot;
- ✓ **onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup** – ichki hodisalar.

Raqamlash turi va HTML tegi raqamining qiymati LI ro‘yxati elementini belgilashda bevosita o‘zgartirilishi mumkin. Bunda yuqorida tavsiflangan type atributi va LI tegi uchun belgilangan qiymat atributidan foydalaniladi. LI ro‘yxat elementi uchun qiymat atributining qiymatini o‘rnatish ro‘yxat elementining raqamini o‘zgartirishga imkon beradi. Bu ro‘yxatning barcha keyingi elementlarining raqamlanishini o‘zgartiradi. OL tegining start atributi natural sonlarni qiymat sifatida qabul qiladi.

```
1 <html>
2 <head>
3   <title></title>
4 </head>
5 <body>
6   <ol> Matematika va informatika
7     fakultetida quyidagi kafedralar mavjud:
8       <li>Informatika va axborot
9         texnologiyalari</li>
10      <li> Informatika o'qitish metodikasi</li>
11      <li>Matematika o'qitish metodikasi va
12        geometriya</li>
13        <li>Algebra va sonlar nazariyasi</li>
14    </ol>
15  </body>
16 </html>
```

HTML

! Markerlangan ro‘yxatlar ** ** tegi ichida beriladi. Ro‘yxat ichidagi xar bir element esa ** ** teglari yordamida hosil qilinadi(2.2-rasm).

NATIJA

Matematika va informatika fakultetida quyidagi kafedralar mavjud:

1. Informatika va axborot texnologiyalari
2. Informatika o'qitish metodikasi
3. Matematika o'qitish metodikasi va geometriya
4. Algebra va sonlar nazariyasi

2.2-rasm. Tartiblangan ro‘yxatlar

Sahifaning jozibadorligini oshirish uchun siz nostonart belgilar bilan ro'yxatlar yaratishingiz mumkin, masalan: yulduzcha, to'p va boshqalar. Buning uchun har bir ro'yxat tegi uchun LI tegi o'rniga IMG tegidan foydalaniladi va BR tegini kiritish orqali yangi qatorga o'tish mumkin.

HTML

```
<!DOCTYPE HTML>
<html><head>
<title> Неупорядоченный (маркированный) список </title>
<meta charset="utf-8">
</head>
<body leftmargin=100 rightmargin=100 text=#78bed0 bgcolor=white>
<h1> nostonart ro'yxat </h1>
<ul>
<h3 align=center> Marafon yo'nalishlari: </h3>
 Uchuvchisiz uchish apparatlari <br>
 Koinot texnologiyasi <br>
 Robototexnika <br>
 Avtomatlashtirilgan boshqaruv tizimlari <br>
 Sanoat dizayni <br>
 Neyrotexnologiya <br>
</ul></body> </html>
```

nostonart ro'yxat

NATIJA

Marafon yo'nalishlari:

- ★ Uchuvchisiz uchish apparatlari
- ★ Koinot texnologiyasi
- ★ Robototexnika
- ★ Avtomatlashtirilgan boshqaruv tizimlari
- ★ Sanoat dizayni
- ★ Neyrotexnologiya

2.3-rasm. Nostonart ro'yxatlar

Ta'riflar ro'yxati qanday tuziladi?

Ta'riflar ro'yxatlari – lug'at yozuvlari ko'rinishidagi ma'lumotlarni ifodalovchi ro'yxatlarning maxsus turi: belgilangan atama va uning chegaralangan ta'rifi. Boshqa ro'yxatlardan farqli o'laroq, bunday ro'yxatning har bir elementi ikki qismdan iborat:

- ✓ <*dt*></*dt*> (Definition Term) tegi yordamida belgilangan atama;

- ✓ <dd></dd> (Definition Description) tegi yordamida ko'rsatilgan atama tarifi(matn).

Ta'riflar ro'yxatining o'zi DL (Definition List) konteyner tegining tarkibiga kiritilgan.

dl, dt va dd teglari faqat standart ixtiyoriy atributlarga ega:

- ✓ *id, class* – hujjat ichidagi identifikatorlar;
- ✓ *lang, dir* – matnning tili va yo'nalishi haqidagi ma'lumotlar;
- ✓ *title* – elementning sarlavhasi (element mazmuni ustiga cursor olib kelganda brauzer tomonidan izoh sifatida ko'rsatiladi);
- ✓ *style* – o'rnatilgan uslub haqida ma'lumot;
- ✓ onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup – ichki hodisalar.

dl tegini ohib yopish majburiydir. dt va dd teglarini ochish talab qilinadi, lekin tegni yopish ixtiyoriydir.

dt tegi kontentida blok darajasidagi teglardan foydalanishga (masalan, h1...h6 sarlavha elementlari yoki P paragraf elementi) ruxsat berilmaydi. dd tegining kontentida ulardan foydalanishga ruxsat beriladi. Bu ta'riflarning ichki ro'yxatlarini yaratish imkoniyatini ko'rsatadi(2.4-rasm).

Ta'riflar ro'yxatidan foydalanishga misol:

<pre><!DOCTYPE html> <html> <head> <title> Ta'riflar ro'yxatiga misol </title> </head> <body leftmargin=100 rightmargin=100 text=blue bgcolor=lightgrey> <h2 align=center> Matematik ma'lumotnoma </h2> <dl> <dt>Nol bo'linuvchi</dt> <dd> Bu halqa yoki yarim guruhning nolga teng bo'limgan elementi bo'lib, ularning mahsuloti va nolga teng bo'limgan ba'zi elementlar nolga teng. Kommutativ bo'limgan holatda chap va o'ng nol bo'luvchilar farqlanadi. </dd> <dt>Diskret matematika</dt> <dd> Matematikaning ham matematikada, ham uni qo'llashda yuzaga keladigan diskret tuzilmalarning xossalarni o'rganish bilan shug'ullanuvchi bo'limi. Bunday tuzilmalarga, masalan, chekli guruhlar, chekli grafiklar, shuningdek, axborotni o'zgartiruvchi ba'zi matematik modellar, chekli avtomatlar, Tyuring mashinalari va boshqalar kiradi.</dd></dl> </body> </html></pre>	HTML
--	------

Matematik ma'lumotnoma**Nol bo'limuvchi**

Bu halqa yoki yaram guruhesung nolga teng bo'lmagan elementi bo'lib, ularning mahsuloti va nolga teng bo'lmagan bo'zi elementlar nolga teng. Kommunikativ bo'lmagan holanda chap va o'ng nol bo'lvchilar farqlanadi.

Delfi Challenge

Kub mazmumonini ikki baravar oshirish. U kubning bir tomonini surʼuj va o'shangich yordamida qurishdan iborat bo'lib, uning hajmi berilgan kub hajmidan ikki baravar kana "Delfan varifasi" nomi afsona bilan bog'liq bo'lib, unga ko'ta Delos oroli sholisi o'latidan qochish uchun Delfi o'sakulimning buyruq ini bajarishlari kerak edi: qurbongob hajmini kubini o'gartirmaidan ikki baravar oshirish shakli.

Diskret matematika

Matemaniqning haen matematikada, ham esa qo'llashda yuzaga keladigan diskret turzimalarning xossalarni o'reganish bilan shug'ullanuvchi bo'limi. Bunday turzimalarga, masalan, chekli guruhi, chekli grafiklar, shuningdek, xaborotni o'gartiruvchi ba'ts matematik modellar, chekli avtomatlar, Tyuring mashinalari va boshqalar kiradi.

Diskret dasturlash

Matematik dasturlashning chekli to'plamalarda aniqlangan funksiyalarning ekatreneallarini topishiga bag'ishlangan bo'limi. Diskret dasturlash masalalari shaniyatsuz bo'lib, real masalalarda amalga oshirish mumkin bo'lgan yechimlar soni shuchochlik ko'pi, ularni to'liq sanab chiqish amalda amalga oshirib bo'lmasdi.

2.4-rasm. Tariflar ro'yxatlar**2.2. Htmlda jadvallar bilan ishlash**

Hujjatlarda ma'lumotlarni taqdim etishning eng keng keng tarqalgan va samarali usullaridan biri bu jadvallardan foydalanishdir. HTMLda jadvallardan foydalanish umumiyoqdir. Ular ishlab chiquvchilarga hujjatdagi ma'lumotlarni tartibga solish imkonini beradi: matn, rasmlar, ob'ektlar, formalar, shakl maydonlari va boshqalar. Bu shuni anglatadiki, jadvallarni shunday qurishdan tashqari, jadval teglari hujjatni formatlash vositasi sifatida ishlatiladi. Ko'rsatilmaydigan (ko'rinnmas) chegaralari bo'lgan jadvallar uzoq vaqtadan beri web-sahifalarni joylashtirish uchun ishlatilgan, bu sizga sahifani maydonlarini to'rtburchaklarga ajratish imkonini beradi, ularning har biri o'z ma'lumotlarini o'z ichiga olishi mumkin. Zamonaviy qatlamlardan foydalanmaguncha, jadvallardan foydalanishning bu usuli ko'plab saytlarda o'z o'rnnini egalladi.

HTML hujjati istalgan sonli jadvallarni o'z ichiga olishi mumkin va jadvallar bir-birining ichiga joylashtirilishi mumkin. Har bir jadval <TABLE> konteyner tegining mazmuni va kamida bitta qatordan iborat bo'ladi. Har bir satr <TR> (Jadval qatori) tegi bilan belgilanadi, uning ichida kataklarga bo'linish amalga oshiriladi. Qatordagi har bir katak <Th> tegi (Jadval sarlavhasi) yoki <TD> tegi (Jadval ma'lumotlari) orqali ifodalanadi. Ushbu teglardan foydalanishdagi farq ular o'z ichiga olgan ma'lumotlarning turli xil vizual tasvirida yotadi:

- ✓ <th> </th> tegi – undagi ma'lumotlar qalin bo'lib ko'rsatiladi va katak markaziga tekislanadi (align=center; valign=middle);

- ✓ *<td> </td>* tegi – undagi ma'lumotlar odatda oddiy shriftda ko'rsatiladi, gorizontal ravishda chapga (align=chap) tekislanadi va vertikal ravishda markazlashtiriladi (valign=middle).

Jadvaldagи qatorlar soni TR teglari soni bilan belgilanadi.

TABLE tegi jadval uchun konteyner tegidir. U jadvalni belgilaydi. Jadvalni tuzish va formatlash uchun barcha teglar uning tarkibiga kiritilishi kerak. Tegni ochib yopish majburiydir. TABLE tegining barcha atributlari ixtiyoriydir. HTML 5 spetsifikatsiyasida ular nomaqbul deb e'lon qilingan, atributlar o'rniha CSS kaskadli jadval uslublaridan foydalanish tavsiya etiladi.

- ✓ ***align*** – hujjatdagи jadvalning o'zi uchun tekislash variantlarini o'rnatadi.

Mumkin bo'lган qiymatlar:

- *left – jadval hujjatning chap chetiga biriktirilgan;*
- *right – jadval hujjatning o'ng chetiga biriktirilgan;*
- *center – jadval hujjatning markazida joylashgan.*

- ✓ ***width*** – tavsiya etilgan jadval kengligini o'rnatadi. Atribut qiymati piksel yoki foizda beriladi. Foizni belgilashda 100% dan ortiq qiymatlarni belgilashga ruxsat beriladi, masalan, width = 200%. Ushbu atributni o'rnatish jadvalning atribut qiymatida ko'rsatilgan kenglikka ega bo'lishiga kafolat bermaydi, chunki bu o'lcham qat'iy talab qilinmaydi, lekin tavsiya etiladi;

- ✓ ***border*** – jadval atrofidagi chegaraning ko'rinishi va qalinligini va har bir katak atrofidagi chegaraning ko'rinishini nazorat qiladi. border atributidan foydalanish kontekstida *<TABLE>* tegini o'rnatish uchta usulda bo'lishi mumkin:

- *<TABLE> – Chegara atributi o'rnatilmagan. Bunday holda, chegaralar chizilmaydi. Ekran faqat katakchalardagi ma'lumotlarni formatlangan shaklda ko'rsatadi;*
- *<TABLE border> – Border atributi o'rnatilgan, lekin uning qiymati o'rnatilmagan;*
- *<TABLE border=pixels> – Border atributi va uning qiymati piksellarda o'rnatiladi. Bunday holda, jadval atrofidagi chegara qiymatida*

ko ‘rsatilgan qalinlik bilan ko ‘rsatiladi, lekin katakchalar atrofidagi chegaralar hali ham 1 piksel qalinlikda ko ‘rsatiladi(2.5-rasm).

- ✓ **cellspacing** – gorizontal va vertikal ravishda qo‘shni katakchalar orasidagi (aniqrog‘i ularning ramkalari orasidagi) piksellardagi masofani ko‘rsatadi;
- ✓ **cellpadding** – ramka va katakcha tarkibi o‘rtasidagi to‘ldirish miqdorini (gorizontal va vertikal) piksellarda belgilaydi;
- ✓ bgcolor – jadvalning katakchalar tarkibiga va ular orasidagi bo‘sh joyga foni rangini o‘rnatadi.

HTML

```
<!DOCTYPE html>
<html>
<head>
<title> Oddiy jadvalga misol </title>
</head>
<body leftmargin=100 rightmargin=100 text=red bgcolor=white
scroll=no>
<h2> Eng oddiy jadval misoli </h2>
<table chegarasi WIDTH=500 border="1">
<tr>
<th> 1-qatorning 1-katakchasi</th>
<th> 1-qatorning 2-katakchasi</th>
<th> 1-qatorning 3-katakchasi</th>
</tr>
<tr>
<td> 2-qatorning 1-katakchasi</td>
<td> 2-qatorning 2-katakchasi</td>
</tr>
<tr>
<td> 3-qatorning 1-katakchasi</td>
<td> 3-qatorning 2-katakchasi</td>
<td> 3-qatorning 3-katakchasi</td>
</tr>
<tr>
<td> 4-qatorning 1-katakchasi</td>
</tr>
</table>
</body>
</html>
```

Eng oddiy jadval misoli

1-qatorning 1-katakchasi	1-qatorning 2-katakchasi	1-qatorning 3-katakchasi
2-qatorning 1-katakchasi	2-qatorning 2-katakchasi	
3-qatorning 1-katakchasi	3-qatorning 2-katakchasi	3-qatorning 3-katakchasi
4-qatorning 1-katakchasi		

2.5-rasm. Jadval yaratish

Standart ixtiyoriy atributlar:

- ✓ *id, class* – hujjat ichidagi identifikatorlar;
- ✓ *lang, dir* – matnning tili va yo‘nalishi haqidagi ma’lumotlar;
- ✓ *title* – elementning sarlavhasi (element mazmuni ustiga kursov olib kelganda brauzer tomonidan izoh sifatida ko‘rsatiladi);
- ✓ *style* – o‘rnatilgan uslub haqida ma’lumot.

Jadval sarlavhasini qanday qo‘sish kerak?

<CAPTION></CAPTION> tegidan foydalanib, jadval nomini o‘rnatishingiz mumkin. Bitta jadval faqat bitta nomga ega bo‘lishi mumkin (agar bir nechta ko‘rsatilgan bo‘lsa, ular e’tiborga olinmaydi)

<CAPTION> tegini ochib yopish majburiydir. Barcha atributlar ixtiyoriy:

- ✓ *Align* – bu jadval nomining joylashuvini belgilovchi ixtiyoriy atribut. U quyidagi qiymatlarni olishi mumkin:
 - top – sarlavha jadvalning yuqori qismida, jadvalning o‘rtasiga tekislangan holda ko‘rsatiladi. Bu qiymat standart hisoblanadi;
 - Button – sarlavha jadvalning pastki qismida, jadvalning o‘rtasiga tekislangan holda ko‘rsatiladi;
 - Left – sarlavha jadvalning yuqori qismida, jadvalning chap chetiga tekislangan holda ko‘rsatiladi;
 - Right – sarlavha jadvalning yuqori qismida, jadvalning o‘ng chetiga tekislangan holda ko‘rsatiladi.

DIQQAT! <CAPTION> tegi faqat ochilgan <TABLE> tegidan so‘ng darhol o‘rnatalishi mumkin.

Jadval qatorlari va katakchalarini qanday tahrirlash mumkin?

<TR> </TR>tegining mazmuni jadval qatori, to‘g‘riroq‘i, bir qatorda joylashgan kataklardan iborat. Jadvaldagi qatorlar soni <TR> teglar soni bilan belgilanadi. Tegni ochish majburiy, tegni yopish ixtiyoriy.

Atributlar (barchasi ixtiyoriy):

- ✓ **align, char, charoff, valign** – tekislash atributlari;
- ✓ **bgcolor** – qatorni o‘z ichiga olgan katakchalarining fon rangini o‘rnatuvchi atribut.

Standart ixtiyoriy atributlar:

- ✓ **id, class** – hujjat ichidagi identifikatorlar;
- ✓ **lang, dir** – matnning tili va yo‘nalishi haqidagi ma’lumotlar;
- ✓ **title** – elementning sarlavhasi (element mazmuni ustiga kursov olib kelganda brauzer tomonidan izoh sifatida ko‘rsatiladi);
- ✓ **style** – o‘rnatilgan uslub haqida ma’lumot;

<TH> va <TD> teglari jadval katakchalarini belgilaydi. Ushbu teglarning tarkibi bitta katakchadagi ma’lumotlardir.

- ✓ **rowspan** – bir ustun ichidagi qo‘shti satrlarni birlashtirish uchun mas’ul bo‘lgan atribut (ya’ni, ustundagi qo‘shti katakchalar). Qiymat sifatida natural sonlarni oladi;
- ✓ **colspan** – qo‘shti ustunlarni bir qatorda birlashtirish uchun mas’ul bo‘lgan atribut (ya’ni, qatordagi qo‘shti katakchalar). Uning qiymati natural sonlarni oladi(2.6-rasm);
- ✓ **width** – bu brauzerga tavsiya etilgan katakcha kengligini piksellarda bildiradi;
- ✓ **Height** – brauzerga tavsiya etilgan katak balandligini piksellarda bildiradigan kiruvchi atributdir;
- ✓ **bgcolor** – katakcha ichidagi fon rangini o‘rnatuvchi atribut. Rang nomini yoki uning RGB formatidagi mos qiymatini belgilaydi.

NATIJA			
nomer	Dushanba	Seshamba	Chorshanba
1	Matematika	Tarix	
2		Rus tili	KT

2.6-rasm. Rowspan va colspan atributlariga namuna

2.3. Htmlda multimedia obyektlari va gipermatnli hujjat

Sahifada rasmlar bilan ishslash

HTML5 ning imkoniyatlari Web-sahifaning jozibadorligini unga turli multimedia obyektlarini: tasvirlar, videokliplar, ovozli kliplar, appletlar va boshqa HTML hujjalarni kiritish orqali sezilarli darajada oshirishi mumkin. Multimedia obyektlarini kirituvchi universal HTML elementi OBJECT tegidir. U bilan birga HTML hujjatiga tasvirlarni kirituvchi tegidan foydalaniladi. OBJECT tegi umumiy bo‘lib, u har qanday mavjud obyektlarni (jumladan, tasvirlarni) va keljakda web-sahifada paydo bo‘ladigan obyektlarni o‘z ichiga olishi mumkin.

Tasvirlardan foydalanish har qanday hujjatni jozibali va tushunarli qilishning asosiy usuli ekanligiga shubha yo‘q. Bosma va Internetdagи deyarli hech bir nashr illyustratsiyalsiz to‘liq emas. Ular ishlab chiquvchiga hujjat mazmunini yaxshiroq va tushunarliroq yetkazish, uni yanada vizual qilish imkonini beradi. Dekorativ tasvirlar hujjatlar bilan ishlashda ko‘zni quvontiradi.

HTML hujjatlari illyustratsiyalar va fon tasvirlari sifatida joylashtirilgan tasvirlarni ajratib turadi. Ikkinchisi bilan ishslash Ish stoliga o‘rnatilganda Windowsda fon tasvirlari bilan ishslash bilan mutlaqo bir xil. Yagona farq shundaki, HTML hujjatlaridagi fon tasvirlari butun brauzer oynasini to‘ldiradi. Fon tasvirining

geometrik o‘lchamlari brauzer oynasidan kichikroq bo‘lsada, oynaning butun maydoni to‘ldiradi.

Fon tasvirlari sifatida tinch, yumshoq rangdagi kichik dekorativ rasmlarni (to‘qimalarni) ko‘zga tashlanmaydigan naqsh bilan ishlatalish tavsiya etiladi. Biroq, kattaroq tasvirlardan ham foydalanish mumkin. Faqat bu holatda, bu tasvir imkon qadar kam xotirani egallashi uchun uni iloji boricha optimallashtirish tavsiya etiladi.

HTMLda fon tasvirini o‘rnatish, yuqorida aytib o‘tilganidek, BODY tegining **BACKGROUND** atributi orqali amalga oshiriladi(2.7-rasm).

HTML

```
<!DOCTYPE html>
<html>
<head> <title>
Orqa fonga rasm joylash
</title> </head>
<body background="fon.jpg">
<h2 align=center> Orqa fonga rasm joylash bo'yicha namuna
</h2>
<hr color=darkgray size=10>
<p> HTML-da fon tasvirini o'rnatish, yuqorida aytib o'tilganidek, BODY tegining BACKGROUND atributi orqali amalga oshiriladi.
</p>
<i> Web texnologiyalar </i>
<hr color=darkgray size=10>
</body> </html>
```

NATIJA



2.7-rasm. Orqa fonga rasm joylash

Sahifalarda rasmlar bilan ishlash uchun **** tegidan foydalilanadi.

IMG tegi quyidagi atributlarga ega:

- ✓ **src** – tasvir joylashuvining URL manzilini (to‘liq yoki nisbiy) ko‘rsatuvchi zarur atribut(2.8-rasm);
- ✓ **Align** – gorizantal va vertikal tekislash;

Rasmni gorizantal joylashtirish:

- ✓ *left* – rasm hujjatning chap chetiga biriktirilgan. Matn o‘ng tarafdan tasvirni o‘rab oladi;
- ✓ *right* – rasm hujjatning o‘ng chetiga biriktiriladi va matn chap tarafdan tasvirni o‘rab oladi(2.9-rasm).

```

1 <!HTML>
2 <head>
3 <title>Linklar bilan ishlash</title>
4 </head>
5 <body>
6 <p>Rasm joylashtirish</p>
7 
8
9 </ul>
10 </html>

```

HTML

NATIJA

2.8-rasm. Sahifaga rasm joylashtirish

Rasmni vertikal joylashtirish:

- ✓ *top* – tepadan tekislash. Bu shuni anglatadiki, matn tasvirning oldida yuqori qismida;
- ✓ *middle* – matn tasvirning oldidan tasvirning o‘rtaligini darajasidan yoziladi;
- ✓ *button* – bu tekislash usuli bilan tasvirning pastki chegarasida matn joylashgan bo‘ladi(2.9-rasm).
- ✓ **width, height** – piksel yoki brauzer oynasining ko‘rinadigan maydonining foizi sifatida ko‘rsatilgan tasvirning geometrik o‘lchamlari;
- ✓ **hspace, vspace** – rasmdan mos ravishda gorizontal va vertikal chekinishlarni belgilaydi. Ushbu atributlarning qiymati piksellarda o‘rnataladi va ma’lum o‘lchamdagisi tasvir atrofida bo‘sh joy qoldirishga imkon beradi;
- ✓ **border** – tasvir atrofidagi hoshiyaning mavjudligi va kengligini boshqaruvchi atribut. Qiymat piksellarda ko‘rsatiladi va ramkaning kengligini aniqlaydi. Odatda, rasm atrofida chizilmaydi;

- ✓ ***alt*** – uning qiymati sifatida matn qatoriga ega bo‘lgan atribut, bu joriy tasvirning muqobil matnli ma’lumoti;
- ✓ ***longdesc*** – rasmga izohni o‘z ichiga olgan hujjat manzilini ko‘rsatish imkonini beruvchi atribut;
- ✓ ***usemap*** – tasvirni mijozning navigatsiya xaritasi bilan birgalikda ishlatish, uning nomi ushbu atributning qiymati sifatida ko‘rsatilgan.

Standart ixtiyoriy atributlar:

- ✓ ***id, class*** – hujjat ichidagi identifikatorlar;
- ✓ ***lang, dir*** – matnning tili va yo‘nalishi haqidagi ma’lumotlar;
- ✓ ***title*** – elementning sarlavhasi (element mazmuni ustiga kursov olib kelganda brauzer tomonidan izoh sifatida ko‘rsatiladi);
- ✓ ***style*** – o‘rnatilgan usslub haqida ma’lumot.

HTML

```
<!DOCTYPE html>
<html>
<head>
<title> rasm joylash </title>
</head>
<body>
    <h2>Rasmni gorizantal joylash</h2>
    <p> Rasm joylashuviga ko‘ra qanday joylashganini aniqlang.Rasm
joylashuviga ko‘ra gorizantalmi yoki vertikal? Rasm joylashuviga ko‘ra qanday
joylashganini aniqlang.Rasm joylashuviga ko‘ra gorizantalmi yoki vertikal?</p>
    <br> <hr />
    <p>Rasm joylashuviga ko‘ra qanday joylashganini aniqlang.Rasm
joylashuviga ko‘ra qanday joylashganini aniqlang.Rasm joylashuviga ko‘ra gorizantalmi
yoki vertikal?Rasm joylashuviga ko‘ra qanday joylashganini aniqlang</p>
    <br> <hr />
    <h2>Rasmni vertikal joylash</h2>
    <p>Rasm joylashuviga ko‘ra qanday joylashganini aniqlang.</p>
    <br> <hr />
    <p>Rasm joylashuviga ko‘ra qanday joylashganini
aniqlang.</p>
    <br> <hr />
    <p>TRasm joylashuviga ko‘ra qanday joylashganini
aniqlang.</p>
</body>
</html>
```

Rasmni gorizantal joylash



Rasm joylashuviga ko'ra qanday joylashganini aniqlang. Rasm joylashuviga ko'ra gorizantalmi yoki vertikal? Rasm joylashuviga ko'ra qanday joylashganini aniqlang. Rasm joylashuviga ko'ra gorizantalmi yoki vertikal?



Rasm joylashuviga ko'ra qanday joylashganini aniqlang. Rasm joylashuviga ko'ra qanday joylashganini aniqlang. Rasm joylashuviga ko'ra gorizantalmi yoki vertikal? Rasm joylashuviga ko'ra qanday joylashganini aniqlang.

Rasmni vertikal joylash



Rasm joylashuviga ko'ra qanday joylashganini aniqlang.



Rasm joylashuviga ko'ra qanday joylashganini aniqlang.



TRasm joylashuviga ko'ra qanday joylashganini aniqlang.

2.9-rasm. Rasmni joylashtirish xususiyatlari

Sahifaga audio va video obyektlar joylash

HTML sahifasida audio va video obyektlarni joylashtirish uchun mos ravishda AUDIO va VIDEO teglaridan foydalaniladi. Manba faylga yo'l o'rnatilgan SOURCE tegi yoki SRC atributi orqali belgilanadi.

<AUDIO> tegi

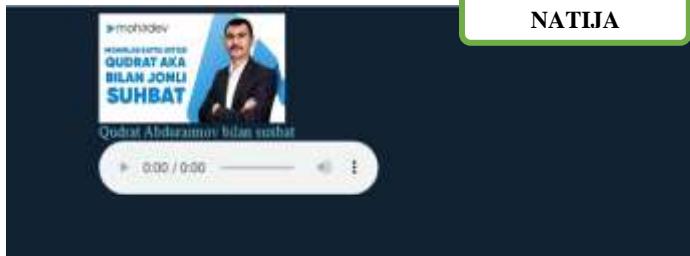
web-sahifaga audio yozuv qo'shish uchun foydalaniladi. Audio faylni tinglash sozlamalari quyidagi atributlar yordamida o'rnatiladi(2.10- rasm):

- ✓ *autoplay* – audio avtomatik ijro;
- ✓ *controls* – audioyozuvga boshqaruv panelini qo'shish imkonini beradi;
- ✓ *loop* – audiofaylni ijro etish tugagandan so'ng uni takroriy tinglash;
- ✓ *preload* – audio faylni sahifa bilan birga yuklaydi;
- ✓ *src* – audio fayl manbasiga yo'lni belgilaydi.

HTML

```
<!DOCTYPE html>
<html>
<head>
<title> Добавление аудио </title>
</head>
<body leftmargin=100 rightmargin=100 text=#78bed0
bgcolor=#112232>
<IMG src="rasm.jpg" align=middle width=200>
<br>
Qudrat Abduraimov bilan suxbat
<br>
<audio src="track01.mp3" controls="controls" autoplay="autoplay">
</audio>
</body>
```

NATIJA



2.10-rasm. Sahifaga video joylash

<VIDEO > tegi

Web-sahifaga video qo'shish uchun ishlatiladi. Video fayli sozlamalari quyidagi atributlar yordamida o'rnatiladi:

- ✓ **autoplay** – videoni avtomatik ijro etish;
- ✓ **control** – videongizga boshqaruv paneli qo'shish imkonini beradi;
- ✓ **loop** – videofaylni ko'rish tugagandan so'ng uni takroriy ijro etish;
- ✓ **preload** – video faylni sahifa bilan birga yuklaydi;
- ✓ **src** – videofayl manbasiga yo'lni belgilaydi;
- ✓ **height, width** – videofaylni ko'rish maydoni parametrlarini o'rnatish.

HTML

```
<html>
<head> <title> Video joylash </title> </head>
<body leftmargin=100 rightmargin=100 text=#78bed0 <br>
Kompyuterning asosiy qurilmalari
<br>
<video width=400 height=300 src="video.mp4" controls="controls"
autoplay="autoplay">
</video>
</body></html>
```



NATIJA

2.11-rasm. Sahifaga video joylash

SOURCE tegidan foydalanish AUDIO va VIDEO teglari uchun SRC atributidan foydalanishga o‘xshaydi. Bu teg web-sahifaga yuklangan manba faylga yo‘lni belgilash imkonini beradi.

Boshqa multimedia obyektlarini qanday kiritish mumkin?

Ushbu bo‘limning boshida Web-sahifada obyektlarni joylashtirish OBJECT tegi orqali amalga oshirilishi aytilgan bo‘lsada, aslida bu maqsadda ikkita HTML tegidan foydalanish mumkin: OBJECT va EMBED.

NAZORAT SAVOLLARI

- 2.1.Ro‘yxatlar qanday e’lon qilinadi va uning qanday turlari mavjud?
- 2.2.Jadval va jadval bilan ishlovchi teglar va ularning vazifasini ayting?
- 2.3.Jadvalning ustun va qatorlaridagi yacheykalar qaysi atributlar yordamida birlashtiriladi?
- 2.4.Sahifaga rasm joylash uchun qanday teglardan foydalaniladi?.
- 2.5. tegi atributlari va vazifalarini ayting?
- 2.6.Video va audio bilan ishlovchi teglar va ularning vazifasini ayting?

III-BOB. HTML TILIDA GIPERMATN VA FORMALAR

JOYLASHTIRISH TEGLARI. FREYMLAR HOSIL QILISH VA UALAR

BILAN ISHLASH

3.1. Htmlda gipermatn bilan ishslash

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

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

Xavolalar [<a>](#) tegi bilan yaratiladi. Foydalanuvchilar ochiluvchi [<a>](#) tegi va yopiluvchi [](#) tegi ichida yozilgan matnni ko‘rishadi va matn ustiga bosilganda href atributi ichida yozilgan sahifaga o‘tiladi.

Gipermurojaat (Xavola (ssilka)) – bu maxsus HTML tegi bo‘lib, unda parametr sifatida zaruriy hujjat manzili saqlanadi. Gipermurojaat faollashganda (zamonaviy grafik brauzerlarda bu ish sichqonchaning chap tugmasi orqali amalga oshiriladi) zaruriy Web-serverga HTTP talab (kerakli hujjat nomi bilan) yuboriladi.

Gipermurojaat (giper xavola (ssilka)) ikki qismdan iborat. Uning birinchi qismini xavola (ssilka) ko‘rsatkichi (anchor), brauzerga instruksiya beradigan ikkinchi qismi esa, xavola (ssilka)ning manzil qismi (URL – manzil) deyiladi. Agar foydalanuvchi sichqoncha kursorini Xavola (ssilka) ko‘rsatkichiga olib kelsa, u holda u ko‘rsatuvchi belgiga aylanadi, (giperaloqaning birinchi belgisi).



Matn ko‘rsatkichlari bir yoki bir necha so‘zlardan iborat bo‘lib, so‘z ostiga chizish bilan ajralib turadi. (3.1-rasm).

```

<!DOCTYPE html>
<html>
<head>
<title> Video joylash </title>
</head>
<body leftmargin=100 rightmargin=100 text=darkblue>
<p>
<ul>
<li><a href="index.html">Home</a></li>
<li><a href="about-us.html">About</a></li>
<li><a href="movies.html">Movies</a></li>
<li><a href="contact.html">Contact</a></li>
</ul> </p>
</body> </html>

```

HTML

- [Home](#)
- [About](#)
- [Movies](#)
- [Contact](#)

NATIJA

3.1-rasm. Xavolalardan foydalanish

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

Xavola (ssilka) ko‘rsatkichi so‘z, so‘zlar guruhi yoki tasvir bo‘lishi mumkin. Ko‘rsatkichlarning matnli va grafikli turlari mavjud.

Xavola (ssilka) sifatida grafik tasvirlardan ham foydalanish mumkin. Grafik xavola (ssilka)lar harakat qilish prinsipi bo‘yicha matn xavola (ssilka)laridan hech qanday farqlanmaydi. Ular tagiga chizilmaydi va ranglari ajralib turmaydi (3.2-rasm).

```

<html>
<head>
<title> Video joylash </title>
</head>
<body leftmargin=100 rightmargin=100 text=darkblue>
<p>
<a href="video.html"><IMG SRC = "video.jpg" width="100" height="70"></a>
</p>
</body>
</html>

```

HTML



3.2-rasm. Rasmlı xavola yaratish

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

Xavolalar [tegi](#) bilan yaratiladi. Foydalanuvchilar ochiluvchi [tegi](#) va yopiluvchi [tegi](#) ichida yozilgan matnni ko‘rishiadi va matn ustiga bosilganda href atributi ichida yozilgan sahifaga o‘tiladi. Xavolalar yangi oynada ochilishi uchun **target=“_blank”** xususiyatidan foydalaniadi. Shuningdek, bitta sahifa ichida ham murojatlar o‘rnatish mumkin. Buning uchun class lardan foydalaniadi ya’ni sahifa class lar yordamida qismlarga bo‘linadi.

```

1 <html>
2 <head>
3   <title></title>
4 </head>
5 <body>
6   <a href="https://www.google.com">
7     Googledan ma'lumot qidirish</a>
8 </body>
9 </html>
```

HTML

Boshqa web-saytga havola qilish uchun, href atributining qiymati URL sifatida tanilgan saytning to‘liq web-manzili bo‘ladi(3.3- rasm).

[Googledan ma'lumot qidirish](#)

3.3-rasm. Boshqa web-saytga o‘tish

3.2. Shakllar va ularning vazifalari

HTML hujjalardagi shakllar(formalar) serverga masofaviy foydalanuvchidan ma'lumotlarni yuborish uchun ishlataladi. Shunday qilib, mijoz-serverning eng oddiy dialogi tashkil etiladi, masalan: foydalanuvchini serverda

onlayn ro‘yxatdan o‘tkazish (anketa to‘ldirish), onlayn do‘konda xarid qilishni amalga oshirish.

Shakllar foydalanuvchi ma’lumotni qo‘lda kiritadigan yoki taklif qilingan qiymatlar ro‘yxatidan tanlaydigan bir yoki bir nechta kiritish maydonlaridan iborat. Shaklni to‘ldirgandan so‘ng, u serverga yuboriladi va server skriptlari tomonidan qayta ishlanadi. Skriptlar boshqacha ko‘rinishda ham bo‘lishi mumkin. Bitta sahifa bir nechta shakllarni o‘z ichiga olishi mumkin. Foydalanuvchi shakllar bilan boshqaruv elementlari yordamida o‘zaro aloqada bo‘ladi.

Boshqarish elementlari

HTML hujjatidagi forma barcha boshqaruv teglarini belgilaydigan FORM konteyner tegi tomonidan amalga oshiriladi. Agar boshqaruv teglari FORM tegining tarkibidan tashqarida ko‘rsatilgan bo‘lsa, u holda ular forma yaratmaydi. Boshqaruv tegi Web-sahifada foydalanuvchi interfeysi qurish uchun, unga turli tugmalar, belgilash katakchalari, kiritish maydonlarini qo‘sish uchun ishlatiladi. HTML hujjati doirasida, unga kiritilgan mijoz skriptlari yordamida individual ravishda qayta ishlanadi. Yoki ular umuman qayta ishlanmasligi mumkin. Masalan, TEXTAREA boshqaruv tegi ko‘pincha hujjat ichida ikkinchi darajali rol o‘ynaydigan katta matnni ko‘rsatish uchun aylantirish paneli oynasini yaratish uchun ishlatiladi. Odatda, litsenziya shartnomalari matnlari, katta sharhlar matnlari yoki ushbu web-resursdan foydalanish qoidalari shunday ko‘rsatiladi.

HTML 5 boshqaruvning quyidagi turlarini belgilaydi:

- ✓ **Tugmalar** – Button va Input teglari yordamida beriladi. Farqlash:
 - yuborish tugmalari(кнопки отправки) – ularni bosganingizda ular formani serverga jo‘natadi;
 - qayta o‘rnatish tugmalari(кнопки сброса) – bosilganda dastlabki qiymatlar boshqaruv elementlariga o‘rnatiladi;
 - boshqa tugmalar(прочие кнопки) – ular bosilganda belgilangan buyruq asosida xarakatlanadi.
- ✓ **radio tugmalar** (radio tugmasi bo‘lgan tugmalar) – INPUT tegi bilan o‘rnatiladi. Agar bir nechta radio tugmalar bir xil nomdagi boshqaruv

elementlari tomonidan o‘rnatilgan bo‘lsa, ular bir-birini istisno qiladi. Bu shuni anglatadiki, agar ulardan biri “yoqilgan” holatiga o‘rnatilgan bo‘lsa, qolganlari avtomatik ravishda “o‘chirilgan” holatiga o‘rnatiladi. Bu ulardan foydalanishning afzalligidir;

- ✓ bayroqlar – INPUT elementi tomonidan o‘rnatiladi va “yoqish / o‘chirish” holatlari mavjud, lekin radio tugmalaridan farqli o‘laroq, “yoqilgan” bayroqlar bir xil nomga ega bo‘lishi mumkin. Eslatma: nomlar name atributi bilan beriladi;
- ✓ menyu – SELECT, OPTGROUP va OPTION teglari yordamida amalga oshiriladi. Menyular foydalanuvchiga mumkin bo‘lgan variantlar ro‘yxatini taqdim etadi;
- ✓ matn kiritish – bitta satr kiritilsa INPUT teglari va bir nechta satr kiritilsa TEXTAREA teglari orqali amalga oshiriladi. Ikkala holatda ham kiritilgan matn boshqaruvning joriy qiymatiga aylanadi;
- ✓ fayl tanlash – INPUT HTML tegi orqali amalga oshirilgan tanlangan fayllarni forma bilan birga yuborish imkonini beradi;
- ✓ yashirin boshqaruv elementlari – INPUT boshqaruv tegi tomonidan yaratiladi. HTMLda formalar, foydalanuvchilar tomonidan ma’lumotlarni kiritish va kiritilgan ma’lumotlardan keyinchalik foydalanish maqsadida tashkil qilingan. Formalar – matnli oynalar, tugmalar, bayroqchalar, ulagich(pereklyuchatel)lar va menyular kabi boshqariladigan elementlardan tashkil topishi mumkin. Formalar <form> va </form> teglari orasida joylashadi.

<form>...</form> – web sahifaga forma joylashtirish, forma foydalanuvchi va web server bilan aloqani ta’minlab beradi. Agar forma elementlari (button, checkbox, radiobutton,...) kiritilmasa, bu teg brauzerga hech nima chiqarmaydi.

Umumiy ko‘rinishi:

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

FORM teg atributlari ro‘yxati:

- ✓ **accept-charset** – serverga jo‘natilayotgan axborot kodirovkasini o‘rnatish;
<form accept-charset=“windows-1251”>...</form>;

✓ ***action*** – forma orqali jo‘natilayotgan ma’lumotlar qaysi sahifa orqali qayta ishlanishi kerakligi beriladi, biror manzil ko‘rsatiladi.

<form action=“index.php”>...</form> formadagi barcha axborotlar “index.php” sahifasiga olib o‘tilib, u yerda tahrirlanadi.

Action – talab qilinadigan atribut. Uning qiymati so‘ralgan CGI dasturining URL manzili bo‘lib, u formadagi ma’lumotlarni qayta ishlaydi. Formani elektron pochta orqali yuboradigan mailto:URL yozuvidan foydalanish maqbuldir. Agar harakat atributi hali ham ko‘rsatilmagan bo‘lsa, forma tarkibi web-sahifa yuklangan URL manziliga yuboriladi;

✓ ***method*** – forma ma’lumotlarini brauzerdan serverga yuborish uchun ishlatiladigan HTTP usulini belgilaydi. Metod atributi ikkita qiymatni qabul qilishi mumkin:

✓ *get* – HTTP “get” usulidan foydalanilganda forma ma’lumotlari HTTP sarlavhasining birinchi qatoriga yuboriladi. Bunday holda, ular Action atributi tomonidan belgilangan URL manziliga qo‘shiladi va u bilan birga yuboriladi. Biriktirish “?” belgisi bilan amalga oshiriladi, undan keyin shakl ma’lumotlari “boshqaruv nomi = qiymat” juftlari ketma-ketligi sifatida belgi bilan ajratiladi. Ma’lumotni qabul qilishda server uni URL manzilidan ajratadi va uni CGI dasturi tomonidan qo‘llaniladigan Query-string muhit o‘zgaruvchisiga tayinlaydi. Odatda Get usuli ishlatiladi;

✓ *post* – HTTP “post” usulidan foydalanganda, forma ma’lumotlar to‘plami serverga HTTP matnida yuboriladi. Agar forma serverda hech qanday o‘zgarishlarga olib kelmasa, “get” usuli tavsiya etiladi (o‘zgartiradi). Ma’lumotlar bazasi, xizmat abonentlari reestriga qo‘shiladi va hokazo). “Get” usulining idealligi ma’lumotlar bazasini qidirishdir. Aks holda “post” usulidan foydalanish tavsiya etiladi;

✓ *enctype* – formani yuborishda kodlash formatini aniqlash uchun foydalaniladigan formaning kontent turini belgilaydi. HTML enctype atributi uchun uchta mumkin bo‘lgan qiymatni belgilaydi: application/x-www-form-urlencoded (standart), multipart/form-data, text/ plain;

- ✓ *id, class* – hujjat ichidagi identifikatorlar;
- ✓ *lang, dir* – matnning tili va yo‘nalishi haqidagi ma’lumotlar;
- ✓ *title* – elementning sarlavhasi (element mazmuni ustiga kursov olib kelganda brauzer tomonidan izoh sifatida ko‘rsatiladi);
- ✓ *style* – o‘rnatilgan uslub haqida ma’lumot;
- ✓ onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup – onsubmit, onreset - ichki hodisalar.

INPUT tegidan foydalanish

INPUT tegi eng ko‘p ishlatiladigan element hisoblanadi, chunki uning yordamida formaning asosiy vazifalari amalga oshiriladi. U sizga forma ichida matn, fayl nomi, parol va boshqalar uchun kiritish maydonlarini yaratish imkonini beradi.

INPUT tegini ochish majburiy, teg yopilmaydi.

INPUT elementining atributlari ro‘yxati.

- ✓ *name* – bu atributning qiymati ushbu INPUT elementi tomonidan tuzilgan boshqaruv elementining nomidir. Name atributini ko‘rsatish talab qilinadi, chunki usiz boshqaruv elementining qiymatini yo‘naltirish mumkin emas;
- ✓ *type* – boshqaruv elementining turini belgilaydigan majburiy atribut. HTMLda type atributiga tegishli qiymatga ega Input elementi tomonidan berilgan boshqaruvning quyidagi turlari mavjud:
 - ✓ *text* – satr kiritish uchun ishlatiladi;
 - ✓ *password* – “matn” qiymatiga o‘xshash, farqi kiritilgan matn yulduzchalar sifatida ko‘rsatiladi ya’ni parol;
 - ✓ *Checkbox* – “on\off” (on\off) bayroqchasini yaratadi. Odatda u “off” holatida bo‘ladi. Xuddi shu nomli, ya’ni nom atributining bir xil qiymatiga ega bo‘lgan bir nechta bayroqlar(checkbox) guruhidan foydalanishga ruxsat beriladi;
 - ✓ *radio* – radio tugma hosil qiladi. U bir xil nomdagagi o‘xshash elementlar guruhining bir qismi sifatida ishlatiladi, ulardan faqat bittasini tanlash

mumkin (“on” holatiga o‘rnataladi) Brauzerlar ularni yumaloq tugma sifatida ko‘rsatadi;

- ✓ *submit* – yuborish tugmachasini yaratadi. Ushbu tugmani bosish formaning butun tarkibini serverga yuboradi. Odatiy bo‘lib, u “Отправить” etiketli to‘rtburchaklar tugma sifatida ko‘rsatiladi;
- ✓ *image* – grafik tasvir sifatida yuborish tugmachasini yaratadi;
- ✓ *reset* – forma qiymatlarini dastlabki holatga qaytaradi;
- ✓ *file* – fayl tanlash boshqaruvini yaratadi. Faylni tanlash klaviaturadan fayl nomini kiritish yoki har doim fayl nomini kiritish maydoni yonida joylashgan “Browse” tugmasi bilan ochiladigan muloqot oynasidan amalga oshiriladi. Shu tarzda tanlangan fayl forma maydoniga biriktiriladi va u bilan birga serverga yuboriladi. Biriktirilgan fayllarni jo‘natish uchun formani o‘rnatishda Enctype = “multipart/ form-data” va Method = post atributlarini belgilashingiz kerak. Aks holda serverga fayl emas, faqat uning nomi yuboriladi. Fayl tanlash vositalariga kelsak, Maxlength va Size atributlari qo‘llanilishi mumkin;
- ✓ *hidden* – brauzer tomonidan ko‘rsatilmaydigan yashirin elementni yaratadi. Ko‘rsatilmaydigan boshqaruv elementi maydonidagi ma’lumotlar forma uztalganda serverga yuboriladi va uni foydalanuvchi ham, brauzer ham o‘zgartira olmaydi.
- ✓ *size* – boshqaruv elementining boshlang‘ich kengligini o‘rnatadi. Qiymatlar piksellarda ko‘rsatilgan, matn va parolni boshqarish vositalaridan tashqari. Ular uchun size atributi satrdagi belgilar sonini belgilaydi;
- ✓ *align* – tekislash;
- ✓ *id, class* – hujjat ichidagi identifikatorlar;
- ✓ *lang, dir* – matnning tili va yo‘nalishi haqidagi ma’lumotlar;
- ✓ *title* – elementning sarlavhasi (element mazmuni ustiga kursov olib kelganda brauzer tomonidan izoh sifatida ko‘rsatiladi);
- ✓ *alt* – berilgan boshqaruv elementi uchun qisqacha tushuntirish ma’lumotlari ushbu atributning qiymati sifatida ko‘rsatiladi;

- ✓ ***usemap*** – mijozning navigatsiya xaritasini ulash uchun ishlataladi. Qiymat - bu xaritaning nomi;
 - ✓ ***readonly*** – mantiqiy atribut, uning mavjudligi foydalanuvchiga maydonni o‘zgartirishga to‘sqinlik qiladi.
- ***Matn turidagi axborotlarni kiritish uchun INPUT tegining type atributiga text qiymati beriladi. (type= “text”).***

Odatda foydalanuvchi ismi, familiyasiga o‘xshash ma’lumotlarni kiritishi uchun ishlataladi.

INPUT(type=text) tegining atributlari:

- ***maxlength*** = n – kiritish maydonida ruxsat etilgan satr uchun maksimal belgilar sonini belgilaydi. Odatda, belgilar soni cheklanmagan;
- ***size*** = n – qatorga kiritish uchun ko‘rsatilgan belgilar sonini bildiradi (belgilar soni maydon kengligini belgilaydi);
- ***value***= text – matn maydonining boshlang‘ich qiymatini o‘rnatadi(3.4-rasm);
- ***Name*** – foydalanuvchilar formaga ma’lumot kiritganda, server har bir ma’lumot bo‘lagi qaysi boshqaruvi elementiga kiritilganligini bilishi kerak. (Masalan, login formasida server foydalanuvchi nomi sifatida nima kiritilganligini va parol sifatida nima berilganligini bilishi kerak.) Shuning uchun har bir forma boshqaruvi name atributini talab qiladi. Ushbu atributning qiymati forma boshqaruvini aniqlaydi va ular serverga kiritgan ma’lumotlar bilan birga yuboriladi.

HTML

```
<!DOCTYPE html>
<html>
<head>
<title> Video joylash </title>
</head>
<body leftmargin=100 rightmargin=100 text=darkblue>
<form action="http://www.example.com/login.php">
<p>Username:<br/>
<input type="text" name="username" size="15" maxlength="30" />
</p>
</form> </body> </html>
```

NATIJA

Username:

3.4-rasm. Input tegining type=text atributi

Agar maydonga **elektron pochta** haqidagi ma'lumotni kiritmoqchi bo'lsangiz **name= "email"** deb yozish tavsiya etiladi

➤ **Parolni kiritish uchun INPUT tegining type atributiga password qiymati yoziladi(type= "password").**

Bu maxsus ma'lumot kiritish maydoni bo'lib, foydalanuvchi tomonidan kiritilgan ma'lumotlar ekranga o'qilmaydigan holatda chiqariladi. Bu text turiga o'xshab ketadi. Uni bo'sh qoldirish mumkin yoki "*" belgisi orqali biror parolni yozish mumkin:

INPUT(type=password) tegining atributlari:

- **maxlength** = n – kiritish maydonida ruxsat etilgan satr uchun maksimal belgilar sonini belgilaydi. Odatda, belgilar soni cheklanmagan;
- **size** = n – qatorga kiritishuchun ko'rsatilgan belgilar sonini bildiradi (belgilar soni maydon kengligini belgilaydi);
- **value**= text – matn maydonining boshlang'ich qiymatini o'rnatadi(3.5-rasm).

HTML

```
<!DOCTYPE html>
<html>
<head>
<title> Video joylash </title>
</head>
<body leftmargin=100 rightmargin=100 text=darkblue>
<form>
<p>Username:
<input type="text" name="username" size="15" maxlength="30" />
</p>
<p>Password:
<input type="password" name="password" size="15" maxlength="30" />
</p>
</form>
</body>
```

Username:	<input type="text"/>	NATIJA
Password:	<input type="password"/>	

3.5-rasm. INPUT tegining(type=password) atributlari

- ***Checkbox turidagi elementlarni boshqarish uchun INPUT tegining type atributi checkbox qiyatiga teng bo‘ladi(type= “checkbox”).***

Foydalanuvchi tavsiya etilgan variantlardan bir nechtasini o‘rnatishi yoki bekor qilishi mumkin. Oldindan birortasini belgilab qo‘yish imkoniyati ham bor.

Atributlari:

- **value** = text – bayroqchalar uchun zarur atribut. Formani to‘ldirishda belgilash katakchasi “yoqilgan” holatda bo‘lsa, belgilash katakchasi ma’lumotlar serverga yuboriladi. Agar belgilash katakchasi “off” holatida bo‘lsa, belgilash katakchasi ma’lumotlar serverga yuborilmaydi;
- **checked** – bu ixtiyoriy mantiqiy atribut bo‘lib, birlamchi belgilash katakchasini “yoqish”ga qo‘yadi. Agar u belgilanmagan bo‘lsa, birlamchi belgilash katakchasi “o‘chirilgan” ga o‘rnataladi;
- **name** – bu atributning qiymati ushbu INPUT elementi tomonidan tuzilgan boshqaruvi elementining nomidir. Name atributini ko‘rsatish talab qilinadi, chunki usiz boshqaruvi elementining qiymatini yo‘naltirish mumkin emas. Har bir checkbox forma ichida bitta nomga (name atributiga) ega bo‘lishi kerak(3.6-rasm).

<pre><!DOCTYPE html> <html> <head> <title> Video joylash </title> </head> <body leftmargin=100 rightmargin=100 text=darkblue> <form> <p>Please select your favorite music service(s):
 <input type="checkbox" name="service" value="itunes" checked="checked" /> iTunes <input type="checkbox" name="service" value="lastfm" /> Last.fm <input type="checkbox" name="service" value="spotify" /> Spotify </p> </form> </body> </html></pre>	HTML
---	------

Please select your favorite music service(s):

iTunes Last.fm Spotify

3.6-rasm. INPUT tegining(type= Checkbox) atributlari

➤ **Radio button turidagi elementlarni boshqarish uchun INPUT tegining type atributi radio qiymatiga teng bo'ladi(type= “radio”).**

Foydalanuvchi bir nechta variantlardan faqat bittasini tanlashi mumkin bo'ladi. Bu yerdagi ma'lumotlardan faqat bittasini tanlash mumkin. Har qaysi radio button **value** atributiga ega bo'lishi shart. Oldindan biror variantni belgilab qo'yish ham mumkin, bu holatda foydalanuvchi ulardan birortasini belgilashni unutgan holda kelib chiqishi mumkin bo'lgan xatolikning oldini olishi mumkin:

Atributlari:

- **value** = text – talab qilinadigan atribut. Formani to'ldirishda radio button “yoqilgan” holatda bo'lsa, qiymat serverga yuboriladi;
- **checked** – standart radio tugmani “yoqish” ga o'rnatadigan ixtiyoriy mantiqiy atribut. Agar belgilanmagan bo'lsa, birlamchi radio tugmasi “off” ga o'rnatiladi;
- **name** – bu atributning qiymati ushbu INPUT elementi tomonidan tuzilgan boshqaruvi elementining nomidir. Har bir radio tugma forma ichida bitta nomga (name atributiga) ega bo'lishi kerak(3.7-rasm).

```
<!DOCTYPE html>
<html>
<head>
<title> Video joylash </title>
</head>
<body leftmargin=100 rightmargin=100 text=darkblue>
<form>
<p>Please select your favorite genre:
<br />
<input type="radio" name="genre" value="rock"
checked="checked" /> Rock
<input type="radio" name="genre" value="pop" />
Pop
<input type="radio" name="genre" value="jazz" />
Jazz</p>
</p>
</form> </body> </html>
```

NATIJA

Please select your favorite genre:

Rock Pop Jazz

3.7-rasm. INPUT tegining(type= radio) atributlari

➤ **Formada kiritilgan ma'lumotlarni yuborish uchun INPUT tegining type atributi submit qiymatiga teng bo'ladi (type= "submit").**

Atributlari:

- **Value**=matn(text) – Yuborish(submit) tugmasi uchun maxsus nomni belgilaydi. Ushbu atribut tugmachadagi yozuvni o'zgartirishga imkon beradi. Yuborish(submmit) tugmalarining nomi (atribut name) yozilmasligi mumkin. Bunday holda, yuborish tugmasining qiymati serverga yuborilmaydi. Agar unga name va value atributlari o'rnatilgan bo'lsa, u holda tugma nomi yuborilgan qiymat sifatida ishlataladi. Formani yuborish uchun siz forma ichida yuborish(submit) tugmasini belgilashingiz kerak. Ya'ni, yuqoridaq misollarda ko'rsatilgan barcha formalar serverga yuborilishi va u yerda qayta ishlanishi uchun yuborish(submit) tugmachaсини o'rnatishingiz kerak. Aks holda, ular shunchaki interfeysning predmeti bo'lib qoladi;
- **Name** – name atributidan foydalanishi mumkin, lekin u bo'lishi shart emas(3.8-rasm).

HTML

```
<body leftmargin=100 rightmargin=100 text=darkblue>
<form>

```

NATIJA

Please select your favorite genre:

Rock Pop Jazz

Subscribe to our email list:

3.8-rasm. INPUT tegining(type= submit) atributi

➤ *Yuborish tugmalarini Grafik ko‘rinishda hosil qilish uchun Input tegining type atributiga image qiymati beriladi(type=“image”).*

Atributlari:

- **src** – rasmning URL manzilini belgilaydi;
- **align** – tasvirni tekislash turini o‘rnatadi;
- **height** – tugmaning bo‘yini belgilaydi;
- **weight** – tugmaning enini belgilaydi(5.6-rasm).

HTML

```
<form>
<p>Please select your favorite genre:
<br />
<input type="radio" name="genre" value="rock"
checked="checked" /> Rock
<input type="radio" name="genre" value="pop" />
Pop
<input type="radio" name="genre" value="jazz" />
Jazz
</p>
</p>
<p>Subscribe to our email list:</p>
<input type="text" name="email" />
<br>
<input type="image" src="submit.png"
width="130" height="40" />
</form>
```

NATIJA

Please select your favorite genre:

Rock Pop Jazz

Subscribe to our email list:

→ Отправить

3.9-rasm. Grafikli tugma

- *reset – forma qiymatlarini dastlabki holatga qaytarish uchun ishlataladi(type= “reset”).*

Tugmalardan foydalanishda type= “reset” qo‘llanilsa, formadagi hamma o‘zgarishlarni tozalash imkoniyati bo‘ladi. Ya’ni tugmani bosish natijasida forma avvalgi holatga qaytadi.

- *Fayl kiritish uchun Input tegining type atributiga file qiymati beriladi(type= “file ”).*

Fayl kiritish artibuti. Agar foydalanuvchilarga faylni (masalan, rasm, video, mp3 yoki PDF) yuklashiga ruxsat bermoqchi bo‘lsangiz, fayl kiritish maydonidan foydalanishingiz kerak bo‘ladi. type=“file” Ushbu turdagи kiritish matn kiritishga o‘xshab ko‘ringan tugmachani hosil qiladi.

Foydalanuvchi ko‘rib chiqish(browse) tugmasini bosganida, kompyuterdan web-saytga yuklanadigan faylni tanlash imkonini beruvchi oyna ochiladi(3.10-rasm).

Foydalanuvchilarga fayllarni yuklashga ruxsat berganingizda, <form> elementidagi method atributi post qiymatiga ega bo‘lishi kerak. (Siz HTTP get usuli yordamida fayllarni yubora olmaysiz).

```
<form method="post">
<p>Upload your song in MP3 format:</p>
<input type="file" name="user-song" /><br />
<input type="submit" value="Upload" />
</form>
```

HTML

NATIJA

Upload your song in MP3 format:

Выберите файл	Файл не выбран
Upload	

3.10-rasm. File kiritish atributi

TEXTAREA elementi

Foydalanuvchi ko‘p satrli ma’lumot kiritishi uchun <TEXTAREA> tegidan foydalaniladi(oddiy matn 3.11-rasm).



3.11-rasm. Textarea elementidan foydalanish

<TEXTAREA> ichida ishlataladigan atributlar (Teg </TEXTAREA>atributisiz ham foydalanish mumkin):

- ***name*** – ma'lumot kirituvchi ob'ekt (maydon) ni nomi.
- ***rows*** – ma'lumot kirituvchi ob'ektni balandligi (satr ko'rinishida belgilaydi).
- ***cols*** – ma'lumot kirituvchi ob'ektni kengligi (ustun ko'rinishida belgilaydi).

Agar siz ma'lumot kirituvchi maydon ichida o'zi bilan birga ma'lumot chiqishini xohlasangiz, unda <TEXTAREA> va </TEXTAREA> tegi orasiga matnni joylashtiring.

SELECT tegidan foydalanish

HTML hujjat formalarida tavsiya etilgan qiymatlar ro'yxatidan tanlash ikki yo'l bilan amalga oshirilishi mumkin: radio tugmalari va tasdiqlash qutilari(bayroqchalar) yordamida yoki SELECT tegi yordamida. Ikkala usulning ham afzalliklari va kamchiliklari bor: radio tugmalari va tasdiqlash qutilari tanlovlar kam bo'lganda tanlov qilish uchun ishlataladi. Ko'p sonli radio tugmalar va tasdiqlash qutilari formani foydalanish uchun noqulay qiladi. Ko'p tanlovli qiymatlar ro'yxatidan tanlovnini amalga oshirish uchun SELECT tegidan foydalaniladi. Uning asosiy afzalligi ham kamchiligi ham – ixchamlikdir, chunki SELECT tegi tasdiqlash qutilari va radio tugmalari yordamida yaratilgan ro'yxatga qaraganda kamroq jozibador ko'rindi. Belgilash katakchalari tartibdan qat'i nazar, ro'yxat elementlari orasida bir nechta tanlash imkonini beradi.

Boshlash va tugatish SELECT teglari talab qilinadi. Bu teg quyidagi atributlarga ega:

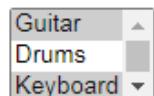
- **name** – o‘z ichiga olgan elementga nom beradi;
- **size** – bir vaqtning o‘zida ko‘rinadigan tanlash elementlari sonini belgilaydi. U qiymat sifatida natural sonlarni oladi. Agar o‘lcham atributi 1 ga o‘rnatalgan bo‘lsa, tavsiya etilgan tanlov qiymatlari ro‘yxati ochiladigan menyu sifatida ko‘rsatiladi. Agar o‘lcham atributining qiymati tanlov ro‘yxati elementlari sonidan kam bo‘lsa, u ko‘rsatilganda vertikal aylantirish paneli ishlataladi;
- **multiple** – mantiqiy atribut, uning belgisi bir vaqtning o‘zida bir nechta taklif qilingan variantlarni bir nechta tanlash imkonini beradi. Agar ushbu atribut mavjud bo‘lmasa, odatda, berilgan qiymatlar ro‘yxatida bir nechta tanlash mumkin emas(3.12-rasm);

HTML

```
<form action="http://www.example.com/profile.php">
<p>Quyidagi asboblardan birini chalasizmi?
(kompyuterda ctrlni bosib turgan holatda bir nechta variantni
tanlashingiz mumkin.)</p>
<select name="instruments" size="3"
multiple="multiple">
<option value="guitar" selected="selected">
Guitar</option>
<option value="drums">Drums</option>
<option value="keyboard"
selected="selected">Keyboard</option>
<option value="bass">Bass</option>
</select>
</form>
```

NATIJA

Quyidagi asboblardan birini chalasizmi? (kompyuterda ctrlni bosib turgan holatda
bir nechta variantni tanlashingiz mumkin.)



3.12-rasm. Bir nechta tanlash qutisi

- **tabindex** – “TAB” tugmachasi orqali o‘tish ketma-ketligidagi seriya raqami;
- **onfocus, onblur, onchange** – ichki hodisalar.

SELECT tegi menyuni yaratadi, unda har bir tanlov SELECT tegining tarkibida ishlatiladigan OPTION tegi bilan belgilanadi (xuddi LI tegi ro‘yxat elementi bo‘lgani kabi). OPTION teglarining tartibi tanlangan ro‘yxatdagi elementlarning ko‘rsatilish tartibini belgilaydi.

➤ ***OPTION tegini ochish majburiy yopish tegi ixtiyoriy.***

Atributlar ro‘yxati:

- ***value*** – agar o‘z ichiga olgan element tavsiya etilgan qiymatlar ro‘yxatidan tanlangan bo‘lsa, serverga yuboriladigan qiymatni belgilaydi;
- ***selected*** – tanlangan atribut sahifa yuklanganda tanlanishi kerak bo‘lgan variantni ko‘rsatish uchun ishlatilishi mumkin. Ushbu atributning qiymati tanlanishi kerak.

Agar bu atribut ishlatilmasa, sahifa yuklanganda birinchi variant ko‘rsatiladi. Agar foydalanuvchi biror variantni tanlamasa, birinchi element serverga ushbu boshqaruv qiymati sifatida yuboriladi (3.13-rasm);

- ***id, class*** – hujjat ichidagi identifikatorlar;
- ***lang, dir*** – matnning tili va yo‘nalishi haqidagi ma’lumotlar;
- ***title*** – elementning sarlavhasi (element mazmuni ustiga kursov olib kelganda brauzer tomonidan izoh sifatida ko‘rsatiladi);
- ***style*** – o‘rnatilgan uslub haqida ma’lumot.

HTML

```
<form action="http://www.example.com/profile.php">
<p>What device do you listen to music on?</p>
<select name="devices">
<option value="ipod">iPod</option>
<option value="radio">Radio</option>
<option value="computer">Computer</option>
</select>
</form>
```

NATIJA

What device do you listen to music on?

iPod ▾

3.13-rasm. Ochiladigan ro‘yxat oynasi

LABEL tegidan foydalanish

Belgilash qutilari yoki radio tugmalari ko‘rsatilganda, foydalanuvchiga yumaloq yoki kvadrat tugma taqdim etiladi. Tugmaning yonida tushuntirish matnini ko‘rsatishingiz mumkin, lekin u va radio tugmasi (tasdiqlash qutisi) brauzer tomonidan butunlay alohida, bog‘liq bo‘lмаган elementlar sifatida qabul qilinadi. Labellardan foydalanish qo‘shimcha ma’lumotlarni boshqaruv elementlari bilan bog‘lash imkonini beradi va foydalanuvchi teglar bilan o‘zaro aloqada bo‘lganida sodir bo‘lgan hodisalar brauzer tomonidan label ma’lumotlari bilan bog‘langan boshqaruv elementlari bilan bog‘liq hodisalar sifatida talqin qilinadi.

Belgilash katakchalari bo‘lsa, tushuntirish matni tasdiqlash qutisi label tarkibiga kiritilgan bo‘lsa, u holda matn ustidagi matnni bosish sichqonchani belgilash katakchasingin o‘zi ustiga bosish bilan teng bo‘ladi va uni almashtiradi. Agar, masalan, label formani yuborish tugmasi bilan bog‘langan bo‘lsa, labelning ko‘rsatilgan tarkibini bosish shaklni serverga yuboradi.

HTML5 da teglar LABEL tegi bilan amalga oshiriladi. Boshlanish va tugatish teglarini belgilash majburiydir.

LABEL teg atributlari ro‘yxati:

- **for** – labelni boshqaruvga bog‘lovchi zarur atribut. Ushbu atributning qiymati qo‘shirnoq ichiga olingan boshqaruv elementining id-nomi bo‘lib, xuddi shu hujjat ichida label bilan joylashgan (3.14-rasm);

Eslatma. Bunday holda, boshqaruv elementi ikkita nomga ega bo‘lishi kerak: boshqaruv elementining qiymati bilan bog‘langan name atributi va boshqaruv elementiga murojaat qilish uchun ishlataladigan id atributi.

HTML

```
<label>Age: <input type="text" name="age" /></label>
<br/>
Gender:
<input id="female" type="radio" name="gender"
value="f">
<label for="female">Female</label>
<input id="male" type="radio" name="gender"
value="m">
<label for="male">Male</label>
```

NATIJA

Age:

Gender: Female Male

3.14-rasm. Label tegini ishlatalishi

- **<fieldset>...</fieldset>** – formadagi bir necha elementlarni guruhlash uchun ishlataladi. Guruhlangan elementlar chiziq bilan chegaralanadi. Uning attributlari:
 - **form** – boshqa formalar bilan bog‘lanish (identifikator orqali);
 - **title** – formadagi guruhga yordam sifatida yozuv chiqarish.
- **<legend>...</legend>** - “fieldset” orqali hosil qilingan guruhga nom berish uchun ishlataladi. Bu nom chizilgan chegara chizig‘ining tepa qismida paydo bo‘ladi.

Attributlari quyidagilar:

- align – guruh nomini tekislash: left, right, center;
- title – forma nomiga sichqoncha oborilsa, yordam sifatida yozuv chiqarish(3.15-rasm).

HTML

```
<fieldset>
<legend>Contact details</legend>
<label>Email:<br />
<input type="text" name="email" /></label><br />
<label>Mobile:<br />
<input type="text" name="mobile" /></label><br />
<label>Telephone:<br />
<input type="text" name="telephone" /></label>
</fieldset>
```

NATIJA

Contact details

Email:

Mobile:

Telephone:

3.15-rasm. Fieldset elementidan foydalanish

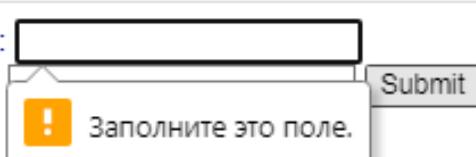
HTML5 imkoniyatlari

Formani tekshirish

Internetda siz forma boshqaruvi to‘g‘ri to‘ldirilmagan bo‘lsa, foydalanuvchilarga xabar beruvchi shakllarni ko‘rgan bo‘lishingiz mumkin. Bu shaklni tekshirish deb nomlanadi. An’anaga ko‘ra, formani tekshirish JavaScript yordamida amalga oshirildi. Ammo HTML5 tekshirishni joriy qiladi. Misol uchun, yuborish tugmasi bosilganda to‘ldirilmagan maydonlarni to‘ldirish haqida ma’lumot beradi(3.16-rasm).

```
<form method="post">
<label for="username">Username:</label>
<input type="text" name="username"
required="required" /></title><br />
<label for="password">Password:</label>
<input type="password" name="password"
required="required" />
<input type="submit" value="Submit" />
```

HTML



NATIJA

3.16-rasm. Formani tekshirish

Vaqtni kiritish(type= “date”)

<input>

Ko‘pgina formalar sanalar, elektron pochta manzillari va URL manzillari kabi ma’lumotlarni to‘plashi kerak. Bu an’anaviy ravishda matn kiritish yordamida amalga oshiriladi.

HTML5 ba’zi ma’lumotlarni to‘plash usulini standartlashtirish uchun yangi shakl boshqaruvlarini joriy qildi. Ushbu kirishlarni tanimaydigan eski brauzerlar ularni faqat bitta satr matn maydoni sifatida ko‘rib chiqadi.

type=“date”

Agar siz foydalanuvchidan sanani so‘rasangiz, <input> elementidan foydalanishingiz va type atributiga sana qiymatini berishingiz mumkin. Bu yangi HMTL5 kiritish turlarini qo‘llab-quvvatlaydigan brauzerlarda sana kiritishni yaratadi(3.17-rasm).

```
<form method="post">
<label for="username">Departure date:</label>
<input type="date" name="depart" />
<input type="submit" value="Submit" />
</form>
```

HTML

The screenshot shows a date picker interface. At the top, there is a text input field labeled "Departure date" containing "12.03.2022". To its right are two buttons: a calendar icon and a "Submit" button. Below the input field is a month calendar for March 2022. The days of the week are labeled in Russian: Пн, Вт, Ср, Чт, Пт, Сб, Вс. The dates are arranged in rows: Row 1: 28, 1, 2, 3, 4, 5, 6; Row 2: 7, 8, 9, 10, 11, 12, 13; Row 3: 14, 15, 16, 17, 18, 19, 20; Row 4: 21, 22, 23, 24, 25, 26, 27; Row 5: 28, 29, 30, 31, 1, 2, 3; Row 6: 4, 5, 6, 7, 8, 9, 10. The date "12" is highlighted with a blue border, indicating it is the selected date. At the bottom left of the calendar is a "Удалить" (Delete) link, and at the bottom right is a "Сегодня" (Today) link.

NATIJA

3.17-rasm. Vaqtni kiritish

Elektron pochta va URL kiritish

<input>

HTML5 shuningdek, tashrif buyuruvchilarga elektron pochta va URL manzillarini kiritish imkonini beruvchi kirish maydonlarini taqdim etadi.

Ushbu kiritish turlarini qo‘llab-quvvatlamaydigan brauzerlar ularni faqat matn qutilari sifatida ko‘rib chiqadi.

type=“email”

Agar siz foydalanuvchidan elektron pochta manzilini so‘rasangiz, elektron pochta kiritish usulidan foydalanishingiz mumkin.

Qo‘llab-quvvatlaydigan brauzerlar

HTML5 tekshirushi foydalanuvchi ma’lumotni elektron pochta manzilining to‘g‘ri formatida taqdim etganligini tekshiradi. Ba’zi aqli telefonlar, shuningdek,

elektron pochta manzilini (masalan, @ belgisi) kiritishda sizga kerak bo‘ladigan kalitlarni ko‘rsatish uchun klaviaturalarini optimallashtiradi(3.18-rasm).

```
<form method="post">
<p>Please enter your email address:</p>
<input type="email" name="email" />
<input type="submit" value="Submit" />
</form>
```

HTML

Please enter your email address:

!

Адрес электронной почты должен содержать символ "@". В адресе "ss" отсутствует символ "@".

3.18-rasm. Pochta ma’lumotlarini kiritish

type = “url”

URL kiritish foydalanuvchidan web-sahifa manzilini so‘raganiningizda ishlatalishi mumkin. HTML5 tekshiruvini qo‘llab-quvvatlaydigan brauzerlar foydalanuvchi URL formatida ma’lumot bergenligini tekshiradi. Ba’zi aqli telefonlar, shuningdek, URL manzilini kiritishda sizga kerak bo‘ladigan kalitlarni ko‘rsatish uchun klaviaturalarini optimallashtiradi(3.19-rasm).

```
<form method="post">
<p>Please enter your website address:</p>
<input type="url" name="website" />
<input type="submit" value="Submit" />
</form>
```

HTML

Please enter your website address:

!

Введите адрес сайта.

3.19-rasm. Url manzilini kiritish

Qidiruv tugmasi

<input>

Agar siz qidiruv so‘rovlari uchun bitta satr matn maydoni yaratmoqchi bo‘lsangiz, HTML5 bu maqsad uchun maxsus kiritish turini taqdim etadi.

type = “search”

Qidiruv so‘rovlari uchun bitta satr matn maydoni yaratmoqchi bo‘lsangiz, HTML5 maxsus qidiruv kiritishni ta’minlaydi.

HTML5 qidiruv maydonini yaratish uchun <input> elementi qiymati type bo‘lgan tip atributiga ega bo‘lishi kerak. Eski brauzerlar uni oddiygina bitta satr matn oynasi deb qaraydilar.

Placeholder

Har qanday matn kiritishdan oldin, siz o‘sha joyni bosmaguncha matn maydonida ko‘rsatiladigan, qiymati matn bo‘lgan to‘ldiruvchi(**Placeholder**) atributdan ham foydalanishingiz mumkin(3.20-rasm).

Eski brauzerlar bu xususiyatga e’tibor bermaydilar.

```
<form>
<p>Search:</p>
<input type="search" name="search"
placeholder="Enter keyword" />
<input type="submit" value="Search" />
</form>
```

HTML

Search:

NATIJA

Search

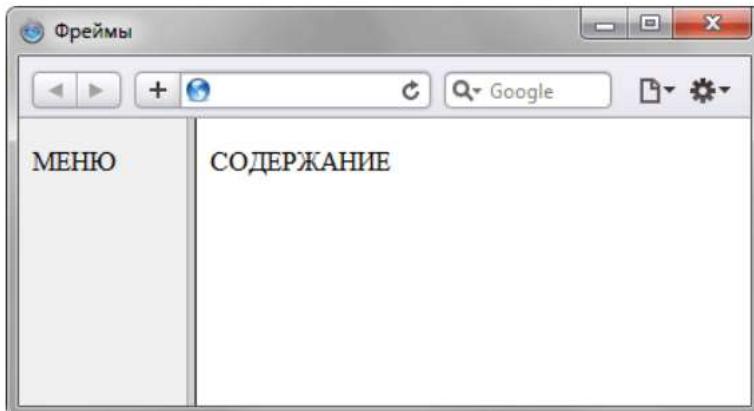


3.20-rasm. Qidiruvni tugmasi bilan ishlash

3.3. Freymlar hosil qilish va ular bilan ishlash

Freymlar saytlar tobora kamdan-kam bo‘lib borayotganiga qaramay, HTMLni o‘rganish freym mavzusini hisobga olmagan holda to‘liq bo‘lmaydi. Bundan tashqari, freymlar ma’lum ma’noda o‘z o‘rnini egallagan, boshqaruv va yordam tizimlari uchun ishlatiladi.

Freym yaratish uchun hujjatdagi <body> tegining o‘rnini bosuvchi va ekranni hududlarga bo‘lish uchun ishlataladigan <frameset> tegidan foydalaniladi. Bu teg ichida <frame> teglari mavjud bo‘lib, ular hududga yuklanishi mo‘ljallangan HTML hujjatiga ishora qiladi (3.21-rasm).



3.21-rasm. Freymga misol

Freymlardan foydalanganda sizga kamida uchta HTML fayl kerak bo‘ladi: birinchisi ramka tuzilishini belgilaydi va brauzer oynasini ikki qismga ajratadi, qolgan ikkita hujjat belgilangan oynalarga yuklanadi. Freymlar soni ikkitaga teng bo‘lishi shart emas, balki undan ko‘p, lekin ikkitadan kam bo‘lishi kerak emas, aks holda freymlardan foydalanishning ma’nosi butunlay yo‘qoladi.

Keling, rasmida ko‘rsatilgan sahifaga asoslangan freymlar yaratish bosqichlarini ko‘rib chiqaylik. 13.21-rasmdagi ko‘rinish uchun bizga uchta fayl kerak bo‘ladi: index.html - hujjat strukturasini belgilaydi, menu.html - chap freymga yuklanadi va content.html – o‘ng freymga yuklanadi. Ulardan faqat index.html o‘z kodining tuzilishi bilan boshqa fayllardan farq qiladi.

index.html fayli kodi

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Фреймы</title>
</head>
<frameset cols="100,*">
<frame src="menu.html" name="MENU">
<frame src="content.html" name="CONTENT">
</frameset>
</html>
```

Agar kodning birinchi qatorida freymlar ishlatsa, quyidagi hujjat turi yoziladi.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

Bu <!DOCTYPE> brauzerga freymlar bilan ishlayotganligini bildiradi; <head> konteynerida sahifa kodlash va hujjat sarlavhasi kabi odatiy ma'lumotlar mavjud. Shuni yodda tutingki, HTML-fayllar freymlar ichida ochilguncha sarlavha bir xil bo'lib qoladi.

Ushbu misolda brauzer oynasi cols atributi yordamida ikki ustunga bo'lingan, chap ustun 100 pikselni, o'ng ustun esa yulduzcha belgisi bilan belgilangan qolgan joyni egallaydi. Ramkalarning kengligi yoki balandligi jadvallarga o'xshash foiz sifatida ham o'rnatilishi mumkin.

<frame> tegi src atributi yordamida belgilangan maydonga yuklangan HTML fayl nomini bildiradi. Chap oynaga menu.html nomli fayl yuklanadi, o'ng oynaga content.html yuklanadi. Hujjatlarni nom atributi yordamida ko'rsatilgan oynaga yuklash uchun har bir freymga o'ziga xos nom berish tavsiya etiladi.

menu.html fayli kodi

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
  
<html>  
  
<head>  
  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
  <title>Навигация по сайту</title>  
  
</head>  
  
<body style="background: #f0f0f0">  
  <p>МЕНЮ</p>  
  
</body>  
  
</html>
```

Ushbu misolda sahifadagi kulrang fon uslublar yordamida o'rnatiladi.

content.html fayli kodi

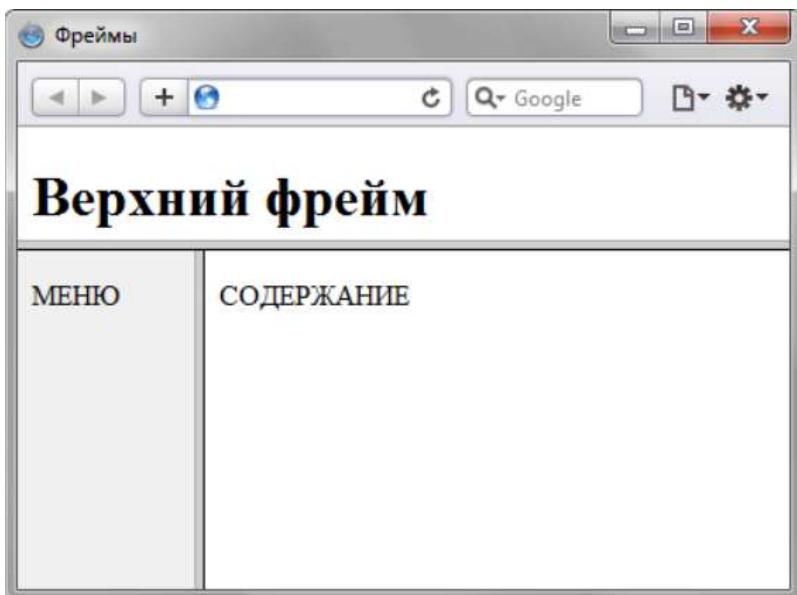
```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Содержание сайта</title>
  </head>
  <body>
    <p>СОДЕРЖАНИЕ</p>
  </body>
</html>

```

Keling, uchta freym yordamida murakkabroq misolni ko‘rib chiqaylik (13.22-rasm).



13.22-rasm. 3 freymli sahifani tashkil qilish

Bunday holda, <frameset> tegi yana, lekin ikki marta, bir teg ikkinchisining ichiga joylashtirilgan holda ishlatiladi. Gorizontal bo‘linish **rows** atributi orqali yaratiladi, bu yerda turli o‘lchamlar uchun foiz belgisi ishlatiladi.

Bunday holda, <frameset> tegi yana, lekin ikki marta, bir teg ikkinchisining ichiga joylashtirilgan holda ishlatiladi. Gorizontal bo‘linish satrlar atributi orqali yaratiladi, bu erda turli uchun foiz belgisi ishlatiladi.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">

```

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
    <title>Фреймы</title>
  </head>
  <frameset rows="25%, 75%">
    <frame src="top.html" name="TOP" scrolling="no" noresize>
    <frameset cols="100, *">
      <frame src="menu.html" name="MENU">
      <frame src="content.html" name="CONTENT">
    </frameset>
  </frameset>
</html>

```

Ushbu misoldan ko‘rinib turibdiki, rows atributiga ega <frameset> konteyneri dastlab ikkita gorizontal ramka hosil qiladi, lekin ikkinchi ramka o‘rniga boshqa <frameset> almashtiriladi. Vertikal aylantirish paneli paydo bo‘lishining oldini olish va foydalanuvchi yuqori ramka hajmini mustaqil ravishda o‘zgartira olmasligi uchun scrolling= “no” va noresize atributlari qo’shilgan.

Nazorat savollari

- 3.1. Formalar nima uchun ishlatiladi?
- 3.2. Formaning qanday turlari bor?
- 3.3. HTML5 da formalarning qanday imkoniyatlari mavjud?
- 3.4. Freymlar qanday yaratiladi?
- 3.5. Formada ko‘p martali tanlov qaysi forma elementi yordamida amalaga oshiriladi?
- 3.6. Formada qidiruv qaysi element yordamida yaratiladi?
- 3.7. HTML5 da pochta va sahifa manzilini kiritish qaysi element yordamida amalga oshiriladi?

IV-BOB. KASKADLI STILLAR JADVALI(CSS), CSS STILLARNI QO'LLASH, HTMLGA ULANISH USULLARI. SELEKTOR TURLARI

4.1. Css kirish. Kaskadli uslublar jadvallarini HTML hujjatlariga ulash

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

Birinchidan, kaskadli uslublar jadvallari nima ekanligini va ular nima uchun ishlatilishini tushunishga harakat qilaylik. Uslublar jadvali – bu uslublar jadvali biriktirilgan hujjatning HTML teglariga qo‘llaniladigan CSSda yozilgan qoidalar to‘plami. Ushbu qoidalar HTML teglari tarkibining vizual tasvirini belgilaydi. Yoki boshqacha qilib aytganda, ular monitor ekranida qanday ko‘rinishi (rang, ramkalar, o‘lcham, tekislash, shrift, to‘ldirish, ko‘rinish va boshqalar)ni belgilaydi.. CSS3 tufayli ushbu parametrlarni atributlari orqali har bir teg uchun alohida css xususiyatlarini o‘rnatish shart emas.

Keling, buni bir misol bilan ko‘rib chiqaylik.

Aytaylik, hujjatda H3da berilgan bir nechta sarlavhalar mavjud va yaratuvchi ularni yashil va kursiv bilan ko‘rsatishi kerak, qolgan barcha matnlar esa qora bo‘lishi kerak. Standart HTML vositalaridan foydalangan holda, hujjat muallifi har bir sarlavha uchun rangni yashil va kursivga o‘rnatishi kerak(4.1-rasm).

HTML	NATIJA
<pre><HTML> <HEAD> <TITLE>Css da ishlash </TITLE> <STYLE type="text/css"> h3{ color:green; font-style:italic; } </STYLE> </HEAD> <BODY> <h3>3-darajali sarlavha</h3> hujjat matni hujjat matni <h3>3-darajali sarlavha</h3> hujjat matni hujjat matni </BODY></HTML></pre>	<p>3-darajali sarlavha</p> <p>hujjat matni hujjat matni</p> <p>3-darajali sarlavha</p> <p>hujjat matni hujjat matni</p>

4.1-rasm. Css da html hujjatga ishlov berish

E’tibor bering, misolda faqat bitta HTML tegi CSS xususiyatlaridan foydalangan holda optimallashtirilgan va hujjat soddalashtirilgan.

Bundan tashqari, CSS3 funksiyalaridan mohirona foydalanish (yakka holda yoki DHTML bilan birgalikda) tasvirlangan original vizual effektlarni amalga oshirish imkonini beradi.

Shunday qilib, kaskadli uslublar jadvallari HTML-hujjat mazmunini tashqi ko‘rinishi tavsifidan ajratish imkonini beradi. Bundan tashqari, buning natijasida sezilarli qulaylik, HTML elementlarining vizual xususiyatlarini tavsiflashning ixchamligi va ularni o‘zgartirish tezligiga erishiladi.

Kaskadli uslublar jadvallari HTML hujjatining boshida (STYLE tegining tarkibida) yoki tashqi css faylida joylashishi mumkin. Bunda uslublar jadvali Link sarlavhasi tegi orqali ulanadi. Tashqi uslublar jadvallaridan foydalanish, ayniqsa, barcha sahifalari bitta dizayn yechimida ishlab chiqilishi kerak bo‘lgan ko‘p sahifali saytlar uchun muhim ahamiyatga ega. Shuning uchun, har bir sahifaning kodiga bir xil uslub sozlamalarini yozish o‘rniga, ularni bir marta yozish, ularni tashqi CSS fayliga joylashtirish va ularni saytning barcha sahifalariga ulash kifoya. Uslublar jadvallari kaskadli deb ataladi, chunki bir nechta uslublar jadvallari bitta HTML hujjatiga ulanganda, ular o‘zlarining ustuvorligiga ko‘ra, hujjat “ishlaydigan” kaskadga joylashadilar.

Kaskadli uslublar jadvallarini HTML hujjatlariga qanday ulash mumkin?

Uslublar jadvali qoidalarini HTML hujjatiga qo'llashi uchun uslublar jadvali va hujjatni bir-biriga bog'lash kerak.

Buni amalga oshirishning to'rtta usuli mavjud:

- O'rnatish – uslublar jadvalini STYLE tegining tarkibi sifatida to'g'ridan-to'g'ri HTML hujjatining boshiga o'rnatish;
- Bog'lash – uslublar jadvali tashqi faylda joylashadi va HTML hujjatiga LINK tegi orqali biriktiriladi. Biroq, tashqi uslublar jadvaliga ega CSS fayli har doim hujjatning HTML fayli bilan birga keladi;
- Import – bu usulning mohiyati shundan iboratki, serverdagи tashqi faylda joylashgan uslublar jadvali matni HTML fayl matni ichidagi Css-xususiyati @import yordamida import qilinadi;
- STYLE atributi barcha HTML teglari ichida foydalanishni belgilaydi. U orqali, CSS3 sintaksisidan foydalanib, siz har bir teg uchun uslubni alohida belgilashingiz (yoki bekor qilishingiz) mumkin.

1. O'rnatish, yuqorida aytib o'tilganidek, STYLE HTML sarlavha tegi tomonidan amalga oshiriladi. Uning tarkibida kaskadli uslublar jadvali o'rnatiladi. Bunday holda, STYLE tegining tip atributi "text/css" ga o'rnatilishi kerak(4.2-rasm).

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

4.2-rasm. O'rnatish usuli orqali cssdan foydalanish

2. HTML hujjatiga Css faylni biriktirish LINK tegi yordamida amalga oshiriladi(4.3-rasm).

CSS stillarini alohida faylga yozib, bu faylni HTML hujjatida chaqirib ishlatalish mumkin. Misol uchun, “style.css” nomli fayl yaratib ichiga quyidagi stil xususiyat va qiymatlarini kiritamiz.

```
Style.css
H1
{
color: blue;
font-size:14 px
}
```

index.html hujjat yaratib, bu faylni yuqorida yaratgan “**style.css**” nomli fayl bilan bitta papkaga joylashtiramiz, so‘ng bu faylga quyidagi HTML teglarni kiritamiz.

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



4.3-rasm. Css faylni bog‘lash orqali ulash

3. Serverdagи tashqi faylda saqlangan uslublar jadvalini STYLE tegining tarkibiga import qilishingiz mumkin yani <style> tegi orqali CSS faylni import qilish mumkin.

Bu kaskadli uslublar jadvallarining maxsus ©import xususiyati orqali amalga oshiriladi. Bunday holda, STYLE tegini o‘rnatish quyidagicha bo‘ladi:

```
<STYLE type=“text/css”>
@import URL(www.servername/mystyles.CSS)
</STYLE>
©import xususiyatining qiymati import qilinadigan uslublar jadvalini o‘z
ichiga olgan faylning URL manzilidir.
```

STYLE tegi ©import xususiyati tavsifi bilan birga boshqa qoidalarni ham o‘z ichiga olishi mumkin. Ular faqat ©import xususiyatini o‘rnatgandan so‘ng ko‘rinadi. Misol uchun:

```
<STYLE type="text/css">
@import URL (www.servername/file_of_style)
body
{
font-family:sans-serif;
font-size:14 px;
}
p
{
Background-color: yellow;
}
</STYLE>
```

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

Individual teg uslubi kamdan-kam qo‘llaniladi va berilgan teg uchun allaqachon o‘rnatilgan uslub qoidalarini bekor qilish uchun ham ishlatiladi, chunki u boshqalarga nisbatan eng yuqori ustunlikka ega.

4.2. CSS3 da mavjud sintaksis qoidalari va selectorlar

Ushbu darsda CSS3 uslublar jadvallarini yozish grammatikasi ko‘rib chiqiladi. CSS ning barcha versiyalari umumiy sintaksis qoidalariga asoslanadi. Aynan shu holat tufayli CSS ning turli versiyalarining uzlusizligi ta’milnadi.

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

- ✓ *kodlarni ixchamlash uchun, turli elementlar uchun bir xil turdag'i stillarni guruhlab ishlatish maqsadga muvofiqdir.*

H1 {font-family: Verdana;}

H2 {font-family: Verdana;}

Yuqoridagi stilni guruhlab quyidagicha yozamiz.

H1, H2 {font-family: Verdana;}

✓ *Guruhashda bir xil element uchun bir necha stillarni birlashtirish mumkin.*

H2 {font-weight: bold;}

H2 {font-size: 14pt;}

H2 {font-family: Verdana;}

Barcha stillarni jamlaymiz.

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

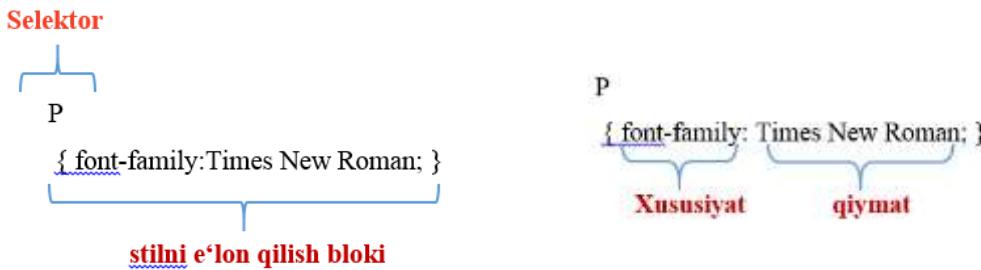
Ba’zi stillar bitta stil ichida berilishi ham mumkin.

H2 {font: bold 14pt Verdana;}

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

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

CSS qoidalari, **selektor** va **stilni e’lon qilish blokidan** tashkil topadi. Selektor har doim chap tomonda joylashadi va undan keyin figurali qavsniga ichida stilni e’lon qilish bloki joylashadi. Har bir e’lon, **hususiyat** va **uni qiymatidan** tashkil topadi. E’lonlar bir-biridan nuqtali vergul bilan ajratiladi. Aynan hususiyat stilni tashqi ko‘rinishini aniqlaydi.



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

Selektorlarning turlari

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

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

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

- ✓ **Sinf bo'yicha selector(class).** Nomidan ko'rinish turibdiki, bu mexanizm, berilgan elementlarni sinflarga ajratadi. Bu ish quyidagicha amalga oshiriladi: Bu hat boshida biror matn keltirilgan, bu hat boshidagi matn esa kichraytirilgan shrift bilan yoziladi. Ko'rsatilgan sinfga stillar quyidagi tartibda yoziladi:

```
P { color: #CCC; font-size: 14px; }
P.smaller { font-size: 12px; }
```

Bu holda birinchi hat boshi 14 piksel o‘lchamli shrift bilan yoziladi. Ikkinci hat boshi esa 12 piksel o‘lchamli shrift bilan yoziladi. Sintaksis ikkala holda ham o‘xshash, lekin sinf(class)ga murojaat nuqta orqali amalga oshiriladi, nuqtadan keyin sinf(class)ni nomi yoziladi. Agar ko‘rsatilgan sinf(class)ni faqatgina hat boshi uchun emas, balki ixtiyoriy element uchun qo‘llasak, stil quyidagicha yoziladi:

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

Agar ikkita <P> va <PRE> element va bir xil sinf(class) bilan berilsa, ularni differentsiallash mumkin:

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

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

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

✓ **ID bo‘yicha selector.** Unikal identifikatorlarni qisqa qilib, ID deb ataymiz. Berilgan hujjatda faqat bir martagina ID selektoridan foydalanish mumkin.

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

```
<H1 ID = “first”> Unikal sarlavha</H1>
```

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

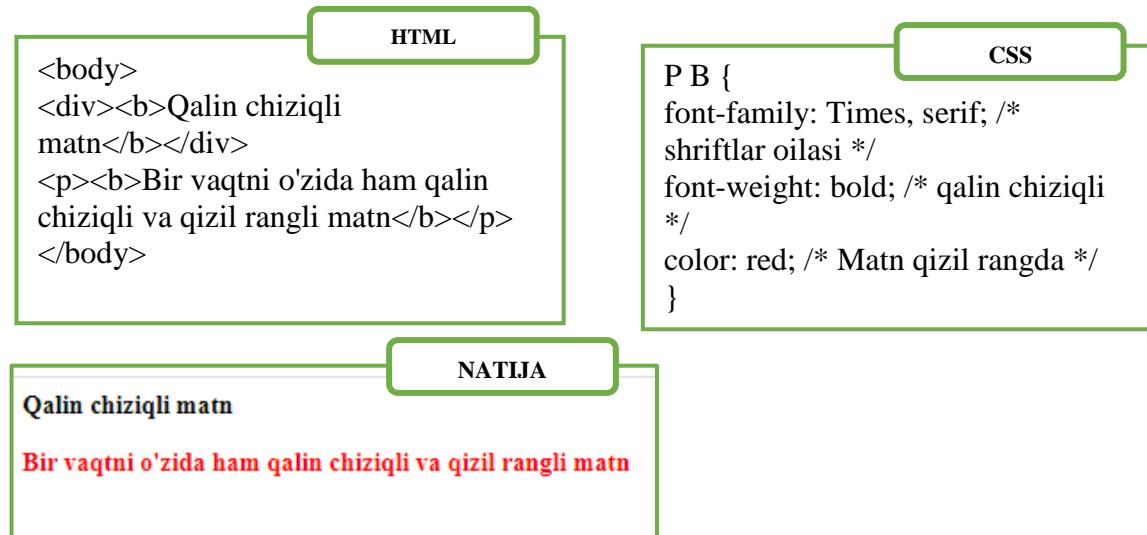
```
H1#first {  
    color: #00F;  
    font-family: Verdana, sans-serif;  
    font-weight: bold;  
    text-align: center;}
```

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

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

Bu yerda E konteyner teg, F esa – <E> konteynerni ichidagi teg. Bu holda stil <F> tegi uchun qo‘llaniladi va quyidagi kodga amal qilinadi <E><F> ...</F></E>.

Faqatgina ikkita teg emas, balki undan yuqori qatlamlar ishlatalish mumkin. Bu holda konstruktsiya quyidagicha bo‘lishi mumkin: div div ul li { ... }.



4.4-rasm. KonMatn selektor

✓ ***Universal selektor.*** Ba’zi hollarda web-sahifaning hamma elementlari uchun bir vaqtida bitta stil o‘rnatishga to‘g‘ri keladi. Bu holda universal selektordan foydalilaniladi. Sintaksis quyidagicha bo‘ladi:

*{ Stil qoidalarini ifodalash }

Universal selektordan foydalinish uchun yulduzcha (*) belgisi ishlataladi.

✓ ***Tur selektori*** – Element nomlariga mos keladi. Misol uchun:

```

h1, h2, h3 {
color: red;
}

```

Sahifadagi <h1>, <h2> va <h3> elementlarini belgilaydi.

✓ ***Child selector*** – boshqa elementning bevosita bolasi bo‘lgan elementga mos keladi. Misol uchun:

```

li>a {
color: red;
font-size: 25px;
}

```

 elementining bolalari bo‘lgan har qanday <a> elementlarini o‘z ichiga oladi (lekin sahifadagi boshqa <a> elementlarni emas).

✓ ***Qo'shni selector***(Adjacent Sibling Selector) – o'zidan keying elementni belgilaydi. Misol uchun:

```
h1+p {  
    color: red;  
    font-size: 25px;  
}
```

Har qanday <h1> elementdan keyingi birinchi <p> elementni hisobga oladi (lekin boshqa <p> elementlarini emas).

4.3. Css da rang berish usullari

Color xususiyati element ichidagi matn rangini belgilash imkonini beradi. CSSda har qanday rangni uchta usuldan birida belgilashingiz mumkin:

✓ ***Rgb qiymatlari***

Bunda ranglarni yaratish uchun qancha qizil, yashil va ko'k ishlatalishini ifodalaydi. Masalan: rgb(100,100,90)

✓ ***hex kodlari***

Bu olti xonali kodlar bo'lib, ular qizil, yashil va ko'k rangdagi miqdorni ifodalaydi, ular oldidan # belgisi qo'yiladi. Masalan: #ee3e80

✓ ***rang nomlari***

Brauzerlar tomonidan tan olingan 147 ta oldindan belgilangan rang nomlari mavjud. Masalan: DarkCyan.

Misol uchun:

```
/* color name */  
h1 {  
color: DarkCyan;}  
/* hex code */  
h2 {  
color: #ee3e80;}  
/* rgb value */  
p {  
color: rgb(100,100,90);}
```

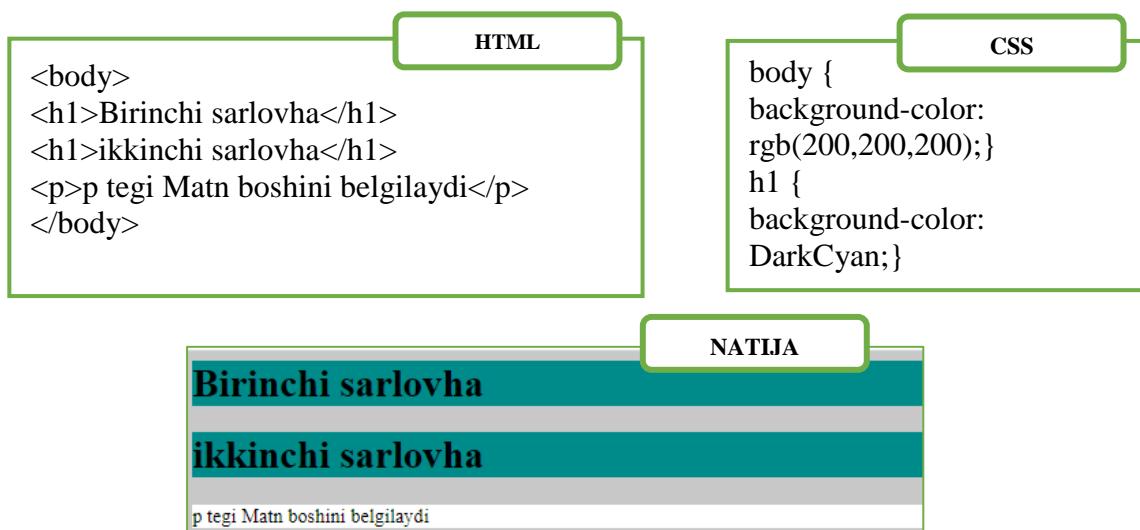
✓ ***background-color – orqa fonga rang berish.***

CSS har bir HTML elementiga xuddi qutida ko‘rinadigandek munosabatda bo‘ladi va background-color xususiyati ushbu quti uchun fon rangini o‘rnatadi(4.5-rasm).

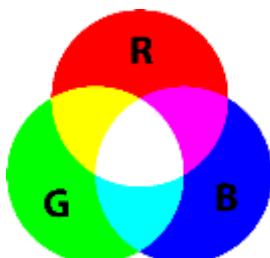
Fon rangini uchta usulda belgilash mumkin: RGB qiymatlari, hex kodlar va rang nomlari.

Agar fon rangini belgilamasangiz, u holda fon shaffof bo‘ladi.

Odatiy bo‘lib, ko‘pchilik brauzer oynalari oq fonga ega, ammo brauzer foydalanuvchilari o‘z oynalari uchun fon rangini o‘rnatishlari mumkin.



4.5-rasm. Elementlarning orqa foniga rang berish



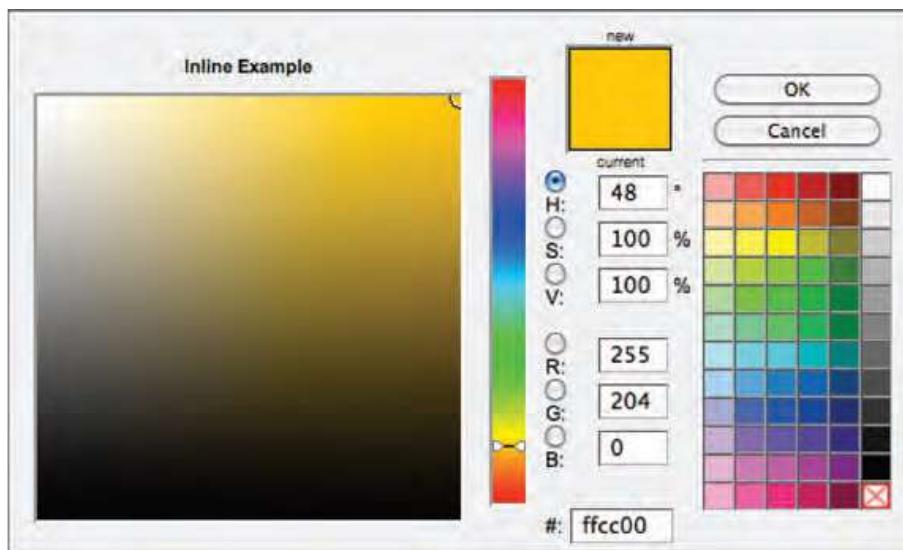
Kompyuter ekranidagi har bir rang qizil, yashil va ko‘k ranglarni aralashtirish orqali hosil bo‘ladi. Istalgan rangni topish uchun siz rang tanlash vositasidan foydalanishingiz mumkin.

Kompyuter monitorlari piksel deb ataladigan minglab mayda kvadratlardan iborat (agar siz monitoringizga diqqat bilan qarasangiz, ularni ko‘rishingiz mumkin).

Ekran yoqilmaganda, u qora rangda bo‘ladi, chunki u hech qanday yorug‘lik chiqarmaydi. U yoqilganda, har bir piksel boshqa rangga ega bo‘lib, tasvir yaratishi mumkin.

Ekrandagi har bir pikselning rangi qizil, yashil va ko‘k aralashmasi bilan ifodalanadi - xuddi televizor ekranidagi kabi.

Rang tanlash vositalari Photoshop va GIMP kabi tasvirlarni tahrirlash dasturlarida mavjud. R, G, B degan radio tugmalar yonida ko'rsatilgan RGB qiymatlarini ko'rishingiz mumkin. Hex qiymati # belgisi yonida berilgan.



4.6-rasm. Rang belgilash usullari

CSS 3: Shaffoflik(rgba, hsla)

CSS3 element va uning har qanday asosiy elementlarining shaffofligini belgilash imkonini beradi. Shaffoflik qiymati 0,0 dan 1,0 gacha bo'lgan qiymatda beriladi (shuning uchun 0,5 qiymati 50% shaffoflik va 0,15 15% shaffoflik).

CSS3 rgba xususiyati xuddi RGB qiymatidagi kabi rangni belgilash imkonini beradi, lekin shaffoflikni ko'rsatish uchun to'rtinchi qiymatni qo'shadi. Bu qiymat alfa qiymati sifatida tanilgan va 0,0 dan 1,0 gacha bo'ladi. Rgba qiymati faqat qo'llaniladigan elementga ta'sir qiladi (bola(child) elementlarga emas).

Ba'zi brauzerlar RGBA ranglarini taniy olmasligi mumkin. Agar bitta elementga taalluqli ikkita qoida mavjud bo'lsa, ulardan ikkinchisi ustuvor bo'ladi.

Color. Bu hususiyat elementning ichidagi matnni rangini berish vositasidir.

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

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

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

Misol.

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

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

NAZORAT SAVOLLARI

- 4.1.Css nima uchun qo‘llaniladi?
- 4.2.Html hujjatidagi elementlarga Css ni qanday usullar yordamida qo‘llash mumkin?
- 4.3.Css da rangning qanaday turlaridan foydalilanadi?
- 4.4.Css da matn va orqa fonga rang berish qaysi xususiyat orqali amalga oshiriladi.
- 4.5.Background-image xususiyatining vazifasi?
- 4.6.Background-repeat xususiyatining qiymatlari?
- 4.7.Background-repeat xususiyatining vazifasi?

V-BOB. CSS DA MATNLAR, RASM VA XAVOLALARGA ISHLOV BERISH

5.1. Cssda matnlarga ishlov berish

Shrift xususiyatlari. CSS da shriftlarni boshqarish imkoniyati HTML ga qaraganda keng. Matnli bloklar uchun CSS qoidalardan foydalanish HTML kodi hajmini anchagina kamaytiradi.

✓ **Font-family.** Font-family xususiyati CSS qoidasi qo'llaniladigan element(lar) ichidagi har qanday matn uchun ishlatilishi kerak bo'lgan shriftni belgilash imkonini beradi.

Qiymatlari:

- *Times New Roman;*
- *Calibri;*
- *Arial va h.k.*

✓ **Font-size.** Font-size xususiyati shrift uchun o'lchamni belgilash imkonini beradi. Shrift o'lchamini belgilashning bir necha yo'li mavjud. Eng keng tarqalganlari:

- *Piksellar(px).* Piksellar ko'p ishlatiladi, chunki ular web-dizaynerlarga matnlari qancha joy egallashini aniq nazorat qilish imkonini beradi. Piksellar sonidan keyin px harflari keladi;
- *Foizlar(%).* Shrift o'lchami foizlarda belgilanadi. Brauzerlardagi matnning standart hajmi 16px. Shunday qilib, 75% o'lcham 12px ga ekvivalent bo'ladi va 200% 32px bo'ladi;
- *Em.* Ems foydalilanayotgan shriftning balandligi(5.1-rasm).



5.1-rasm. O'lchov usullari

✓ **Font-weight.** Shriftni qalinlik darajasini belgilaydi.

Qiymatlari:

- *Normal* – matn oddiy holatda ko‘rinadi;
 - *Bold* – Bu matnning qalın ko‘inishiga olib keladi.
- ✓ **Font-style.** Kursiv yoki og‘ma shriftni o‘rnatadi.

Qiymatlari:

- *Normal* – matn oddiy holatda ko‘rinadi;
 - *Italic* – matn kursiv stili o‘rnataladi;
 - *Oblique* – bu matnning qiya ko‘inishiga olib keladi.
- ✓ **text-transform.** Matnning registrini o‘zgartirish uchun ishlataladi.

Qiymatlari:

- *capitalize* – har bir so‘zning birinchi harfini registri o‘zgaradi, ya’ni bosh harfda yoziladi;
 - *uppercase* – hamma harflarning registri bosh harfga almashadi;
 - *lowercase* – hamma harflarning registri yozma harfga almashadi;
 - *none* – harflarning registri almashmaydi.
- ✓ **text-decoration.** Bu quyidagi qiymatlardan birini qabul qilishi mumkin:
- *none* – qo‘shimcha bezash ishlatilmaydi;
 - *underline* – matn tagiga chiziq chizadi;
 - *overline* – matnni har bir qatorini ustiga chizadi;
 - *line-through* – matnni har bir qatori chizib tashlangan holda aks etadi;
 - *blink* – ekranda yonib, o‘chib turadigan matn aks etadi.
- ✓ **text-indent.** Bu hususiyat orqali hat boshining birinchi qatoriga otstup beriladi. Matnli blokni ko‘rsatishda ishlatiladi. Birinchi galda hat boshi va sarlavhalar uchun qo‘llaniladi. Misol. .credits { text-indent: 20px; }
- ✓ **line-height** – bu tipograflar matn satrlari orasidagi vertikal bo‘shliq uchun foydalananadigan atama. Misol: p { line-height: 1.4em; }
- ✓ **text-align.** Bu hususiyat matnning tekislash usulini o‘rnatish imkonini beradi.

Qiymatlari:

- *left* – chap hoshiyaga tekislash;
- *center* – o‘rtaga tekislash;
- *right* – o‘ng hoshiyaga tekislash;
- *justify* – butun sahifa bo‘ylab tekislash.

```

<!DOCTYPE html>
<html>
<head> <title>Text</title>
<style type="text/css">
body {
padding: 20px;}
h1, h2, h3, a {
font-weight: normal;
color: #0088dd;
margin: 0px;}
h1 {
font-family: Georgia, Times, serif;
font-size: 250%;
text-shadow: 2px 2px 3px #666666;
padding-bottom: 10px;}
h2 {
font-family: "Gill Sans", Arial, sans-serif;
font-size: 90%;
text-transform: uppercase;
letter-spacing: 0.2em;}
h3 {
font-size: 150%;}
p {
font-family: Arial, Verdana, sans-serif;
line-height: 1.4em;
color: #665544;}
p.intro:first-line {
font-weight: bold;}
.credits {
font-style: italic;
text-align: right;}
a {
text-decoration: none;}
a:hover {
text-decoration: underline;}
</style> </head>
<body>
<h1>Briards</h1>
<h2>Mo'ynaga o'rالgan yurak</h2>
<p class="intro"><a class="breed" href="http://en.wikipedia.org/wiki/Briard">briard</a>yoki berger de brie - an'anaviy ravishda cho'pon va qo'ylar qo'riqchisi sifatida ishlatiladigan yirik it zoti</p>
<h3>Zot tarixi</h3>
<p>Frantsiyada paydo bo'lган deb hisoblangan briard asrlar davomida qo'ylarni boqish va himoya qilish uchun etishtirilgan. Bu zot frantsuz armiyasi tomonidan qo'riqchilar, xabarchilar va eshitish qobiliyati tufayli yaralangan askarlarni qidirish uchun ishlatilgan. Briardlar Birinchi jahon urushida deyarli yo'q bo'lib ketish nuqtasiga qadar ishlatilgan. Hozirgi vaqtida briardlar soni asta-sekin tiklanmoqda. Charlemagne, Napoleon, Tomas Jefferson va Lafayette briardlarga ega edi.</p>
<p class="credits">Ivy Duckett tomonidan</p>
</body> </html>

```

Briards
БОЛГАРДА ОРАЛГАН ЎРАК
Бриард биргер де бри - айнанувиштада еле-паш ва қаржига сарнадиган ўзинибоз
Зор турли
Пратекада пирло бўйнида бўлаларнинг бриллент жумла давомиди, оғизлени боғлиқ ва биримга оғизленинига, бу зор турлини аммо вакондан тарзига тийб, худоюннинг яхшиятини
яралаган ишларни оғизленинига мөланган. Бриард Бриллент шундай оғизленинига яхшиятини - бозига ваконда сарнадиган ўзинибозга оғизленинига да, биримнига да, биримнига да.

by Cassetta Webdesign

5.2-rasm. Shirft xususiyatlari bilan ishslash

Chegara xususiyatlari

✓ **border-width(chegara kengligi)** – xususiyati chegaraning kengligini boshqarish uchun ishlataladi. Ushbu xususiyatning qiymati **piksellarda** yoki ulardan biri yordamida berilishi mumkin.

qiymatlari:

- thin – ingichka chiziqli chegara;
- medium – o‘rtacha qalinlikdagi chegara;
- thick – qalin chiziqli chegara.

✓ **border-style(chegara uslubi)** – Siz chegara uslubi xususiyatidan foydalanib, chegara uslubini boshqarishingiz mumkin. Bu xususiyat quyidagi qiymatlarni olishi mumkin:

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

Siz turli xil chegaralarning uslublarini alohida o‘zgartirishingiz mumkin:

- ✓ **border-top-style;**
- ✓ **border-left-style;**
- ✓ **border-right-style;**
- ✓ **border-bottom-style.**
- ✓ **border-color(chegara rangi)** – Chegara rangini RGB qiymatlari, hex yoki rang nomlari yordamida belgilash mumkin.

Quyidagilar yordamida qutining turli tomonlaridagi chegaralarning ranglarini alohida nazorat qilish mumkin:

- ✓ border-top-color – yuqori segmentning rangi;
- ✓ border-right-color – o‘ng segmentning rangi;
- ✓ border-bottom-color – quyi segmentning rangi;
- ✓ border-left-color – chap segmentning rangi.
- ✓ **Border(Chegara chiziqlari)** – Chegara xususiyati kenglik, uslub va chegara rangini bittada belgilash imkonini beradi.

HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Text</title>
<style type="text/css">
p {
width: 250px;
border: 3px dotted #0088dd; } }
</style>
</head>
<body>
<p>
```

Qutining o'lchamini belgilashning eng mashhur usullari piksellar, foizlar yoki emlardan foydalanishdir. An'anaga ko'ra, piksellar eng mashhur usul bo'lib kelgan, chunki ular dizaynerlarga ularning o'lchamlarini aniq nazorat qilish imkonini beradi. Foizlarni ishlatganda, qutining o'lchami brauzer oynasining o'lchamiga nisbatan bo'ladi.

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

NATIJA

Qutining o'lchamini belgilashning eng mashhur usullari piksellar, foizlar yoki emlardan foydalanishdir. An'anaga ko'ra, piksellar eng mashhur usul bo'lib kelgan, chunki ular dizaynerlarga ularning o'lchamlarini aniq nazorat qilish imkonini beradi. Foizlarni ishlatganda, qutining o'lchami brauzer oynasining o'lchamiga nisbatan bo'ladi.

5.3-rasm. Border xususiyatidan foydalanish

Padding – Kontentning atrofidagi bo‘sh soha hoshiya (padding) deyiladi. To‘ldirish xususiyati element tarkigi va uning chegarasi o‘rtasida qancha bo‘sh joy qolishi kerakligini belgilash imkonini beradi(5.4-rasm).

Ushbu xususiyatning qiymati ko‘pincha piksellarda ko‘rsatiladi (lekin foizlar yoki emslardan foydalanish ham mumkin). Agar foiz ishlatilsa, to‘ldirish brauzer oynasining (yoki boshqa cuti ichida bo‘lsa, o‘z ichiga olgan qutining) foizi sifati qaraladi.

HTML

```
<!DOCTYPE html>
<html> <head>
<title>Text</title>
<style type="text/css">
p {
width: 275px;
border: 2px solid #0088dd;}
p.example {
padding: 40px;}
</style></head>
<body>
<p>Qutining o‘lchamini belgilashning eng mashhur usullari piksellar, foizlar yoki emlardan foydalanishdir. An'anaga ko'ra, piksellar eng mashhur usul bo'lib kelgan, chunki ular dizaynerlarga ularning o'lchamlarini aniq nazorat qilish imkonini beradi. Foizlarni ishlatganda, qutining o'lchami brauzer oynasining o'lchamiga nisbatan bo'ladi.
</p> <p class="example"> Qutining o‘lchamini belgilashning eng mashhur usullari piksellar, foizlar yoki emlardan foydalanishdir. An'anaga ko'ra, piksellar eng mashhur usul bo'lib kelgan, chunki ular dizaynerlarga ularning o'lchamlarini aniq nazorat qilish imkonini beradi. Foizlarni ishlatganda, qutining o'lchami brauzer oynasining o'lchamiga nisbatan bo'ladi.
</p>
</body> </html>
```

NATIJA

Qutining o‘lchamini belgilashning eng mashhur usullari piksellar, foizlar yoki emlardan foydalanishdir. An'anaga ko'ra, piksellar eng mashhur usul bo'lib kelgan, chunki ular dizaynerlarga ularning o'lchamlarini aniq nazorat qilish imkonini beradi. Foizlarni ishlatganda, qutining o'lchami brauzer oynasining o'lchamiga nisbatan bo'ladi.

Qutining o‘lchamini belgilashning eng mashhur usullari piksellar, foizlar yoki emlardan foydalanishdir. An'anaga ko'ra, piksellar eng mashhur usul bo'lib kelgan, chunki ular dizaynerlarga ularning o'lchamlarini aniq nazorat qilish imkonini beradi. Foizlarni ishlatganda, qutining o'lchami brauzer oynasining o'lchamiga nisbatan bo'ladi.

5.4-rasm. Padding xususiyatidan foydalanish

Quyidagilardan foydalanib, qutining har bir tomoni uchun turli qiymatlarni belgilashingiz mumkin:

- padding-top;
- padding-right;
- padding-bottom;
- padding-left.

✓ **Margin** – Margin xususiyati qutilar orasidagi bo'shliqni nazorat qiladi. Uning qiymati odatda piksellarda beriladi, lekin siz foizlar yoki emlardan ham foydalanishingiz mumkin(5.5-rasm).

Quyidagilardan foydalanib, qutining har bir tomoni uchun qiymatlarni belgilashingiz mumkin:

- margin-top;
- margin-right;
- margin-bottom;
- margin-left.

HTML

```
<style type="text/css">
p {
    width: 200px;
    border: 2px solid #0088dd;
    padding: 20px;
}
p.example {
    margin: 40px;
    padding: 0px;
}</style>
```

NATIJA

Qutining o'lchamini
belgilashning eng mashhur
usullari piksellar, foizlar yoki
emlardan foydalansidir.

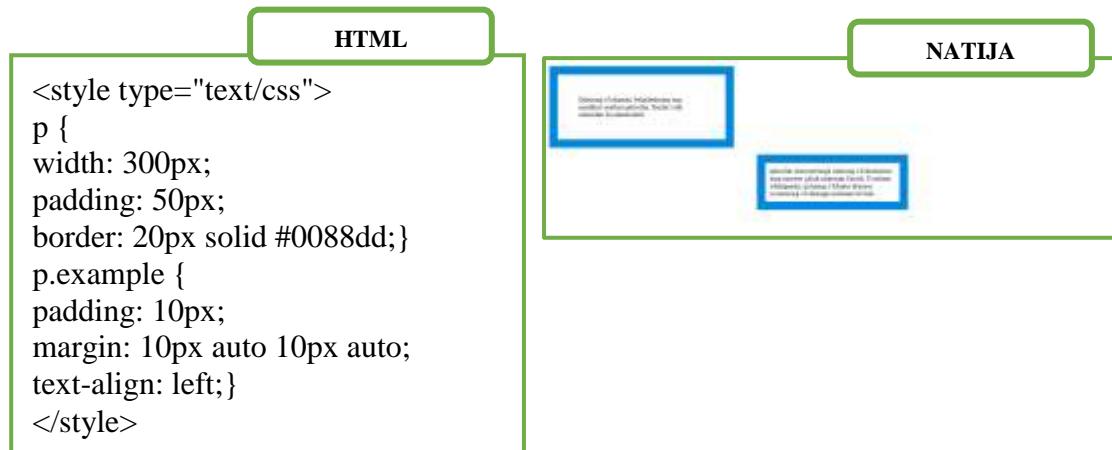
o'lchamni daxsynerberga
o'shamning o'lchamlarini aniq
nazorat qilish imkonini beradi.
Foydalanishiga qandayda, qutining
o'lchami browser oynasining
o'lchamiga nisbatan beladi.

5.5-rasm. Margin xususiyatidan foydalanish

Agar siz sahifadagi qutini(box) markazga qo'ymoqchi bo'lsangiz (yoki uni o'rnatilgan element ichiga markazlashtirmoqchi bo'lsangiz), chap(left-margin) va o'ng(right-margin) chekkalarni avtomatik(auto) qilib belgilashingiz mumkin.

Sahifada qutini markazlashtirish uchun siz qutining kengligini belgilashingiz kerak (aks holda u sahifaning to'liq kengligini egallaydi).

Qutining kengligi belgilaganidan so‘ng, chap va o‘ng chekkalarni avtomatik xolatga o‘rnatilsa brauzerning o‘zi qutining har bir tomonidan teng bo‘shliq qoldiradi(yoki cuti ichida joylashgan element ichida).



5.6-rasm. Margin:auto; xususiyatini o‘rnatish

- ✓ ***border-radius*** – chegara uchlariga radius berish.

5.2. Rasmlarga ishlov berish

CSSda kenglik va balandlik xususiyatlaridan foydalanib, tasvir hajmini boshqarish mumkin.

Tasvir o‘lchamlarini belgilash sahifalarni yanada qulayroq yuklashga yordam beradi, chunki HTML va CSS kodlari ko‘pincha rasmlardan oldin yuklanadi va brauzerga tasvir uchun qancha bo‘sh joy qoldirish kerakligini aytib berish, rasmning yuklab olinishini kutmasdan sahifaning qolgan qismini ko‘rsatishga imkon beradi.

Ko‘plab saytlar o‘zlarining ko‘plab sahifalarida bir xil o‘lchamdagি tasvirdan foydalanadilar.

Web-sahifa mualliflari `` elementining align atributidan foydalanish o‘rniga, rasmlarni tekislash uchun float xususiyatidan tobora ko‘proq foydalanmoqda. Bunga odatda ikkita usul bilan erishiladi:

1. float xususiyati tasvir o‘lchamini ifodalash uchun yaratilgan sinfga qo‘shiladi.
2. Tasvirlarni sahifaning chap yoki o‘ng tomoniga tekislash uchun align-left yoki align-right kabi nomlar bilan yangi sinflar yaratiladi.

Ushbu sinf nomlari tasvir hajmini ko‘rsatadigan sinflarga qo‘shimcha sifatida ishlatiladi.

Ushbu misolda siz tasvirni tekislash uchun ishlatiladigan align-chap va align-o‘ng sinflarini ko‘rishingiz mumkin. Matn ularning chetlariga tegmasligini ta’minlash uchun rasmga margin xususiyati berilgan(5.7-rasm).

```
<!DOCTYPE html>
<html>
<head>
<title>Text</title>
<style type="text/css">
img.align-left {
float: left;
margin-right: 10px;}
img.align-right {
float: right;
margin-left: 10px;}
img.medium {
width: 150px;
height: 150px;}
</style>
</head>
<body>
<p>
<b><i>Magnolia</i></b> is a large genus that
contains over 200 flowering plant species...</p>
<p>
Some magnolias, such as <i>Magnolia stellata</i>
and <i>Magnolia soulangeana</i>. flower quite
```

HTML



Magnolia is a large genus that contains over 200 flowering plant species...

Some magnolias, such as *Magnolia stellata* and *Magnolia soulangeana*, flower quite early in the spring before the leaves open...

NATIJA

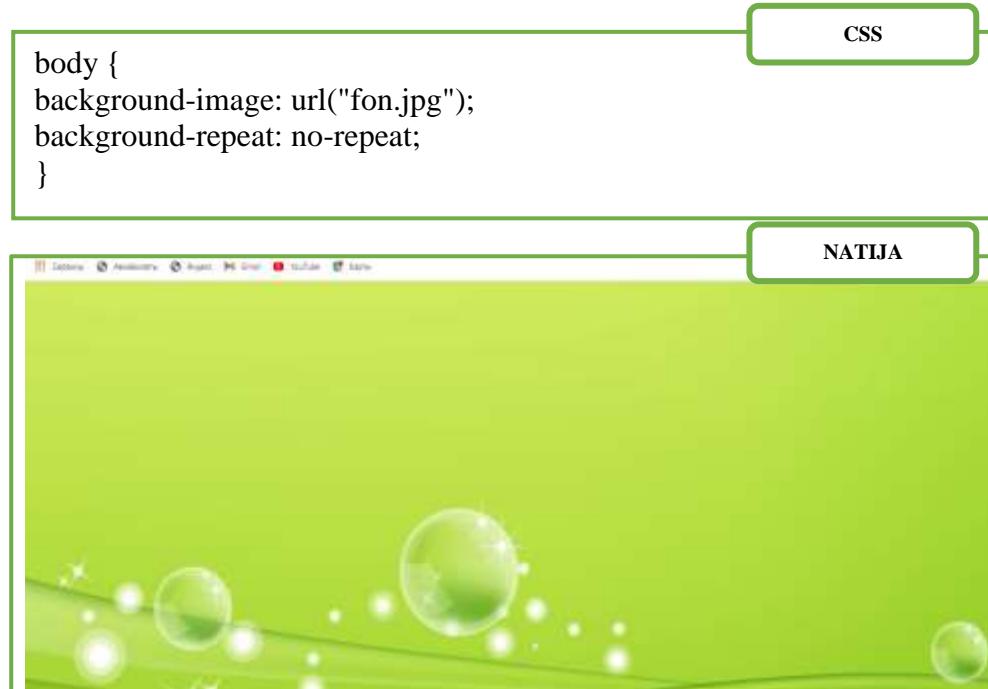


5.7-rasm. Rasmlarga css xususiyatlarini qo‘llash

➤ ***background-image***. Orqa fon rasm joylashtirish uchun ushbu xususiyat ishlataladi. Rasmga boradigan yo‘l urlda keyin, qavslar va qo‘shtirnoqlar ichiga qo‘yiladi(7.8-rasm).

➤ Orqa fondagi rasm takrorlanishi kerak bo‘lmasa,

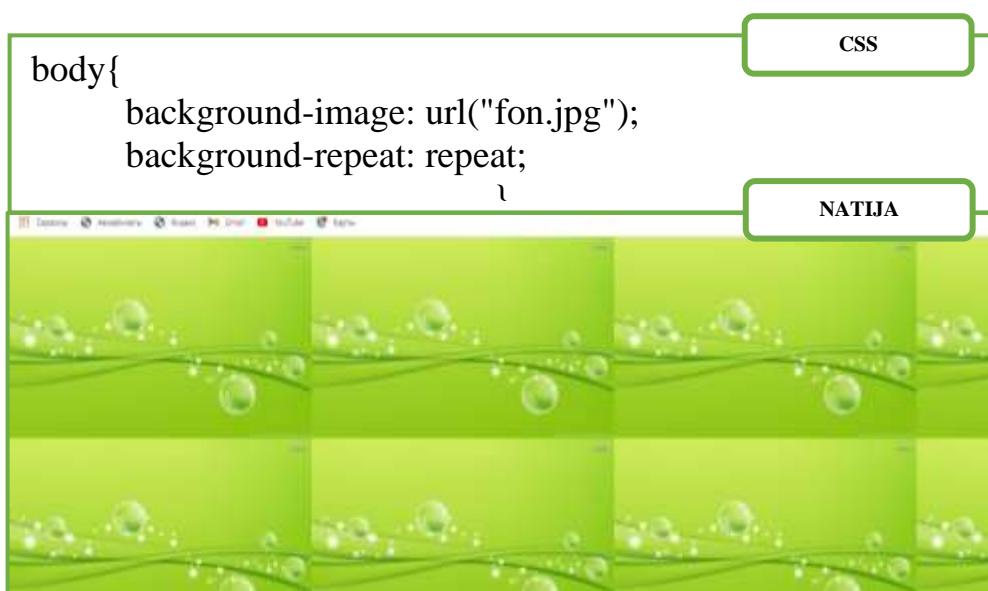
background-repeat: no-repeat; xususiyati va qiymati beriladi



5.8-rasm. Orqa fonga rasm qo‘yish

➤ Orqa fondagi rasm takrorlanishi kerak bo‘lsa,

background-repeat: repeat; xususiyati va qiymati beriladi(5.9-rasm).

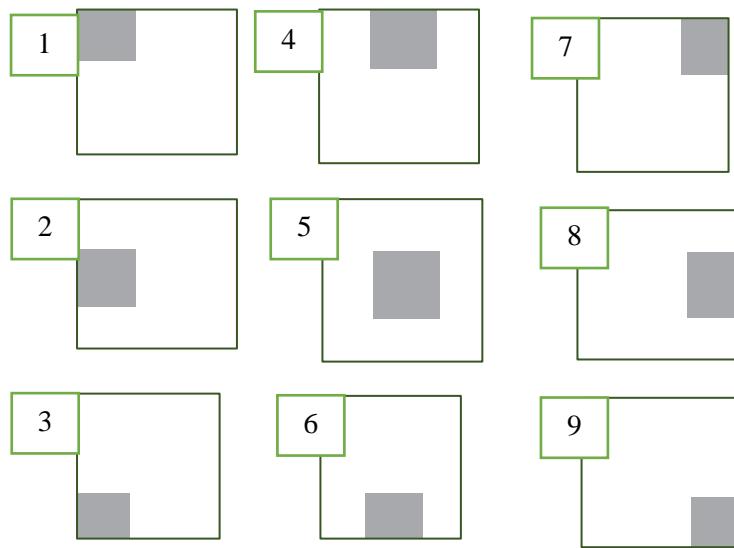


5.9-rasm. Orqa fondagi rasmning takrorlanish xususiyati

- **Background-position.** Tasvir takrorlanmasa, brauzer oynasida fon tasviri qayerga joylashtirilishini belgilash uchun fon holati (background-position) xususiyatidan foydalanishingiz mumkin.

Bu xususiyat odatda bir juft qiymatga ega. Birinchisi **horizontal** holatni, ikkinchisi esa **verticalni** ifodalaydi.

- 1.left top
- 2.Left center
- 3.Left bottom
- 4.Center top
- 5.Center center
- 6.Right top
- 7.Right top
- 8.Right center
- 9.Right bottom



Namuna:

```
body {
    background-image: url("images/tulip.gif");
    background-repeat: no-repeat;
    background-position: 50% 50%;}
```

5.3. Xavolalarga ishlov berish

CSS da xavola (ssilka) lar bilan ishlash uchun to‘rt xil Psevdosinf mavjud:

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

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

a: link

```
{  
Color: green;  
Font size: 14px;  
Xususiyat: qiymat;  
....oddiy (ko‘rib chiqilmagan) xavola (ssilka) ni bezash stili  
}
```

a:active

```
{  
Color: green;  
Font size: 14px;  
... aktiv xavola (ssilka) ni bezash stili ...}
```

a:visited

```
{  
Color: green;  
Font size: 14px;  
...ko‘rib chiqilgan xavola (ssilka) ni bezash stili...}
```

a:hover

```
{  
Color: green;  
Font size: 14px;  
... ko‘rib chiqilmoqchi bo‘lgan xavola (ssilka) ni bezash stili }  
Psevdosinflardan, oddiy sinf va identifikatorlar bilan birgalikda turli  
kombinatsiya tuzib ham foydalanish mungkin.
```

```
<A class=“pagelink” href=“page.html”>... tekst...</A>  
a.pagelink:hover  
{  
... stilni ifodalash ...  
}
```

Yoki identifikator bilan birgalikda foydalanish:

```
<A id=“mylink” href=“page.html”>  
...tekst...  
</A>
```

Kursor turlari – Cursor

✓ **Cursor (xususiyat)** – Kursor xususiyati foydalanuvchilarga ko‘rsatilishi kerak bo‘lgan sichqoncha kursorining turini boshqarish imkonini beradi(7.10-rasm).

Misol uchun, formada foydalanuvchi kursorni uning ustiga olib borganida kursorni qo‘l sifatida o‘rnatishingiz mumkin.

Atributlari:

- auto
- crosshair
- default
- pointer
- move
- text
- wait
- help

The screenshot shows a browser window with three tabs: 'chapter-14/cursor.html', 'HTML', and 'CSS'. The 'HTML' tab displays the following code:

```
<a href="http://www.whitmanarchive.org">  
Walt Whitman</a>
```

The 'CSS' tab displays the following code:

```
a {  
cursor: move;}
```

The 'NATIJA' tab shows the result: a link labeled 'Walt Whitman' with a mouse cursor icon (a hand) positioned over it.

5.10-rasm. Cssda kursorlar bilan ishslash

NAZORAT SAVOLLARI

- 5.1. Cssda matnlarga ishlov beruvchi qanday atributlar mavjud?
- 5.2. Padding nima va u qanday o‘rnatiladi?
- 5.3. Margin nima va u qanday o‘rnatiladi?
- 5.4. Orqa fonga rasm qaysi xususiyat yordamida beriladi?
- 5.5. Cssda xavolalarga ishlov beruvchi qanday xususiyatlar mavjud?

VI-BOB. CSSDA RO‘YXATLAR, JADVALLAR VA SHAKLLARGA ISHLOV BERISH

6.1. Cssda ro‘yxatlarga ishlov berish

✓ **List-style-type** – xususiyati tartiblangan va tartiblanmagan ro‘yxatlarning shaklini boshqarish imkonini beradi.

U , va elementlariga taalluqli qoidalarda ishlatilishi mumkin.

Tartibsiz ro‘yxatlar

Tartibli ro‘yxat uchun siz quyidagi **qiymatlardan** foydalanishingiz mumkin:

list-style-type:

None(yoq;

- *Disc*(ichi bo‘yalgan doira);
- *Circle*(ichi bo‘yalmagan doira);
- *Square*(to‘rtburchak).

Tartibli ro‘yxatlar

Tartibli ro‘yxat uchun siz quyidagi **qiymatlardan** foydalanishingiz mumkin:

list-style-type:

- *None(yoq);*
- *Decimal(1 2 3);*
- *decimal-leading-zero (01 02 03);*
- *lower-alpha(a b c);*
- *upper-alpha(A B C);*
- *lower-roman(i. ii. iii.);*
- *upper-roman(I II III).*

list-style-image – xususiyatidan foydalanib, tartiblangan va tartiblanmagan ro‘yxatlarning shakli uchun tasvirni belgilash mumkin.

Qiymat url harflari bilan boshlanadi. Qavslar ichida tasvirga yo‘l qo‘sh tirnoq ichida beriladi.

Bu xususiyatdan va elementlariga taalluqli qoidalarda foydalanish mumkin.

Ushbu misolda ro‘yxatdagi har bir element orasidagi vertikal bo‘shliqni oshirish uchun margin xususiyatidan foydalanishni ko‘rsatadi(6.1-rasm).

HTML	CSS	NATIJA
<pre><h1>Index of Translated Poems</h1> <h2>Arthur Rimbaud</h2> Ophelia To Music A Dream for Winter Vowels The Drunken Boat </pre>	<pre><style type="text/css"> ul { list-style-image: url("ruyxat.png"); } li { margin: 10px 0px 0px 0px; }</pre>	<h1>Index of Translated Poems</h1> <h2>Arthur Rimbaud</h2> <ul style="list-style-type: none"> Ophelia To Music A Dream for Winter Vowels The Drunken Boat

6.1-rasm. Cssda ro‘yxatlarga ishlov berish

✓ **list-style-position** – ushbu xususiyat nishonlarni o‘z ichiga olgan matn blokining ichki yoki tashqi tomonida paydo bo‘lishi kerakligini ko‘rsatadi.

Bu xususiyat ikkita qiymatdan birini qabul qilishi mumkin:

- ✓ **outside** – Marker matn blokining chap tomonida joylashgan. (Agar bu xususiyat ishlatilmasa, bu standart xatti-harakatlardir);
- ✓ **inside** – Marker matn qutisi ichida joylashadi.

Ko‘rsatilgan misolda ro‘yxatning kengligi 150 piksel bilan cheklangan. Bu matnning yangi satrga o‘tishini ta’minlaydi, shunda siz “inside” qiymati matnning birinchi qatoridagi nishon bilan qanday joylashishini ko‘rishimiz mumkin(5.2-rasm).

HTML	CSS	NATIJA
<pre><ul class="illuminations"> That idol, black eyes and Gracious son of Pan! ... When the world is reduced <ul class="season"> Once, if my memory serves Hadn't I once a youth ... Autumn already! ... </pre>	<pre>ul { border:solid 1px green; width: 150px;} li { margin: 10px;} ul.illuminations { list-style-position: outside;} ul.season { list-style-position: inside;}</pre>	<ul style="list-style-type: none"> That idol, black eyes and ... Gracious son of Pan! ... When the world is reduced ... <ul style="list-style-type: none"> Once, if my memory serves ... Hadn't I once a youth ... Autumn already! ...

6.2-rasm. Ro‘yxatlarning inside va outside xususiyatlari

List-style – boshqa bir qancha CSS xususiyatlarida bo‘lgani kabi, ro‘yxat uslublari uchun qisqartma vazifasini ham bajaradigan xususiyat mavjud. U ro‘yxat uslubi(list-type) deb ataladi.

```
ul {
list-style: inside circle; width: 300px;}
li { margin: 10px 0px 0px 0px;}
```

6.2. Jadvallarga chegaralar va fon qo‘shish

Siz allaqachon jadvallar bilan tez-tez ishlataladigan bir nechta xususiyatlarni uchratdingiz. Quyida biz ularni bitta misolda birlashtiramiz(6.3-rasm):

- ✓ **Width** – jadvalning kengligini belgilash uchun ishlataladi;
- ✓ **Padding** – har bir jadval katakchasi chegarasi va uning qismi orasidagi bo‘shliqni o‘rnatish uchun ishlataladi;
- ✓ **text-transform** – jadval sarlavhalari tarkibidagi matnni bosh harfga aylantirish uchun ishlataladi;
- ✓ **:hover** – foydalanuvchi bosgan jadval qatorini ajratib ko‘rsatish uchun ishlataladi;
- ✓ **letter-spacing, font-size** – jadval sarlavhalari tarkibiga qo‘shimcha uslublar qo‘shish maqsadida harflar oralig‘i, shrift o‘lchamini belgilash uchun ishlataladi.

```

<body>
<h1>Ilk auksion</h1>
<table> <tr>
<th>Avtorlar</th>
<th>Sarlovhasi</th>
<th class="money">Zaxira narxi</th>
<th class="money">Hozirgi taklif</th>
</tr> <tr>
<td>E.E. Cummings</td>
<td>Tulips & Chimneys</td>
<td class="money">$2,000.00</td>
<td class="money">$2,642.50</td> </tr>
<tr class="even">
<td>Charles d'Orleans</td>
<td>Poemes</td>
<td class="money"></td>
<td class="money">$5,866.00</td> </tr>
<tr>
<td>T.S. Eliot</td>
<td>Poems 1909 - 1925</td>
<td class="money">$1,250.00</td>
<td class="money">$8,499.35</td> </tr>
<tr class="even">
<td>Sylvia Plath</td>
<td>The Colossus</td>
<td class="money"></td>
<td class="money">$1031.72</td> </tr>
</table> </body>

```

HTML

```

<style type="text/css">
body {
font-family: Arial, Verdana, sans-serif;
color: #111111;}
table {
width: 600px;}
th, td {
padding: 7px 10px 10px 10px;}
th {
text-transform: uppercase;
letter-spacing: 0.1em;
font-size: 90%;}
border-bottom: 2px solid #111111;
border-top: 1px solid #999;
text-align: left;}
tr.even {
background-color: #efefef;}
tr:hover {
background-color: #c3e6e5;}
.money {
text-align: right;}
</style>

```

CSS

Ilk auksion

NATIJA

AVTORLAR	SARLOVHASI	ZAXIRA NARXI	HOZIRGI TAKLIF
E.E. Cummings	Tulips & Chimneys	\$2,000.00	\$2,642.50
Charles d'Orleans	Poemes		\$5,866.00
T.S. Eliot	Poems 1909 - 1925	\$1,250.00	\$8,499.35
Sylvia Plath	The Colossus		\$1031.72

6.3-rasm. Jadvallar bilan ishlanuvchi xususiyatlari

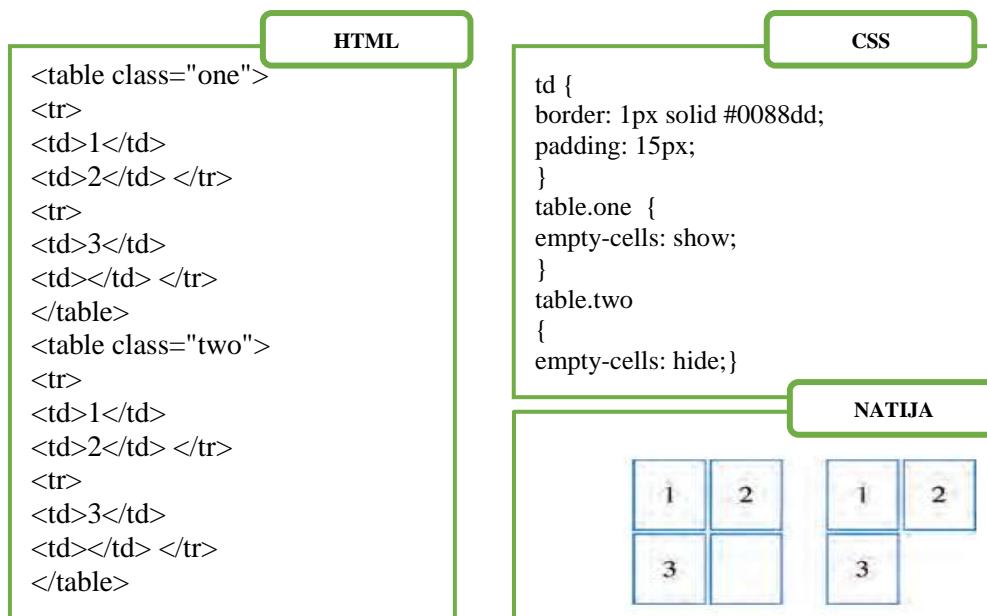
- ✓ **border-top, border-bottom** – jadval sarlavhalari ustidagi va ostidagi chegaralarni o‘rnatish uchun ishlatiladi;
- ✓ **text-align** – yozuvni jadvalning ba’zi katakchalarining chap tomoniga va boshqalarning o‘ng tomoniga tekislash uchun ishlatiladi;

- ✓ **background-color** – jadval qatorlarining fon rangini o‘zgartirish uchun ishlataladi;
- ✓ **empty-cells** – agar jadvalingizda bo‘sh katakchalar bo‘lsa, ularning chegaralari ko‘rsatilishi yoki ko‘rsatilmasligini belgilash uchun empty-cells xususiyatidan foydalanishingiz mumkin.

Brauzerlar bo‘sh katakchalarga turlicha munosabatda bo‘lganligi sababli, agar siz bo‘sh katakchadagi chegaralarni aniq ko‘rsatish yoki yashirishni istasangiz, bu xususiyatdan foydalaning.

U uchta **qiymatdan** birini qabul qilishi mumkin:

- **Show** – bu har qanday bo‘sh kataklarning chegaralarini ko‘rsatadi;
- **Hide** – bu har qanday bo‘sh kataklarning chegaralarini yashiradi;
- **Inherit** – agar sizda bitta jadval boshqasining ichiga joylashtirilgan bo‘lsa, meros qiymati jadval katakchalarini o‘z ichiga olgan jadval qoidalariga rioya qilishni buyuradi(6.4-rasm).



6.4-rasm. Empty-cells xususiyati

- ✓ **border-spacing**, **border-collapse** – ushbu xususiyatlar qo‘shti katakchalar orasidagi masofani boshqarish imkonini beradi.

Odatiy bo‘lib, brauzerlar ko‘pincha har bir jadval katakchasi orasida kichik bo‘shliqni qoldiradilar, shuning uchun agar siz ushbu bo‘shliqni ko‘paytirish yoki kamaytirishni istasangiz, chegara oralig‘i xususiyati(border-spacing, border-

collapse) bo'shliqni boshqarishga imkon beradi. Ushbu xususiyatning qiymati odatda piksellarda ko'rsatiladi. Gorizontal va vertikal oraliqlar uchun alohida raqamlarni belgilashni istasangiz, ikkita qiymatni belgilashingiz mumkin.

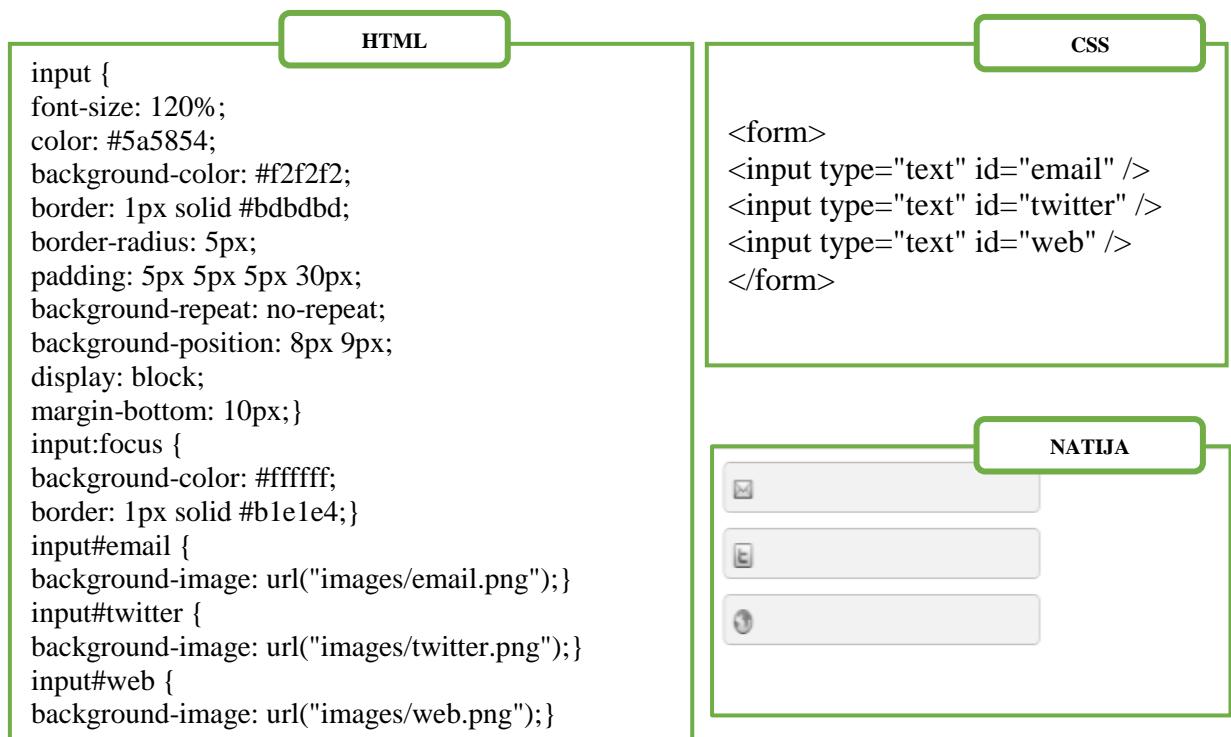
Mumkin qiymatlar:

- **collapse** – katakchalar bir-biriga suriladi va chegaralar orasida bo'shliq qolmaydi
- **separate** – Chegaralar bir-biridan ajratilgan holatga o'tadi.

6.3. Forma elementlarining ko'rinishini o'zgartirish

Quyidagi misolda xozigacha o'rgangan CSS xususiyatlarimizni forma elementlariga tadbiq etamiz(6.5-rasm).

- ✓ **font-size** foydalanuvchi kiritgan matn hajmini belgilaydi;
- ✓ **color** matn rangini, **background-color** esa input elementining fon rangini o'rnatadi;
- ✓ **border** input maydonining chetiga chegara qo'yadi va **border-radius** yumaloq burchaklar yaratish uchun ishlatiladi(bu xususiyatni qo'llab-quvvatlaydigan brauzerlar uchun);



6.5-rasm. Formaga css xususiyatlarini qo'llash

- ✓ **:focus** pseudo-sinfı matn kiritilayotganda fon rangini o‘zgartirish uchun ishlataladi va **:hover** psuedo-sinfı foydalanuvchi kursorni ularning ustiga olib borganida css *uslublarini* qo‘llaydi.
- ✓ **background-image** – qutiga fon tasvirini qo‘sadi. Har bir kirish(input)elementi uchun turli xil tasvir mavjud bo‘lganligi sababli, biz har bir kirishda id atributidan foydalanamiz.

Forma elementlari uchun teglar ko‘pincha turli uzunliklarda bo‘ladi, ya’ni forma boshqaruv elementlari bir tekis turmaydi. Bu misolda ko‘rsatilgan (shakl boshqaruviga CSS qo‘llanilmagan 6.6-rasm).

HTML	NATIJA
<pre><form method="post"> <div> <label for="name" class="title">Name:</label> <input type="text" id="name" name="name" /> </div></pre>	<p>Name: <input type="text"/></p> <p>Email adress: <input type="text"/></p> <p>Gender: <input type="radio"/> Male <input type="radio"/> Female</p> <p><input type="button" value="Register"/></p>

6.6-rasm. Forma elementlari joylashuvi

Bu formada biz foydalanuvchidan so‘ragan har bir mavzu, har bir savol yangi satrda paydo bo‘lishini ta’minlash uchun **<div>** elementi ichiga joylashtiriladi. Forma boshqaruv elementlari to‘g‘ri vertikal chiziqda tekislangan bo‘lsa, foydalanuvchilarga formani to‘ldirish osonroq bo‘ladi.

HTML	CSS
<pre><form method="post"><div> <label for="name" class="title">Name:</label> <input type="text" id="name" name="name" /> </div> <div> <label for="email" class="title">Email adress:</label> <input type="email" id="email" name="email" /> </div> <div> Gender: <input type="radio" name="gender" id="male" value="M" /> <label for="male">Male</label> <input type="radio" name="gender" id="female" value="F" /> <label for="female">Female</label>
 </div><div> <input type="submit" value="Register" id="submit" /> </div> </form></pre>	<pre><style type="text/css"> div { border-bottom: 1px solid #efefef; margin: 10px; padding-bottom: 10px; width: 260px;} .title { float: left; width: 100px; text-align: right; padding-right: 10px;} .radio-buttons label { float: none;} .submit { text-align: right;}</pre>

		NATIJA
Name: <input type="text"/>		
Email adress: <input type="text"/>		
Gender: <input type="radio"/> Male <input type="radio"/> Female		
<input type="button" value="Register"/>		

6.7-rasm. Formaga Css xususiyatini qo‘llash

Foydalanuvchilarning jinsini so‘ragan joyimizga qarasangiz, ikkita radio tugmachaning har birining o‘ziga xos <tegi> bor (biri erkak va boshqasi ayol). Sarlavhaga ushbu boshqaruv elementlarini moslashtirishga yordam beradigan elementi qo‘shildi.

Styling Fieldsets & Legends

Maydon to‘plamlari(Fieldsets), formadagi bir necha elementlarni guruhlash uchun ishlatiladi. Guruhlangan elementlar chiziq bilan chegaralanadi. Legends “fieldset” orqali hosil qilingan guruhga nom berish, bu nom chizilgan chegara chizig‘ining tepe qismida paydo bo‘ladi(6.8-rasm).

Ushbu ikki element bilan tez-tez ishlatiladigan xususiyatlar quyidagilarni o‘z ichiga oladi:

- ✓ **width** maydonlar to‘plamining (Fieldsets) kengligini boshqarish uchun ishlatiladi. Ushbu misolda maydonlar to‘plamining kengligi forma elementlarini yangi qatorda bo‘lishiga moslashtirilgan (Agar maydonlar to‘plamining kengligi kengroq bo‘lsa, elementlar bir qatorda joylashishi mumkin edi);
- ✓ **color** – matn rangini boshqarish uchun ishlatiladi;
- ✓ **background-color** – ushbu elementlarning orqasidagi rangni o‘zgartirish uchun ishlatiladi;
- ✓ **border** fieldset va/yoki Legends atrofidagi chegara ko‘rinishini nazorat qilish uchun ishlatiladi;

- ✓ **border-radius** ushbu xususiyatni qo‘llab-quvvatlaydigan brauzerlarda ushbu elementlarning qirralarini yumaloq qilish uchun ishlataladi;
- ✓ **padding** ushbu elementlar ichida bo‘sh joy qo‘shish uchun ishlatalishi mumkin.

HTML

```
<form><fieldset>
<legend>Newsletter</legend>
<label for="name">Name: </label><input type="text" id="name" />
<label for="email">Email: </label><input type="text" id="email" />
<input type="submit" value="Subscribe" id="submit" />
</fieldset></form>
```

CSS

```
<style type="text/css">
* {
    font-family: Arial, Verdana, sans-serif;
    color: #665544;}
input {
    border-bottom: 1px dotted #dcdcdc;
    border-top: none;
    border-right: none;
    border-left: none;
    padding: 5px;
    width: 280px;
    margin-bottom: 20px;}
input:focus {
    border: 1px dotted #dcdcdc;
    outline: none;}
input#submit {
    color: #ffffff;
    background-color: #665544;
    border: none;
    border-radius: 5px;
    width: 80px;}
input#submit:hover {
    color: #665544;
    background-color: #efefef;}
fieldset {
    width: 350px;
    border: 1px solid #dcdcdc;
    border-radius: 10px;
    padding: 20px;
    text-align: right;}
legend {
    background-color: #efefef;
    border: 1px solid #dcdcdc;
    border-radius: 10px;
    padding: 10px 20px;
    text-align: left;
    text-transform: uppercase;}
</style>
```

6.8-rasm. Fieldsets & Legends

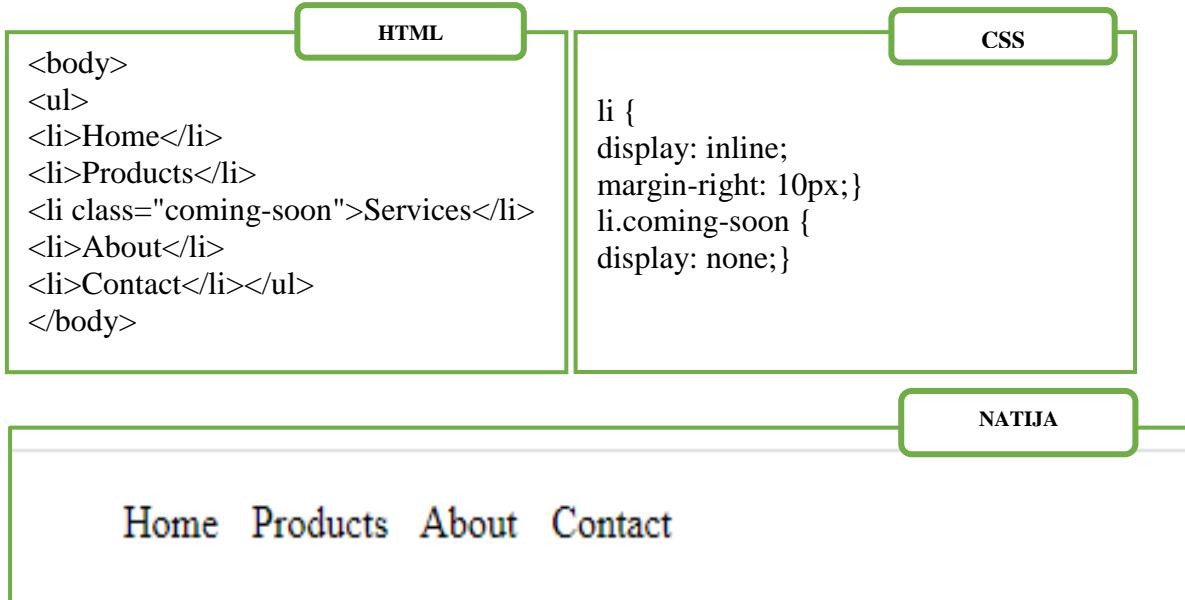
6.4. Bloklar bilan ishlash

Elementlarni joylashtirishdagi asosiy tushunchalar. Blok(quti)lar. Bloklarning xususiyatlari.

CSS har bir HTML elementiga xuddi o‘z qutisidagidek munosabatda bo‘ladi. HTML ning hamma elementlarini ikkita turga bo‘lish mumkin: blok(block) va qator turlari(inline).

✓ **Display** – Display **xususiyati** inline elementni blok darajasidagi elementga yoki blok elementni inline elementga aylantirish imkonini beradi va undan elementni sahifadan yashirish uchun ham foydalanish mumkin.

- **Inline** – bu blok darajasidagi elementni inline element kabi ishlashiga olib keladi;
- **Block** – bu inline elementning blok darajasidagi element kabi harakat qilishiga olib keladi;
- **inline-block** – bu blok darajasidagi elementning boshqa xususiyatlarini saqlab qolgan holda, inline element kabi oqishiga olib keladi;
- **None** – Bu sahifadagi elementni yashiradi. Bunday holda, element u sahifada umuman yo‘qdek harakat qiladi.



6.9-rasm. Display xususiyatiga misol

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

CSS da hujjatni formatlashda ham blok tushunchasidan foydalaniladi. CSS da bloklar to‘g‘ri to‘rtburchak ko‘rinishiga ega bo‘lgan mustaqil strukturaviy birlik hisoblanadi. CSSda bloklar struktura jixatidan olganda, bir-biridan ajratilgan.

✓ *quti o‘lchamlari*

Quti o‘lchamlarini o‘rnatish uchun balandlik va kenglik xususiyatlaridan foydalaniladi.

- **Width** – qutining kengligi;
- **Height** – qutining balandligi.

Qutining o‘lchamini belgilashning eng mashhur usullari piksellar, foizlar yoki emlardan foydalanishdir. An’anaga ko‘ra, piksellar eng mashhur usul bo‘lib kelgan, chunki ular dizaynerlarga ularning o‘lchamlarini aniq nazorat qilish imkonini beradi. Foizlarni ishlatganda, qutining o‘lchami brauzer oynasining o‘lchamiga nisbatan bo‘ladi.

```

<body>
<div class="box">
<p>Qutining o'lchamini belgilashning eng
mashhur usullari piksellar, foizlar yoki emlardan
foydanishdir. An'anaga ko'ra, piksellar eng
mashhur usul bo'lib kelgan, chunki ular
dizaynerlarga ularning o'lchamlarini aniq nazorat
qilish imkonini beradi. Foizlarni ishlatganda,
qutining o'lchami brauzer oynasining o'lchamiga
nisbatan bo'ladi.
</p> </div>
</body>

```

HTML

```

div.box {
height: 300px;
width: 300px;
background-color: #bbbbbaa;}
p {
height: 75%;
width: 75%;
background-color: #0088dd;}

```

CSS

Qutining o'lchamini belgilashning
eng mashhur usullari piksellar,
foizlar yoki emlardan
foydanishdir. An'anaga ko'ra,
piksellar eng mashhur usul bo'lib
kelgan, chunki ular dizaynerlarga
ularning o'lchamlarini aniq nazorat
qilish imkonini beradi. Foizlarni
ishlatganda, qutining o'lchami
brauzer oynasining o'lchamiga
nisbatan bo'ladi.

NATIJA

6.10-rasm. Quti o'lchamlarini qo'yish

✓ **Overflow.** Overflow xususiyati brauzerga qutidagi ma'lumot qutining o'zidan kattaroq bo'lsa, nima qilish kerakligini aytadi. U ikkita qiymatdan biriga ega bo'lishi mumkin:

- *Hidden* – bu xususiyat shunchaki qutiga sig'maydigan har qanday qo'shimcha ma'lumotni yashiradi;
- *Scroll* – bu xususiyat foydalanuvchilar ko'ra olmayotgan ma'lumotlarni ko'rishi uchun qutiga aylantirish paneli qo'shadi.

```

<body>
<p class="one">

```

HTML

Qutining o'lchamini belgilashning eng mashhur usullari piksellar, foizlar yoki emlardan foydanishdir. An'anaga ko'ra, piksellar eng mashhur usul bo'lib kelgan, chunki ular dizaynerlarga ularning o'lchamlarini aniq nazorat qilish imkonini beradi.

```

</p>
<p class="two">

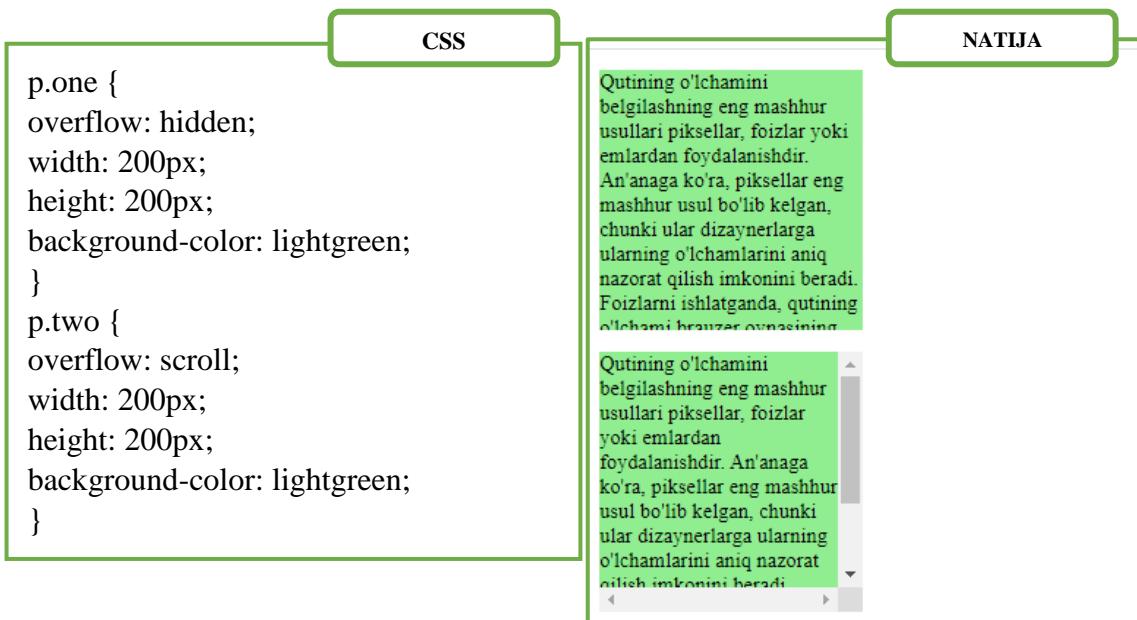
```

Qutining o'lchamini belgilashning eng mashhur usullari piksellar, foizlar yoki emlardan foydanishdir. An'anaga ko'ra, piksellar eng mashhur usul bo'lib kelgan, chunki ular dizaynerlarga ularning o'lchamlarini aniq nazorat qilish imkonini beradi..

```

</p>
</div>
</body>

```



6.11-rasm. Overflow xususiyat

6.5. Elementlarning joylashishini nazorat qilish

CSSda sahifa tartibini boshqarish imkonini beruvchi quyidagi joylashishni aniqlash sxemalari mavjud: oddiy oqim, nisbiy joylashishni aniqlash va mutlaq joylashishni aniqlash. CSSdagi joylashuv xususiyatidan foydalanib, joylashishni aniqlash sxemasi belgilanadi. Float xususiyatidan foydalanib, elementlarni joylashtirish ham mumkin.

Normal oqim

Normal oqim bo'yicha blokni joylashtirish sxemasi oddiy. Har bir blok darajasidagi element yangi satrda paydo bo'ladi, bu esa har bir elementning sahifada avvalgisidan pastroq ko'rinishiga olib keladi. Agar siz qutilarning kengligini belgilasangiz va ikkita element yonma-yon turishi uchun joy bo'lsa ham, ular bir-birining yonida ko'rinishiga olib keladi. Hamma struktura yoki qator turidagi bloklar vertikal yo'naliш boyicha yuqoridaн pastga qarab ketma-ket joylashaveradi va shu tarzda monitor ekranida aks etadi. Bunda ikkita strukturali blok hech qachon bir gorizontal chiziqda joylasha olmaydi. Normal oqim o'z hususiyatiga ko'ra HTML kod qanday tartibda yozilsa, shu tartibda ish bajaradi. Agar jadvaldan foydalanilmasa, normal oqim yordamida, murakkab sahifalar tayyorlash mumkin emas.

- ***position:static***

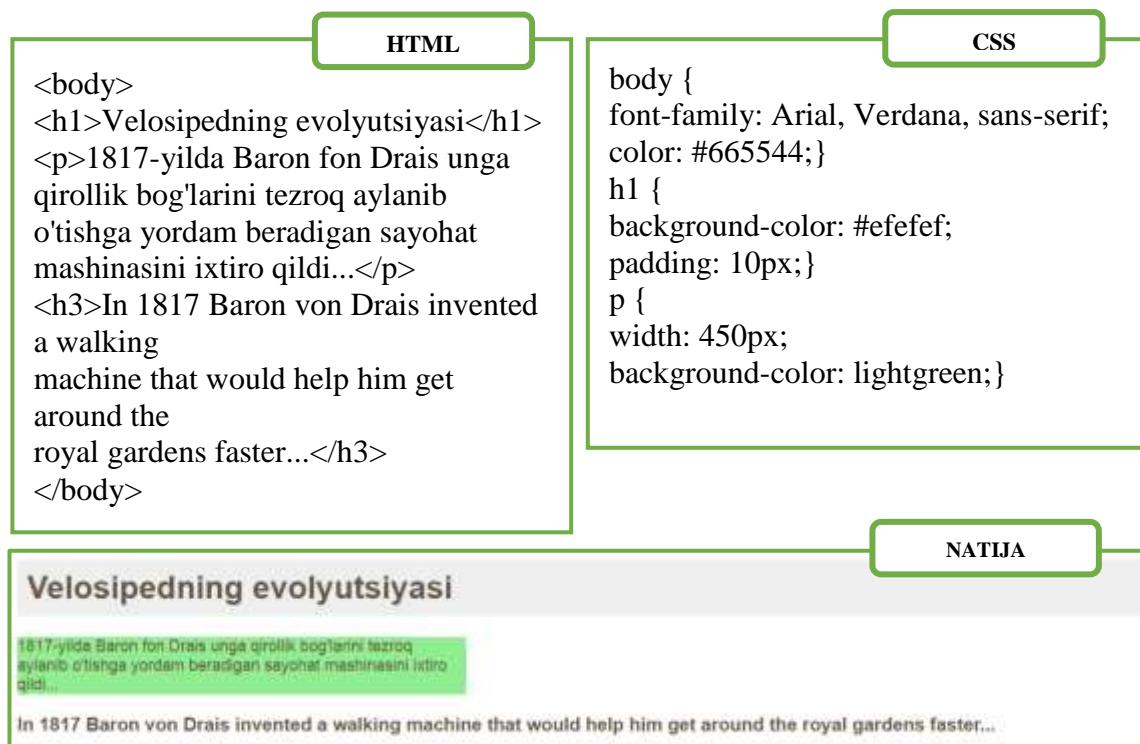
Oddiy oqimda har bir blok darajasidagi element keyingi elementning tepasida paydo bo‘ladi. Chunki bu brauzerlar uchun odatiy usul.

HTML elementlari uchun elementlar normal oqimda paydo bo‘lishi kerakligini ko‘rsatish uchun sizga CSS xususiyati kerak bo‘lmaseda, sintaksis quyidagicha bo‘ladi:

- ***position: static;***

Misolda, men sarlavha elementi uchun kenglik xususiyatini(width) ko‘rsatmadim, shuning uchun u standart butun brauzer oynasining kengligini qanday egallasini ko‘rishingiz mumkin.

Paragraflar kengligi 450 piksel bilan cheklangan. Bu oddiy oqimdagি elementlar brauzer oynasining to‘liq kengligini egallamasa ham, yangi satrda qanday boshlanishini ko‘rsatadi.



6.12-rasm. Bloklar joylashuvi. Normal oqim

Nisbiy pozitsiya(Relative Positioning)

Nisbiy joylashish elementni normal oqimda bo‘lgan joyga nisbatan harakatlantiradi. Masalan, siz uni oddiy oqimdagidan 10 piksel pastroq yoki 20% o‘ngga siljitingiz mumkin.

Keyin elementni normal oqimda bo'lgan joydan qanchalik uzoqqa ko'chirishni ko'rsatish uchun ofset xususiyatlaridan (yuqori yoki pastki va chap yoki o'ng) foydalaniladi.

Qutini gorizontal ravishda siljитish quyidagi hususiyatlar yordamida beriladi:

- left – blokni chap chegarasi, konteynerni chap chegarasiga nisbatan suriladi;
- right – blokni o'ng chegarasi, konteynerni o'ng chegarasiga nisbatan suriladi;
- top – blokni yuqori chegarasi, konteynerni yuqori chegarasiga nisbatan suriladi;
- bottom – blokni quyi chegarasi, konteynerni quyi chegarasiga nisbatan suriladi.

Box ofset xususiyatlarining qiymatlari odatda piksel, foiz yoki ems ko'rinishida beriladi.

Qiymatlар uzunlik o'lchov birligida yoki protsentlarda beriladi.

HTML

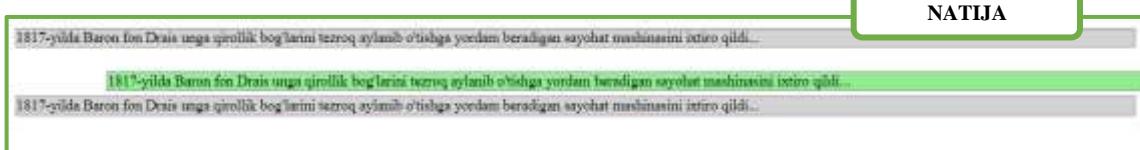
```
<body>
<p>1817-yilda Baron fon Drais unga
qirollik bog'larini tezroq aylanib
o'tishga yordam beradigan sayohat
mashinasini ixtiro qildi...</p>
<p class="example">1817-yilda Baron
fon Drais unga qirollik bog'larini tezroq
aylanib o'tishga yordam beradigan
sayohat mashinasini ixtiro qildi...</p>

<p>1817-yilda Baron fon Drais unga
qirollik bog'larini tezroq aylanib
o'tishga yordam beradigan sayohat
mashinasini ixtiro qildi...
</p>
</body>
```

CSS

```
p{
background-color: lightgrey;
border:solid 1px grey;
}
p.example {
position: relative;
top: 10px;
left: 100px;
background-color: lightgreen;
border:solid 1px grey;
}
```

NATIJA



6.13-rasm. Bloklar joylashuvi. Nisbiy positsiya

Absolute Positioning(Absolyut pozitsiyalash)

position: absolute;

Pozitsiya xususiyatiga qiymat berilganda, quti normal oqimdan chiqariladi va endi sahifadagi boshqa elementlarning holatiga ta'sir qilmaydi. (Ular u erda yo'qdek harakat qilishadi.)

Qutidagi offset xususiyatlari (yuqori yoki pastki va chap yoki o'ng) elementni o'z ichiga olgan elementga nisbatan qayerda ko'rinishi kerakligini belgilaydi. Bunda ham bloklarni surish quyidagi **qiymatlardan** birini oladi:

- *left;*
- *right;*
- *top;*
- *bottom.*

Ushbu misolda sarlavha sahifaning yuqori qismida va chap chetidan 500 piksel masofada joylashgan. Sarlavhaning kengligi 250 piksel kengligida o'rnatiladi. Ushbu misoldagi `<p>` elementlariga kenglik xususiyati ham matnning bir-biriga yopishib qolmasligi va o'qib bo'lmaydigan holga kelishining oldini olish uchun qo'llanilgan.

Odatiy bo'lib, ko'pchilik brauzerlar `<h1>` elementining yuqori qismiga chekka qo'shadilar. Shuning uchun brauzerning yuqori qismi va `<h1>` elementini o'z ichiga olgan quti o'rtasida bo'shliq mavjud. Agar siz ushbu chegarani olib tashlamoqchi bo'lsangiz, `<h1>` elementining uslub qoidalariga quyidagi kodni qo'shishingiz mumkin:

`margin: 0px;`

HTML

```
<body>
<h1>The Evolution of the Bicycle</h1>
<p>1817-yilda Baron fon Drais unga qirollik bog'larini tezroq aylanib o'tishga
yordam beradigan sayohat mashinasini ixtiro qildi...</p>
<p class="example">1817-yilda Baron fon Drais unga qirollik bog'larini tezroq
aylanib o'tishga yordam beradigan sayohat mashinasini ixtiro qildi...</p>
<p>1817-yilda Baron fon Drais unga qirollik bog'larini tezroq aylanib o'tishga
yordam beradigan sayohat mashinasini ixtiro qildi...
</p>
```

CSS

```
<style type="text/css">
    h1
    {
        position: absolute;
        top: 0px;
        left: 500px;
        width: 250px;
        background-color: lightgreen;
    }
    p
    {
        width: 400px;
        background-color: yellow;
    }
</style>
```

1817-yilda Baron fon Drais unga qirollik bog'larini tezroq aylanib o'tishga yordam beradigan sayohat mashinasini ixtiro qildi...

1817-yilda Baron fon Drais unga qirollik bog'larini tezroq aylanib o'tishga yordam beradigan sayohat mashinasini ixtiro qildi...

1817-yilda Baron fon Drais unga qirollik bog'larini tezroq aylanib o'tishga yordam beradigan sayohat mashinasini ixtiro qildi...

NATIJA

The Evolution of the Bicycle

6.14-rasm. Bloklar joylashuvi. Nisbiy positsiya

Fixed Positioning(Fiksirlangan bloklar)

position:fixed

Fiksirlangan blok hech qayerga surilmaydi va har doim o'rnatilgan sohada o'zgarishsiz turadi. U elementni brauzer oynasiga nisbatan joylashtiradi. Shuning uchun, foydalanuvchi sahifani pastga aylantirganda, u o'z joyida qoladi.

Natijani ko'rish uchun ushbu misolni brauzerda sinab ko'ramiz.

Brauzer oynasiga nisbatan belgilangan joylashuv oynasi qayerda paydo bo'lishini nazorat qilish uchun qutining offset xususiyatlaridan foydalilanadi. Ushbu misolda sarlavha brauzer oynasining yuqori chap burchagiga joylashtirilgan. Foydalanuvchi sahifani pastga aylantirganda, paragraflar sarlavha orqasida qoladi.

`<p>` elementlari normal oqimda va `<h1>` elementi egallagan joyda turibti. Shuning uchun, margin-top xususiyatini `<h1>` elementidan keyingi `<p>` elementni pastga surish uchun ishlataligancha.

HTML

```
<h1>The Evolution of the Bicycle</h1>
<p class="example">1817-yilda Baron fon Drais unga qirolik bog'larini tezroq
aylanib o'tishga yordam beradigan sayohat mashinasini ixtiro qildi...</p>
<p>1817-yilda Baron fon Drais unga qirolik bog'larini tezroq aylanib o'tishga
yordam beradigan sayohat mashinasini ixtiro qildi...</p>
<p>1817-yilda Baron fon Drais unga qirolik bog'larini tezroq aylanib o'tishga
yordam beradigan sayohat mashinasini ixtiro qildi..
.</p>
```

CSS

```
<style type="text/css">
  h1 {
    position: fixed;
    top: 0px;
    left: 50px;
    padding: 10px;
    margin: 0px;
    width: 100%;
    background-color: #efefef;
  }
  p.example {
    margin-top: 100px;
  }
</style>
```

NATIJA

The Evolution of the Bicycle

1817-yilda Baron fon Drais unga qirolik bog'larini tezroq aylanib o'tishga yordam beradigan sayohat mashinasini ixtiro qildi...

1817-yilda Baron fon Drais unga qirolik bog'larini tezroq aylanib o'tishga yordam beradigan sayohat mashinasini ixtiro qildi...

1817-yilda Baron fon Drais unga qirolik bog'larini tezroq aylanib o'tishga yordam beradigan sayohat mashinasini ixtiro qildi...

6.15-rasm. Bloklar joylashuvi. Fiksirlangan bloklar

Overlapping Element(z-index)

Relative, fixed yoki absolute joylashishni qo'llaganingizda, qutilar bir-birining ustiga chiqishi mumkin. Agar katakchalar bir-birining ustiga chiqsa, HTML kodida keyinroq paydo bo'ladigan elementlar sahifaning oldingi qismining tepasida joylashadi. Qaysi element tepada turishini nazorat qilmoqchi bo'lsangiz, z-index

xususiyatidan foydalanishingiz mumkin. Uning qiymati raqam bo‘lib, raqam qanchalik baland bo‘lsa, bu element old tomonga yaqinroq bo‘ladi. Masalan, z indeksi 10 bo‘lgan element z indeksi 5 bo‘lgan elementning tepasida paydo bo‘ladi.

Ushbu misol <p> elementlari uchun nisbiy joylashishni qo‘llaydi. Paragraflar nisbatan joylashtirilganligi sababli, foydalanuvchi sahifani pastga aylantirganda, ular odat bo‘yicha sarlavhaning tepasida paydo bo‘ladi. <h1> elementi yuqorida turishini ta’minlash uchun biz <h1> elementi uchun qoidadagi z-index xususiyatidan foydalananamiz.

HTML

```
<h1>Velosipedning evolyutsiyasi</h1>
<p>1817-yilda Baron fon Drais unga qirolik bog'lar bo'ylab tezroq aylanib o'tishga yordam beradigan piyoda mashinasini ixtiro qildi: ikkita bir xil o'lchamdagи qatorli g'ildiraklar, oldingisi boshqariladigan, siz o'tirgan ramkaga o'rnatilgan. Qurilma oyoqlaringizni yerga surish orqali harakatlantirildi va shu tariqa o'zingizni va qurilmani qandaydir sirpanish yurishida oldinga aylantiring.</p>
<p>Mashina Draisienne (yoki "xobbi ot") nomi bilan mashhur bo'ldi. U butunlay yog'ochdan qilingan. Bu moda bo'lib qisqa vaqt davomida mashhur bo'ldi, chunki park yoki bog' kabi yaxshi saqlangan yo'ldan boshqa joyda tashish uchun qulay emas.</p>
```

CSS

```
body { width: 750px;
        font-family: Arial, Verdana, sans-serif;
        color: #665544;}
h1 { position: fixed;
      top: 0px;
      left: 0px;
      margin: 0px;
      padding: 10px;
      width: 100%;
      background-color: #efefef;}
p { position: relative;
      top: 70px; left: 70px;}
```

NATIJA

1817-yilda Baron fon Drais unga qirolik bog'lar bo'ylab tezroq aylanib o'tishga yordam beradigan piyoda mashinasini ixtiro qildi: ikkita bir xil o'lchamdagи qatorli g'ildiraklar, oldingisi boshqariladigan, siz o'tirgan ramkaga o'rnatilgan. Qurilma oyoqlaringizni yerga surish orqali harakatlantirildi va shu tariqa o'zingizni va qurilmani qandaydir sirpanish yurishida oldinga aylantiring.

Mashina Draisienne (yoki "xobbi ot") nomi bilan mashhur bo'ldi. U butunlay yog'ochdan qilingan. Bu moda bo'lib qisqa vaqt davomida mashhur bo'ldi, chunki park yoki bog' kabi yaxshi saqlangan yo'ldan boshqa joyda tashish uchun qulay emas.

Ikkı g'ildirakli minish mashinasining keyingi ko'rinishi 1865 yilda, pedallar to'g'ridan-to'g'ri old g'ildirakka qo'yilganda paydo bo'lgan. Ushbu mashina velosiped ("tezkor oyoq" degan ma'noni anglatadi) va "suyak silkituvchi" nomi bilan tanilgan, chunki u yog'och konstruktisiyasi bilan o'sha davrdagi tosh yo'llar bilan juda noqulay sayolot qilish uchun yaratilgan. Ular, shuningdek, modaga aylandi va katta shaharlarda rolikli maydonchalarga o'shash yopiq sport akademiyalarini topish mumkin edi.

1870 yilda birinchi to'liq metall mashina paydo bo'ldi. (Bundan oldin metallurgiya kichik va engil qismlarni yasash uchun etarlichcha mustahkam bo'lgan metall bilan ts'uminish uchun etarlichcha rivojlangan emas edi.) Pedallar to'g'ridan-to'g'ri old g'ildirakka erkin aylanish mekanizmisisiz burikturilgan. Qattiq rezina shinalar va katta old g'ildirakning uzun turgaklari oldingisiga qaraganda ancha silliq yurish imkonini berdi.

Old g'ildiraklar kattalashib, kattalashdi, chunki ishlab chiqaruvchilar g'ildirak qanchalik katta bo'lsa, pedallarning bir marta aylanishi bilan shunchalik uzoqqa borish imkonligini tushuniishi. Shu sababli, siz oyog'ingiz uzunligi imkon beradigan diusajada katta g'ildirak so'tib olasiz. Bu mashina birinchi bo'lib velosiped ("ikkı g'ildirak") deb atalgan. Ushbu velosipedlar 1880-yillarda yosh yigitlar orasida insha mashhur hol'dan. (Ular e'tibariga ishechinining otti odilik maoshiga ta'ovi kaladi.)

6.16-rasm. Bloklar joylashuvi. Z-indeks qo‘yilmagan holat

<h1>Velosipedning evolyutsiyasi</h1>
 <p>1817-yilda Baron fon Drais unga qirollik bog'lari bo'ylab tezroq aylanib o'tishga yordam beradigan piyoda mashinasini ixtiro qildi: ikkita bir xil o'lchamdag'i qatorli g'ildiraklar, oldingisi boshqariladigan, siz o'tirgan ramkaga o'rnatilgan. Qurilma oyoqlaringizni yerga surish orqali harakatlantirildi va shu tariqa o'zingizni va qurilmani qandaydir sirpanish yurishida oldinga aylantiring.</p>
 <p>Mashina Draisienne (yoki "xobbi ot") nomi bilan mashhur bo'ldi. U butunlay yog'ochdan qilingan. Bu moda bo'lib qisqa vaqt davomida mashhur bo'ldi, chunki park yoki bog' kabi yaxshi saqlangan yo'ldan boshqa joyda tashish uchun qulay emas.</p>

CSS

```
h1 {  
position: fixed;  
top: 0px;  
left: 0px;  
margin: 0px;  
padding: 10px;  
width: 100%;  
background-color: #efefef;  
z-index: 10;}  
  
p {  
position: relative;  
top: 70px;  
left: 70px;}
```

NATIJA



6.17-rasm. Bloklar joylashuvi. Z-indeks qo'llanilgan holat

Floating Elements

Float xususiyati oddiy oqimdag'i elementni olish va uni o'z ichiga olgan elementdan iloji boricha chap yoki o'ng tomonga joylashtirish imkonini beradi.

Matn esa uning atrofidan aylanib o'tadi. Matnning bunday harakati toshni yonidan oqib o'tuvchi suvni eslatadi, shuning uchun ham bunday hususiyatli bloklar, suzuvchi blok deyiladi. Float xususiyatidan foydalanganda, suzuvchi element qanchalik keng bo'lishi kerakligini ko'rsatish uchun width xususiyatidan ham foydalanish kerak.

Bu misolda qo'shtirnoqni ushlab turish uchun <blockquote> elementi qo'llaniladi. U o'z ichiga olgan element <body> elementidir.

<blockquote> elementi o'ngga suriladi va qo'shtirnoqdan keyingi paragraflar suzuvchi element atrofida aylanadi.

HTML

```
<body> <h1>Velosipedning evolyutsiyasi</h1>
<blockquote>"Hayot velosiped haydashga o'xshaydi. Muvozanatni saqlash uchun
harakatda davom etish kerak." - Albert Eynshteyn</blockquote>
<p>Mashina Draisienne (yoki "xobbi ot") nomi bilan mashhur bo'ldi. U butunlay
yog'ochdan qilingan. Bu moda bo'lib qisqa vaqt davomida mashhur bo'ldi, chunki
park yoki bog' kabi yaxshi saqlangan yo'lidan boshqa joyda tashish uchun qulay
emas.</p>
<p>Ikki g'ildirakli minish mashinasining keyingi ko'rinishi 1865 yilda, pedallar
to'g'ridan-to'g'ri old g'ildirakka qo'yilganda paydo bo'lgan. Ushbu mashina velosiped
("tezkor oyoq" degan ma'noni anglatadi) va "suyak silkituvchi" nomi bilan tanilgan,
chunki u yog'och konstruktsiyasi bilan o'sha davrdagi tosh yo'llar bilan juda noqulay
sayohat qilish uchun yaratilgan. Ular, shuningdek, modaga aylandi va katta
shaharlarda rolikli maydonchalarga o'xshash yopiq sport akademiyalarini topish
mumkin edi.</p> </body>
```

CSS

```
blockquote {
float: right;
width: 275px;
font-size: 130%;
font-style: italic;
font-family: Georgia, Times,
serif;
margin: 0px 0px 10px 10px;
padding: 10px;
border-top: 1px solid #665544;
border-bottom: 1px solid
#665544;}
```

NATIJA

Velosipedning evolyutsiyasi

Mashina Draisienne (yoki "xobbi ot") nomi bilan mashhur bo'ldi. U butunlay yog'ochdan
qilingan. Bu moda bo'lib qisqa vaqt davomida mashhur bo'ldi, chunki bog' kabi
yaxshi saqlangan yo'lidan boshqa joyda tashish uchun qulay emas.

Ikkii g'ildirakli minish mashinasining keyingi ko'rinishi 1865 yilda, pedallar to'g'ridan-
to'g'ri old g'ildirakka qo'yilganda paydo bo'lgan. Ushbu mashina velosiped ("tezkor oyoq"
degen ma'noni anglatadi) va "suyak silkituvchi" nomi bilan tanilgan, chunki u yog'och
konstruktsiyasini bilan o'sha davrdagi tosh yo'llar bilan juda noqulay sirsat qilish uchun
yaratilgan. Ular, shuningdek, modaga aylandi va katta shaharlarda rolikli maydonchalarga
o'chadan yopiq sport akademiyalarini topishi mumkin edi.

"Hayot velosiped haydashga
o'xshaydi. Muvozanatni
saqlash uchun harakatda
davom etish kerak." - Albert
Eynshteyn

1870 yilda birinchi to'liq metall masdon paydo bo'ldi. (Bundan oldin installerga kichik va engil qurilumi yaxshi ochna surʼishcha
molibden bo'lgan metall bilan temʼish tutish etadidagi murojiga ensiz eidi.) Pedallar to'g'ridan-to'g'ri old g'ildirakka erkin yulash
meʼmoriyati binkinligiga. Qotiq rezin shunalari va katta old g'ildirakka ostida qolningiga qiraziga ancha sifʼiq yutʼish
mumkin berdi.

6.18-rasm. Bloklar joylashuvi. Float xususiyatidan foydalanish

Elementlarni yonma-yon joylashtirish uchun Floatdan foydalanish

Ko'pgina maketlarda qutilar bir-birining yonida joylashadi. Bunga erishish uchun odatda float xususiyatidan foydalaniladi.

Ushbu misolda oltita paragrafni ko'rishingiz mumkin, ularning har biri kenglik va float xususiyatlari to'plamiga ega.

To'rtinchchi xatboshi, kutilgandek, sahifaning chap chetiga o'tmaydi.

Aksincha, u uchinchi xatboshi ostida joylashadi. Buning sababi shundaki, to'rtinchchi xatboshida uchinchi xatboshi ostidan boshlash uchun joy bor, lekin u chapga o'ta olmaydi, chunki ikkinchi xatboshi balandligi yuqori. Xullas, joylashish chapdan o'ngga va qaytishda o'ngdan chapga harakatlanadi, ammo o'ngdan chapga harakatlanganda balandligi eng past bo'lgan element ostidan harakatlanadi.

Float quyidagi qiymatlardan birini qabul qilishi mumkin.

- ✓ **Left** strukturali blok chapga suriladi, hujjatning qolgan qismi esa blokning o‘ng tomoniga o‘tadi;
- ✓ **Right** strukturali blok o‘ng tomonga suriladi, hujjatning qolgan qismi esa blokning chap tomoniga o‘tadi;
- ✓ **none** blok surilmaydi, ya’ni positionda ko‘rsatilgan algoritm bo‘yicha joylashadi. Agar position hususiyati ko‘rsatilmasa, normal oqim holiday bo‘ladi;
- ✓ **inherit** o‘zi joylashgan konteynerni qiymatini oladi.

HTML

```
<body>
<h1>Velosipedning evolyutsiyasi</h1>
<p>1.Mashina Draisienne (yoki "xobbi ot") nomi bilan mashhur bo'ldi. U butunlay
yog'ochdan qilingan. Bu moda bo'lib qisqa vaqt davomida mashhur bo'ldi, chunki
park yoki bog' kabi yaxshi saqlangan yo'lidan boshqa joyda tashish uchun qulay
emas.</p>
<p>2.Ikki g'ildirakli minish mashinasining keyingi ko'rinishi 1865 yilda, pedallar
to'g'ridan-to'g'ri old g'ildirakka qo'yilganda paydo bo'lган. Ushbu mashina velosiped
("tezkor oyoq" degan ma'noni anglatadi) va "suyak silkituvchi" nomi bilan tanilgan,
chunki u yog'och konstruktsiyasi bilan o'sha davrdagi tosh yo'llar bilan juda noqulay
sayohat qilish uchun yaratilgan. Ular, shuningdek, modaga aylandi va katta
shaharlarda rolikli maydonchalarga o'xshash yopiq sport akademiyalarini topish
mumkin edi.</p>
<p>3.1870 yilda birinchi to'liq metall mashina paydo bo'ldi. (Bundan oldin
metallurgiya kichik va engil qismlarni yasash uchun etarlicha mustahkam bo'lган
metall bilan ta'minlash uchun etarlicha rivojlangan emas edi.) Pedallar to'g'ridan-
to'g'ri old g'ildirakka erkin aylanish mexanizmisiz biriktirilgan. Qattiq rezina shinalar
va katta old g'ildirakning uzun tirkaklari oldingisiga qaraganda ancha silliq yurish
imkonini berdi.</p>
<p>4.Old g'ildiraklar kattalashib, kattalashdi, chunki ishlab chiqaruvchilar g'ildirak
qanchalik katta bo'lsa, pedallarning bir marta aylanishi bilan shunchalik uzoqqa
borish mumkinligini tushunishdi. Shu sababli, siz oyog'ingiz uzunligi imkon
beradigan darajada katta g'ildirak sotib olasiz. Bu mashina birinchi bo'lib velosiped
("ikki g'ildirak") deb atalgan. Ushbu velosipedlar 1880-yillarda yosh yigitlar orasida
juda mashhur bo'lган. (Ular o'rtacha ishchining olti oylik maoshiga to'g'ri
keladi.)</p>
<p>5.Chavvand og'irlilik markazidan juda baland o'tirgani uchun, agar old g'ildirakni
yo'lida tosh yoki yo'l-yo'lakay to'xtatib qo'ygan bo'lsa yoki to'satdan it paydo
bo'lsa, butun qurilma oldingi o'qda oldinga burilib, chavandoz , oyoqlari rul ostida
qolib, boshiga tantanali ravishda tushirildi. Shu tariqa “sarlavha olish” atamasi
vujudga keldi.</p>
<p>
6. </p></body>
```

```

body {
width: 750px;
font-family: Arial, Verdana, sans-serif;
color: #665544; }

p {
width: 230px;
float: left;
margin: 5px;
padding: 5px;
background-color: #efefef; }

```

Velosipedning evolyutsiyasi

1.Mashina Draislenne (yoki "xobbi of") nomi bilan mashhur bo'ldi. U butunlay yog'ochdan qilingan. Bu moda bo'lib qisqa vaqt davomida mashhur bo'ldi, chunki park yoki bog' kabi yaxshi saqlangan yoldan boenqa joyda tashish uchun quay emas.

2.ikki g'ildirakli minish mashinasining keyingi ko'rinishi 1855 yilda, pedallar to'g'ridan-to'g'ri old g'ildirakka qo'yilganda paydo bo'lgan. Ushbu mashina velosiped ("tezkor oyog" degan ma'noni anglatadi) va "suyak silkituvchi" nomi bilan tanilgan, chunki u yog'och konstruktelyasi bilan o'sha davrdagi tosh yo'llar bilan juda noqulay sayohat qilish uchun yaratilgan. Ular, shuningdek, modaga aylandi va katta shaharlarda ro'likl maydonchalarga o'xshash yopiq sport akademiyalarini topish mumkin edi.

3.1870 yilda birinchi to'lq metall mashina paydo bo'ldi. (Bundan oldin metallurgiya kichik va engil qismi yasaish uchun etaricha mustahkam bo'lgan metall bilan ta'minlash uchun etaricha rivojlangan emas edi.) Pedallar to'g'ridan-to'g'ri old g'ildirakka erkin aylanish mexanizmisi birkotirilgan. Qattiq rezina shhnalar va katta old g'ildirakning uzun tingaklari oldingisiga qaraganda ancha silliq yunish imkonini berdi.

4.Old g'ildiraklar kattalashib, kattalashdi, chunki ishab chiqaruvchilar g'ildirak qanchalik katta bo'lsa, pedallarning bir marta aylanishi bilan shunchalik uzoqqa borish mumkinligini tushunishdi. Shu sababli, siz oyog'ingiz uzunligi imkon beradigan darajada katta g'ildirak etib olasiz. Bu mashina birinchi bo'lib velosiped ("ikki g'ildirak") deb atalgan. Ushbu velosipedlar 1880-yillarda yosh yigitlar orasida juda maehhur bo'lgan. (Ular o'ttacha ishchining otti oylik masehiga to'g'it keladi.)

5.Chavvarid og'irlik markazidan juda baland o'tirgani uchun, agar old g'ildirakni yo'da tosh yoki yo'l-yo'lakay to'xtatib qo'yan bo'ssa yoki to'satdan it paydo bo'lea, butun qurilma oldingi o'rqa oldinga burilib, chavandoz , oyolgari rui ostida qolib, boshiga tantanali ravishda tushirildi. Shu tarqa "sarlavha olish" atamasini vujudga keldi.

6.

6.19-rasm. Bloklar joylashuvi. Elementlarni yonma-yon joylashtirish

Paragraflar balandligini eng baland paragraf bilan bir xil balandlikda belgilash bu muammoni hal qiladi, lekin u kamdan-kam hollarda mos keladi.

Ushbu muammoni hal qilish uchun clear xususiyatdan foydalilanadi va u quyidagi qiymatlarni oladi:

- Left;
- Right;
- Both;
- None.

Ushbu misolda, to‘rtinchi xatboshida clear deb nomlangan sinf mavjud. Ushbu sinf uchun CSS qoidasi uning chap tomoniga hech narsa tegmasligini ko‘rsatish uchun clear xususiyatdan foydalanadi. Shuning uchun to‘rtinchi xatboshi sahifaning chap tomoniga boshqa hech qanday element tegmasligi uchun pastga siljiydi.

HTML

```
<body>
<h1>Velosipedning evolyutsiyasi</h1>
<p>1.Mashina Draisienne (yoki "xobbi ot") nomi bilan mashhur bo'ldi. U butunlay
yog'ochdan qilingan. Bu moda bo'lib qisqa vaqt davomida mashhur bo'ldi, chunki
park yoki bog' kabi yaxshi saqlangan yo'lidan boshqa joyda tashish uchun qulay
emas.</p>
<p>2.Ikki g'ildirakli minish mashinasining keyingi ko'rinishi 1865 yilda, pedallar
to'g'ridan-to'g'ri old g'ildirakka qo'yilganda paydo bo'lgan. Ushbu mashina velosiped
("tezkor oyoq" degan ma'noni anglatadi) va "suyak silkituvchi" nomi bilan tanilgan,
chunki u yog'och konstruktsiyasi bilan o'sha davrdagi tosh yo'llar bilan juda noqulay
sayohat qilish uchun yaratilgan. Ular, shuningdek, modaga aylandi va katta
shaharlarda rolikli maydonchalarga o'xshash yopiq sport akademiyalarini topish
mumkin edi.</p>
<p>3.1870 yilda birinchi to'liq metall mashina paydo bo'ldi. (Bundan oldin
metallurgiya kichik va engil qismlarni yasash uchun etarlicha mustahkam bo'lgan
metall bilan ta'minlash uchun etarlicha rivojlangan emas edi.) Pedallar to'g'ridan-
to'g'ri old g'ildirakka erkin aylanish mexanizmisiz biriktirilgan. Qattiq rezina shinalar
va katta old g'ildirakning uzun tirkaklari oldingisisiga qaraganda ancha silliq yurish
imkonini berdi.</p>
<p class=“clear”>4.Old g'ildiraklar kattalashib, kattalashdi, chunki ishlab
chiqaruvchilar g'ildirak qanchalik katta bo'lsa, pedallarning bir marta aylanishi bilan
shunchalik uzoqqa borish mumkinligini tushunishdi. Shu sababli, siz oyog'ingiz
uzunligi imkon beradigan darajada katta g'ildirak sotib olasiz. Bu mashina birinchi
bo'lib velosiped ("ikki g'ildirak") deb atalgan. Ushbu velosipedlar 1880-yillarda yosh
yigitlar orasida juda mashhur bo'lgan. (Ular o'rtacha ishchining olti oylik maoshiga
to'g'ri keladi.)</p>
<p>5.Chavvand og‘irlik markazidan juda baland o‘tirgani uchun, agar old g'ildirakni
yo‘lda tosh yoki yo‘l-yo‘lakay to‘xtatib qo‘ygan bo‘lsa yoki to‘satdan it paydo
bo‘lsa, butun qurilma oldingi o‘qda oldinga burilib, chavandoz , oyoqlari rul ostida
qolib, boshiga tantanali ravishda tushirildi. Shu tariqa “sarlavha olish” atamasi
vujudga keldi.</p> <p> 6. </p> </body>
```

css

```
body { width: 750px;  
font-family: Arial, Verdana, sans-serif;  
color: #665544;}  
p {  
width: 230px;  
float: left;  
margin: 5px;  
padding: 5px;  
background-color: #efefef;}  
.clear {  
clear: left;}
```

NATIJA

Velosipedning evolyutsiyasi

1. Mashina Draisenne (yoki "xobbi or") nomi bilan mashhur bo'ldi. U butunlay yog'ochdan qilingan. Bu moda bol'so qisqa vaqt davomida mashhur bo'ldi, chunki park yoki bog' kabli yaxshi joyda sojangan yoldan bo'eqsa joyda tashish uchun quay emas.

2. ikki gildirakchi minish mashinasining keyingi korinishi 1855 yilda, pedallar tog'idan tog'ini oldi gildirakka qoyilganda paydo bolgan. Ushbu mashina velosiped ("tezkor oyod") degan man'oni anglatadi va "suyak silkituvchini" nomi bilan tanilgan, chunki u yog'och konstruktivasi bilan o'sha davridagi tosi yollar bilan juda noqslay sayohat qilish uchun shart bo'lgan. Utar, shuningdek, mozaik ayaridi va katta ehanchanida rolik maydonchalalarga o'xshash yopiq sport akademiyalarini topish mumkin edi.

3. 1870 yilda birinchi to'liq metalli mashina paydo bo'ldi. (Bundan oldin metallurgiya kishik va engil qismilarni yasaesh uchun etarlicha mustahkam bolgan metall bilan ta'minlash uchun etarlicha rivoljangan emas edi.) Pedallar tog'idan-tog'ini oldi gildirakka erkin aylanish mekanizmlari binkiltingan. Qatliz rezoora shinhatar va katta old gildirakning uzun tirkiklarini ololingibiga qaranganda ancha siliq yurish imkonini berdi.

4. Old gildiraklar kattalashib, kattalashib, chunki lehib chiqaruvchilar gildirak qanchalik katta bo'lsa, pedallaring bir marta aylanishi bilan shunchalik uzqaqa borish mumkinligi tuerishdi. Shu sababli, siz oyodning usmonligiga boz qo'shishga darajada katta gildirak sotib olasiz. Bu mashina birinchi bol'ib velosiped ("ikki gildirak") deb atalgan. Ushbu velosipedlar 1850-yillarda yosh yigitlar orasida juda mashhur bolgan. (Ular o'rtacha ishnching olti oylik maoshiga tog'ini keladi).

5. Chavandagi og'irlik markazidan juda baland o'tirgani uchun, agar old gildirakni yolda tosh yoki yo'l-yotakay to'satib qo'ygan bo'lsa yoki to'satidan it paydo bo'lsa, butun qurima o'sha yoki to'satidan it paydo bo'lsa, chavandagi oyotgan nul ostida qolli, bosqiga tantarsali rivojsha tushintid. Shu tarqa "saralvra olish" atamasini yujudga heldi.

6.

6.20-rasm. Bloklar joylashuvi. Elementlarni yonma-yon joylashtirishda clear xususiyati

Float yordamida ko'p ustunli maketlarni yaratish

Ko'pgina web-sahifalar o'z dizaynlarida bir nechta ustunlardan foydalanadilar. Bunga har bir ustunni ifodalash uchun `<div>` elementidan foydalanish orqali erishiladi. Ustunlarni yonma-yon joylashtirish uchun quyidagi uchta CSS xususiyatidan foydalaniladi:

- width – bu ustunlar kengligini o'rnatadi;
- float – bu ustunlarni bir-birining yonida joylashtiradi;
- margin – bu ustunlar orasidagi bo'shliqni yaratadi.

<div> elementlarining har birida sarlavhalar, paragraflar, rasmlar va hatto boshqa <div> elementlari bo‘lishi mumkin.

Ushbu sahifada ko‘rsatilganidek, uchta <div> elementni bir-birining yonida float orqali uchta ustunli tartib yaratish mumkin.

HTML

```
<body>
<h1>Velosipedning evolyutsiyasi</h1>
<div class="column1of3">
<p>1.Mashina Draisienne (yoki "xobbi ot") nomi bilan mashhur bo'ldi. U butunlay yog'ochdan qilingan. Bu moda bo'lib qisqa vaqt davomida mashhur bo'ldi, chunki park yoki bog' kabi yaxshi saqlangan yo'ldan boshqa joyda tashish uchun qulay emas.</p></div>
<div class="column2of3">
<p>2.Ikki g'ildirakli minish mashinasining keyingi ko'rinishi 1865 yilda, pedallar to'g'ridan-to'g'ri old g'ildirakka qo'yilganda paydo bo'lgan. Ushbu mashina velosiped ("tezkor oyoq" degan ma'noni anglatadi) va "suyak silkituvchi" nomi bilan tanilgan, chunki u yog'och konstruktsiyasi bilan o'sha davrdagi tosh yo'llar bilan juda noqulay sayohat qilish uchun yaratilgan. Ular, shuningdek, modaga aylandi va katta shaharlarda rolikli maydonchalarga o'xshash yopiq sport akademiyalarini topish mumkin edi.</p></div>
<div class="column2of3">
<p>3.1870 yilda birinchi to'liq metall mashina paydo bo'ldi. (Bundan oldin metallurgiya kichik va engil qismlarni yasash uchun etarlicha mustahkam bo'lgan metall bilan ta'minlash uchun etarlicha rivojlangan emas edi.) Pedallar to'g'ridan-to'g'ri old g'ildirakka erkin aylanish mexanizmisisiz biriktirilgan. Qattiq rezina shinalar va katta old g'ildirakning uzun tirkaklari oldingisiga qaraganda ancha silliq yurish imkonini berdi</p>
</div>
</body>
```

css

```
.column1of3, .column2of3,
.column3of3 {
width: 300px;
float: left;
margin: 10px;}
```

NATIJA

Velosipedning evolyutsiyasi

1.Mashina Draisienne (yoki "xobbi ot") nomi bilan mashhur bo'ldi. U butunlay yog'ochdan qilingan. Bu moda bo'lib qisqa vaqt davomida mashhur bo'ldi, chunki park yoki bog' kabi yaxshi saqlangan yo'ldan boshqa joyda tashish uchun qulay emas.

2.Ikki g'ildirakli minish mashinasining keyingi ko'rinishi 1865 yilda, pedallar to'g'ridan-to'g'ri old g'ildirakka qo'yilganda paydo bo'lgan. Ushbu mashina velosiped ("tezkor oyoq" degan ma'noni anglatadi) va "suyak silkituvchi" nomi bilan tanilgan, chunki u yog'och konstruktsiyasi bilan o'sha davrdagi tosh yo'llar bilan juda noqulay sayohat qilish uchun yaratilgan. Ular, shuningdek, modaga aylandi va katta shaharlarda rolikli maydonchalarga o'xshash yopiq sport akademiyalarini topish mumkin edi.

3.1870 yilda birinchi to'liq metall mashina paydo bo'ldi. (Bundan oldin metallurgiya kichik va engil qismlarni yasash uchun etarlicha mustahkam bo'lgan metall bilan ta'minlash uchun etarlicha rivojlangan emas edi.) Pedallar to'g'ridan-to'g'ri old g'ildirakka erkin aylanish mexanizmisisiz biriktirilgan. Qattiq rezina shinalar va katta old g'ildirakning uzun tirkaklari oldingisiga qaraganda ancha silliq yurish imkonini berdi

6.21-rasm. Ko‘p ustunli maketlarni yaratish

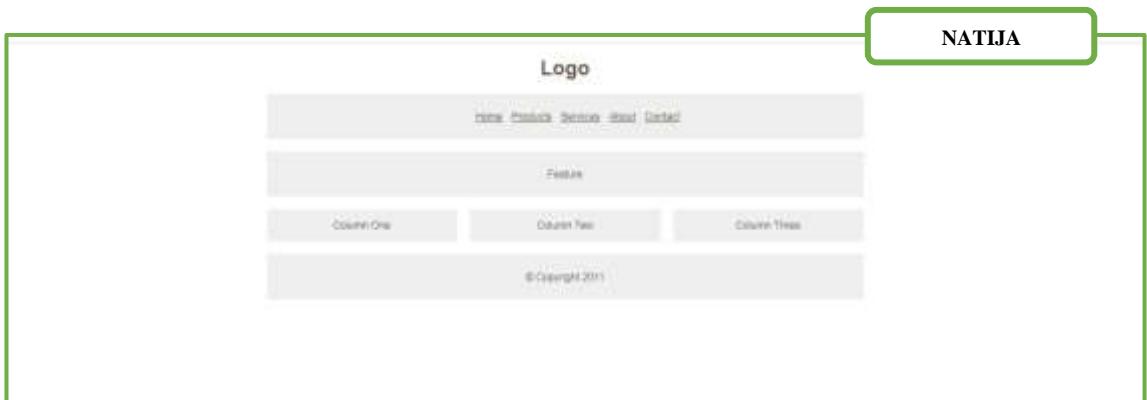
Maketlarga ruxsat etilgan kengliklar. Ruxsat etilgan kenglikdagi tartib dizaynlari foydalanuvchi o‘z brauzer oynasining o‘lchamini kattalashtirgani yoki kichraytirgani sababli hajmini o‘zgartirmaydi. O‘lchovlar odatda piksellarda beriladi.

HTML

```
<body>
    <div id="header">
        <h1>Logo</h1>
        <div id="nav">
            <ul>
                <li><a href="">Home</a></li>
                <li><a href="">Products</a></li>
                <li><a href="">Services</a></li>
                <li><a href="">About</a></li>
                <li><a href="">Contact</a></li>
            </ul>
        </div> </div>
    <div id="content">
        <div id="feature">
            <p>Feature</p>
        </div>
        <div class="article column1">
            <p>Column One</p>
        </div>
        <div class="article column2">
            <p>Column Two</p>
        </div>
        <div class="article column3">
            <p>Column Three</p>
        </div>
    <div id="footer">
        <p>&copy; Copyright 2011</p>
    </div>
</body>
```

css

```
* { font-family: Arial, Verdana, sans-serif;
    color: #665544;
    text-align: center; }
body { width: 960px;
      margin: 0 auto; }
#content { overflow: auto;
           height: 100%; }
#nav, #feature, #footer {
    background-color: #efefef;
    padding: 10px;
    margin: 10px; }
.column1, .column2, .column3 {
    background-color: #efefef;
    width: 300px;
    float: left;
    margin: 10px; }
li {
    display: inline;
    padding: 5px; }
```



6.22-rasm. Maketlarga ruxsat etilgan kengliklar

NAZORAT SAVOLLARI

- 6.1.Ro‘yxatlarga Cssning qaysi xossalari qo‘llash mumkin?
- 6.2.Jadvallarga ishvori berishda Cssning qaysi xossalari ishlataladi?
- 6.3.Jadvalga o‘lchamlar qaysi xususiyat orqali beriladi?
- 6.4.Jadvalning orqa qismiga rang qaysi xususiyat orqali beriladi?
- 6.5.Fieldsets & Legends elementiga qanday Css xossalari qo‘llasa bo‘ladi?
- 6.6.Blok nima?
- 6.7.Elementlarni qanday joylashtirish mumkin?
- 6.8.Odatda maketlarga qanday kengliklar o‘rnataladi?

VII-BOB. JAVASCRIPTGA KIRISH. MA'LUMOT TURLARI.

ARIFMETIK OPERATORLAR

JavaScript - bu interaktiv va dinamik veb-saytlarni yaratish uchun keng qo'llaniladigan dasturlash tili. U birinchi marta 1995-yilda Brendan Eich tomonidan ishlab chiqilgan va hozirda ECMAScript standartlari tomonidan qo'llab-quvvatlanadi. JavaScript barcha asosiy veb-brauzerlar tomonidan qo'llab-quvvatlanadigan yuqori darajali, talqin qilingan dasturlash tilidir.

JavaScript asosan veb-brauzerde ishlaydigan mijoz tomoni skriptlarini yaratish uchun ishlatiladi. Bu shuni anglatadiki, u serverda emas, balki foydalanuvchining kompyuterida bajariladi. Bu foydalanuvchi tajribasini tezroq va sezgirroq qilish imkonini beradi, chunki kod to‘g‘ridan-to‘g‘ri brauzerda serverga borishni talab qilmasdan ishlashi mumkin. Bundan tashqari, JavaScript-dan Node.js kabi texnologiyalar yordamida server tomonidagi skriptlarni yaratish uchun foydalanish mumkin.

JavaScriptning eng mashhur qo'llanilish sohalaridan biri bu interaktiv veb-sahifalarni yaratishdir. Masalan, JavaScriptdan ochiladigan menyular, tasvir slayderlarini yaratish va shaklni tekshirish uchun foydalanish mumkin. Bundan tashqari, animatsiyalar, qalqib chiquvchi oynalar va modal oynalar kabi dinamik effektlarni yaratish uchun ham foydalanish mumkin.

JavaScript odatda HTML va CSS kabi boshqa veb-texnologiyalar bilan birgalikda ishlatiladi. HTML veb-sahifaning tuzilishini ta’minlaydi, CSS esa maket va dizaynni yaratish uchun ishlatiladi. JavaScript veb-sahifaga interaktivlik va dinamik xatti-harakatlarni qo‘sish uchun ishlatilishi mumkin.

HTML da istalgan miqdorda JavaScript kodlarini joylashtirishimiz mumkin. JavaScript kodingiz HTML sahifani `<body></body>` hamda `<head></head>` teglaridan birida yoki bo'lmasam ikkalasida ham joylashtirish mumkin.

7.1. Script yaratish

Skriptni yozishdan oldin siz kerakli maqsadni aniqlab olishingiz kerak, so‘ngra unga erishish uchun hal qilinishi kerak bo‘lgan vazifalar ro‘yxatini

tuzishingiz kerak. Inson juda murakkab operatsiyalarni qanday bajarishi haqida o‘ylamasdan amalga oshirishi mumkin. Misol uchun, ko‘p odamlar hech qanday batafsil ko‘rsatmalarsiz mashina haydashlari, elektron pochta xabarlarini yuborishlari va nonushta tayyorlashlari mumkin. Ammo hali bunday harakatlarni bajarishga o‘rganmaganimizda, ular bizga juda qiyin bo‘lib tuyuladi. Shuning uchun, yangi mahoratni o‘rganayotganda, biz uni ko‘pincha kichik kichik vazifalarga ajratamiz va keyin har birini alohida bajarishni o‘rganamiz. Tajriba orttirganingiz sayin, bunday kichik vazifalar tezroq bajariladi va osonroq ko‘rinadi.

Har qanday skript faqat qisqa ko‘rsatmalar to‘plamidir. Ko‘rsatmalar sizning oldingizda muammoni hal qilish imkonini beruvchi tartibda bajariladi. Shuning uchun skriptni dasturlash retsept yoki qo‘llanma yozishga o‘xshaydi va skriptning o‘zi kompyuterga muammoni bosqichma-bosqich hal qilish imkonini beradi. Ammo shuni ta’kidlash kerakki, kompyuter odamlar kabi o‘rganishga qodir emas. U tajriba to‘plamaydi, shuning uchun har safar muammoni hal qilganda u barcha ko‘rsatmalarga boshidan oxirigacha amal qilishi kerak. Dastur kompyuterga yechimning etarlicha batafsil tavsifini berishi kerak, chunki u har doim vazifani birinchi marta duch kelgandek bajaradi.

Skript nima va uni qanday yozish kerak. Skript - bu maqsadga erishish uchun kompyuter tomonidan bajariladigan bir qator ko‘rsatmalar to‘plamidir. Skript har safar ishga tushganda, unda yozilgan barcha ko‘rsatmalarni emas, balki faqat bir qismini bajarishi mumkin. Kompyuterlar muammolarni odamlar kabi hal qilmaydi, shuning uchun sizning ko‘rsatmalingiz muammoni dasturiy ravishda hal qilishga imkon berishi kerak. Skript yozishni boshlaganingizda, maqsadingizga boradigan yo‘lni bir qator kichik vazifalarga ajrating va keyin unga erishish uchun barcha zarur qadamlarni ishlab chiqing (buning uchun sxemani chizishingiz kerak bo‘lishi mumkin).

Hujjat ob’ekti html sahifasini ifodalaydi

Hujjat **obektidan** foydalanib, foydalanuvchilar sahifada ko‘rgan tarkibga kirish, shuningdek ular tomonidan amalga oshirilgan harakatlarga javob berish mumkin.

Ushbu obekt quyidagilarga ega:

XUSUSIYATLARI(СВОЙСТВА)

ular joriy veb-sahifaning xususiyatlarini tavsiflaydi (masalan, uning sarlavhasi);

USULLARI(МЕТОДЫ)

ular hozirda brauzerda yuklangan hujjat bilan bog‘liq vazifalarni bajaradilar (masalan, usul ko‘rsatilgan element haqida ma’lumot olish yoki yangi tarkib qo‘shish imkoniyatiga ega);

VOQEALAR(СОБЫТИЯ)

sichqonchani bosish yoki sensorli ekrandagi elementga tegish kabi hodisalarga munosabat bildirishingiz mumkin.

Obyekt va usullar bilan qanday ishlash

Quyida ushbu tilning obektlar va usullar bilan qanday ishlashini tushunish imkonini beruvchi JavaScript kodining qatori keltirilgan. Bunday holda, dasturchilar ma’lum bir ob’ektning ma’lum bir usuliga murojat qilishini aytishadi.

Document obekti butun veb-sahifani ifodalaydi. Barcha brauzerlar ushbu obektni amalga oshiradilar va undan foydalanish uchun uning nomini ko‘rsatish kifoya.

Document obyekti bir qancha usullar va xossalarga ega. Ular ushbu ob’ektning a’zolari deb ataladi. Ob’ektning a’zolariga murojat qilish uchun ob’ekt nomi va murojat qilmoqchi bo‘lgan a’zo o‘rtasida nuqta qo‘yishingiz kerak. A’zoga shunday murojaat qilinadi.



7.1-rasm. Document obekti va usullar

Hujjat obyektining write() usuli yangi tarkibni skript elementi joylashgan veb-sahifaga yozish imkonini beradi.

Qachonki, biror usul ishlashi uchun ba’zi ma’lumotlar talab qilinsa, bu ma’lumotlar qavs ichida ko‘rsatiladi.

Qavslar ichidagi ma’lumotlarning har bir qismi usul parametri deb ataladi. Bunday holda, write() usuli sahifaga nima yozish kerakligi haqida ma’lumot kerak. Document kabi ko‘plab obektlar, shuningdek write () kabi ko‘plab usullar mavjud. O‘z skriptlaringizni yozishda ular foydali bo‘ladi.

Web-sahifa uchun skriptni qanday yozish kerak

- 1) JavaScript kodingizni alohida faylda saqlash yaxshidir. JavaScript kod fayllari matnli fayllardir (HTML sahifasi va CSS uslublar jadvali fayllari kabi), lekin .js kengaytmasisiga ega.
- 2) HTML skript elementi web-sahifalarda brauzerga JavaScript faylini yuklab olishni bildirish uchun ishlatiladi. Shu tarzda, u CSS faylini yuklash uchun ishlatilishi mumkin bo‘lgan havola elementiga o‘xshaydi.

7.2. Javascriptda izohlar

Kod ichida u yoki bu fragment nima qilishini tushuntiruvchi izohlarni qoldirish kerak. Shuningdek, ular skriptni o‘qish va tushunishni yaxshilaydi. Sharhlar nafaqat sizga, balki sizning kodingiz bilan ishlaydigan boshqa odamlarga ham yordam berishi mumkin.

Ko‘p qatorli izohlar

Bir nechta qatorni qamrab oluvchi sharh yozish uchun boshida /* va oxiriga */ belgilarini qo‘yiladi. Ushbu juft belgilar orasidagi matnlar JavaScript interpretatori tomonidan qayta ishlanmaydi. Ko‘p qatorli izohlar ko‘pincha skript qanday ishlashini aniq tasvirlaydi yoki dastur ishga tushirilganda skriptning bir qismini bajarilishini bloklash uchun ishlatiladi.

Bir satli izohlar

Bir qatorli izohda, bir qatorda bir juft chiziqdan // keyin har qanday matn JavaScript interpretatori tomonidan tahlil qilinmaydi. Bir qatorli sharhlar ko‘pincha

kodning ishlashi haqida qisqacha eslatma berish uchun ishlatiladi. Sharhlardan to‘g‘ri foydalanish bilan siz bir necha kun yoki hatto oydan keyin kodingiz ustida osongina ishlappingiz mumkin. Bundan tashqari, izohli dastur kodlari uni birinchi marta ko‘rganlarga kodingizni tushunishga yordam beradi.

7.3. JavaScriptda o‘zgaruvchi

Skript ko‘pincha vazifani bajarish uchun zarur bo‘lgan ma’lumotlarni ma’lum vaqt davomida saqlashi kerak.

Ushbu ma’lumotlar o‘zgaruvchilarda saqlanishi mumkin. JavaScript kodini yozganingizda, siz bajarmoqchi bo‘lgan har bir operatsiyani innterpretatorga aytib berishingiz kerak. Ba’zida bu kutilganidan ko‘ra ko‘proq tafsilotlarni o‘z ichiga oladi. Aytaylik, devorning maydonini hisoblamoqchimiz. Matematikada to‘rtburchakning maydoni ikkita raqamning mahsulotidir:

Kenglik × Balandlik = Maydon.

Bunday hisob-kitoblar boshida osonlik bilan amalga oshirilishi mumkin, ammo bunday muammoni hal qilish uchun skriptni yozishda kompyuterga uni qanday bajarish bo‘yicha juda batafsil ko‘rsatmalar berilishi kerak. Masalan, u quyidagi to‘rtta operatsiyani bajarishi kerak bo‘ladi.

1. Kenglik qiymatini eslab qoling.
2. Balandlik qiymatini eslab qoling.
3. Maydonni olish uchun balandlikni kenglikka ko‘paytiring.
4. 3-bosqichda olingan qiymatni foydalanuvchiga qaytaring.

Bunday holda, o‘zgaruvchilar balandlik va kenglik qiymatlarini eslab qolish uchun ishlatiladi. Ushbu misolda biz kompyuterda nima qilish kerakligi va qanday tartibda juda aniq ko‘rsatmalar talab qilinishini ham ko‘ramiz. O‘zgaruvchilarni insonning qisqa muddatli xotirasi bilan taqqoslash mumkin, chunki siz veb-sahifani tark etishingiz bilan brauzer u haqida eslab qolgan barcha ma’lumotlarni darhol o‘chirib tashlaydi.

O‘zgaruvchi - bu kontseptsiya uchun juda aniq nom, chunki o‘zgaruvchida saqlangan ma’lumotlar skript har safar ishga tushirilganda o‘zgarishi (o‘zgarishi) mumkin.

Bitta devorning o‘lchamlaridan qat’i nazar, biz bilamizki, uning maydonini topish uchun devorning kengligini balandligi bilan ko‘paytirish kifoya. Skriptlar ko‘pincha bir xil muammolarni hal qilishlari kerak, garchi ular turli raqamlar bilan ishlashi mumkin. Shuning uchun tez-tez o‘zgarib turadigan ma’lumotlar bilan ishslash uchun skriptlarda o‘zgaruvchilar ko‘pincha ishlatiladi.

Natija o‘zgaruvchilarda saqlangan qiymatlar asosida hisoblanadi.

O‘zgaruvchini e’lon qilish

O‘zgaruvchini ishlatishdan oldin, o‘zgaruvchini yaratishingiz va unga nom berishingiz kerak. Dasturchilar bu jarayonni o‘zgaruvchini e’lon qilish deb atashadi.



Dasturlashda **var** obektli kalit so‘zga misol.

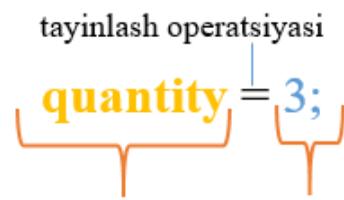
O‘zgaruvchidan foydalanish uchun unga nom berish kerak (ba’zan identifikator deb ham ataladi). Bunday holda,

o‘zgaruvchining nomi - qiymati

Agar o‘zgaruvchining nomi bir nechta so‘zlardan iborat bo‘lishi mumkin. Ular bo‘sh joy qoldirilmagan holda yozilishi kerak.

O‘zgaruvchiga qanday qiymat berish kerak

O‘zgaruvchi yaratilganda, siz unga qanday ma’lumotlarni saqlamoqchi ekanligingizni bildirishingiz mumkin. Bunday holda, dasturchi o‘zgaruvchiga qiymat berilishini aytadi.



o‘zgaruvchi nomi o‘zgaruvchi qiymati

Dastur davomida o‘zgaruvchini nomi bilan chaqirish orqali foydalaniladi. Bu yerda biz **quantity** deb nomlangan o‘zgaruvchiga qiymat o‘rnatamiz. Iloji bo‘lsa, o‘zgaruvchining nomi uning tarkibidagi ma’lumotlarni tavsiflashi kerak.

Tenglik belgisi (=) tayinlash operatoridir. Uni o‘rnatish orqali o‘zgaruvchiga qiymat beriladi. Bundan tashqari, u o‘zgaruvchida saqlangan ma’lumotlarni yangilash uchun ishlataladi.

Ma’lumot turlari

JavaScript ma’lumotlarning bir nechta turlarini ajratib turadi. Ulardan eng muhimi raqamlar, satrlar, rost yoki yolgon holatda bo‘lgan mantiqiy qiymatlar bo‘lishi mumkin.

Raqamli ma’lumotlar

Raqamli ma’lumotlar raqamlar bilan ishlash uchun mo‘ljallangan.

0,75

Satrli ma’lumotlar

String ma’lumotlari harflar va boshqa belgilardan iborat.

‘Salom!!!’

Mantiqiy qiymatlar

Mantiqiy (mantiqiy) ma’lumotlar ikkita qiymatdan faqat bittasini olishi mumkin: true yoki false.
true

Raqamni ma’lumotlari saqlashga namuna.

<pre><!DOCTYPE html> <html> <head> <title>javaScriptda ishlash</title> </head> <body> <div id="narx1">5 ta maxsulot narxi</div> <script src="java.js"></script> </body> </html></pre>	index.html	<pre>var narx; var miqdori; var jami; narx =5000; miqdor = 5; jami = narx * miqdor; var el = document.getElementById('narx1'); el.textContent = jami + ' sum';</pre>	java.js
---	-------------------	--	----------------

7.2-rasm. Raqamli ma’lumotlar bilan ishlashga namuna

```

<!DOCTYPE html>
<html>
<head>
    <title>javaScriptda ishlash</title>
</head>
<body>
<span id="name">Html!!</span>!
<div id="note">java Script..</div>
<script src="java.js"></script>
</div>
</body>
</html>

```

index.html

```

var ism;
var xabar;
ism = 'Aziza';
xabar = 'Salom Js!!';
var elName = document.getElementById('name');
elName.textContent = ism;
var elNote = document.getElementById('note');
elNote.textContent = message;

```

java.js

natija

Aziza!
java Script..

C:/Users/HP%2001

7.3-rasm. Belgili ma'lumotlar bilan ishlash

O'zgarchilardan foydalanishning qisqa yo'llari

```

var narx, miqdori, jami;
narx = 5000;
miqdori = 5;
jami = narx * miqdori;
var el = document.getElementById('narx1');
el.textContent = jami + ' sum';

```

java.js

```

var narx = 5000, miqdori = 5, jami;
jami = narx * miqdori;
var el = document.getElementById('narx1');
el.textContent = jami + ' sum';

```

java.js

7.4-rasm. O'zgaruvchilardan foydalanishning qisqa yo'llari

7.4. Arifmetik amallar

JavaScript raqamlar bilan ishlash uchun quyidagi matematik amallar foydalanadi (7.1-jadval).

7.1-jadval. Matematik amallar

Nomi	Amallar	Tavsifi	Namuna	Natija
Qo'shish	+	Bir qiymatni boshqasiga qo'shadi	10+5	15
Ayirish	-	Bir qiymatdan boshqasini ayiradi	10-5	5
Bo'lish	/	Bir qiymatni boshqasiga bo'ladi	10/5	2

Ko‘paytirish	*	Bir qiymatni boshqasiga ko‘paytirad	$10 * 5$	50
Bittaga oshirish	++	Joriy qiymatga bir qo‘shadi	$i = 10;$ $i++;$	11
Bittaga kamaytirish	--	Joriy qiymatdan biri ayiradi	$i = 10;$ $i--;$	9
Modul	%	Bir qiymatni boshqasiga bo‘ladi va qolgan qismini oladi	$10 \% 3$	1

Ushbu amallari string tipidagi ma’lumotlarda ham ishlatalish mumkin(7.5-rasm)

The screenshot shows a browser window with two tabs. The left tab is titled "index.html" and contains the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
    <title>javaScriptda ishlash</title>
</head>
<body>
    <div id="bir">Salom, HTML!!!</div>
    <script src="java.js"></script>
</body>
</html>
```

The right tab is titled "java.js" and contains the following JavaScript code:

```
var greeting = 'Salom, ';
var name = 'JavaScript';
var welcomeMessage = greeting + name + '!';
var el = document.getElementById('bir');
el.textContent = welcomeMessage;
```

Below the tabs, there is a message bubble from a bot named "javaScriptda ishlash" which says "natija". At the bottom of the browser window, there are navigation buttons and a file path: "C:/Users/HP%200".

Salom, JavaScript!

7.5-rasm. String toifasidagi ma’lumotlarni qo‘shish

NAZORAT SAVOLLARI

- 7.1. JavaScriptning asosiy vazifasi?
- 7.2. Web-sahifa uchun skriptni qanday yoziladi?
- 7.3. Javascriptda ko‘p qatorli va bir qatorli izohlar qanday beriladi?
- 7.4. O‘zgaruvchi nima?
- 7.5. O‘zgaruvchi qanday e’lon qilinadi?
- 7.6. Ma’lumotlarning qanday turlari bor?
- 7.7. Qanday arifmetik amallarni bilasiz?

VIII-BOB. JAVASCRIPTDA SHART OPERATORLARI

8.1. JavaScriptda IF shart operatori

Agar belgilangan shart rost bo‘lsa, shart operatori ishlasa biriktirilgan kod blok ishlaydi.

Kalit so‘z shart

```
if (ball >= 50) {  
    congratulate(); }  
} shart rost bo‘lganida  
bajariladigan amallar
```

Ushbu misolda **if** iborasi ball o‘zgaruvchisi qiymati 50 dan katta yoki teng ekanligini tekshiradi).

<pre><!DOCTYPE html> <html> <head> <title>javaScriptda ishlash</title> </head> <body> <div id="javob">imtixon </div> <script src="java.js"></script> </body> </html></pre>	Index.html	java.js
--	------------	---------

8.1-rasm. If shart operatoriga namuna

Bu yerda shart rost bo‘ladi, chunki ball o‘zgaruvchisi 75 qiymatiga ega, ya’ni 50 dan ortiq. Shuning uchun, blokning ichidagi kod bajariladi.

Ushbu kod ekranda ko‘rsatiladigan “tabriklaymiz!” xabarini yaratadi va foydalanuvchidan “keyingi bosqichga utish” so‘raladi.

Kod bloki bajarilgandan so‘ng, sahifada xabar chop etiladi.

Agar ball o‘zgaruvchisi qiymati 50 dan kam bo‘lsa, u holda kodning keyingi blokidagi ko‘rsatmalar bajarilmaydi.

8.2. JavaScriptda if else operatori

Agar **if** shart operatori yolg‘on qiymat qabul qilsa, aks holda **else** tarkibidagi kod bajariladi.

Kalit so‘z shart

```
if (ball >= 50) {  
    congratulate(); }  
else {  
    encourage(); }
```

shart rost bo‘lganida
bajariladigan amallar

shart yolg‘on bo‘lganida
bajariladigan amallar

If else operatoriga quyidagi misolni ko‘ramiz.

The screenshot shows a code editor with two files: index.html and java.js. The index.html file contains a simple HTML structure with a script tag pointing to java.js. The java.js file contains JavaScript code that defines a variable ball and checks if it's greater than or equal to 50. If true, it concatenates 'Tabriklaymiz!' and 'Siz utdingiz.' into a variable xabar. If false, it sets xabar to 'Qayta topshiring!'. Finally, it gets the element with id 'javob' and sets its textContent to xabar. Below the code editor is a browser window with two tabs both titled 'javaScriptda ishlash'. The first tab shows the source code of index.html. The second tab shows the result 'Qayta topshiring!', which is the output of the script.

index.html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>javaScriptda ishslash</title>  
  </head>  
  <body>  
    <div id="javob">imtixon</div>  
    <script src="java.js"></script>  
  </body>  
</html>
```

java.js

```
var ball = 45, utish_balli=50;  
var xabar;  
if (ball >= utish_balli) {  
    xabar = 'Tabriklaymiz!';  
    xabar += ' Siz utdingiz.';  
}  
else{  
    xabar = 'Qayta topshiring!';  
}  
var el = document.getElementById('javob');  
el.textContent = xabar;
```

Natija

javaScriptda ishslash

javaScriptda ishslash

← → ⌂ ⌂ Файл C:/Users/HP%20OMEN/Desktop/darslik/index.html

Qayta topshiring!

8.2-rasm. If .. else shart operatoriga misol

Bu yerda if..else shart operatori bir vaqtning o‘zida ikkita kod blokidan foydalanishga imkon berishini ko‘rashingiz mumkin:

1. birinchi blok — agar shart rost bo‘lsa;
2. boshqa blok, agar shart noto‘g‘ri bo‘lsa.

Ushbu dastur test jarayonida ikkita variantidan birini ta’minlaydi: foydalanuvchi o‘tish balliga teng yoki undan yuqori ball to‘plashi mumkin yoki o‘tish balliga qaraganda kamroq ball to‘plashi mumkin.

Demak, If...else shartli buyrug‘i, agar shart rost bo‘lsa, bitta kod blokini, agar shart noto‘g‘ri bo‘lsa, boshqasini bajarar ekan.

8.3. Switch operatori

Agar dastur mantiqiy emas, balki ba'zi ifodalarning qiymatlariga qarab harakatlar variantlarini tanlashi kerak bo'lsa, if iborasi o'rniga switch-case konstruktsiyasidan foydalanish qulayroqdir. **Switch case sintaksisi:**

```
switch(ifoda) {  
    case value_expression_1:  
        operator_set_1;  
        [break;]  
    case value_expression_2:  
        operator_set_2;  
        [break;]  
    ...  
    case value_expression_N:  
        operatorlar_to'plami_N;  
        [break;]  
    [default: statements_executed_by_default;  
    [break]]  
}
```

Yuqoridagi konstruksiyada switch operatorida ifoda xisoblanadi va natija har bir case blokining ifoda_value parametrlari bilan taqqosланади. Agar qiymatlar mos kelsa, u holda berilgan ish blokining operatorlar to'plami bajariladi. Agar ish bloklaridagi ifoda_qiymatlarining hech biri ifodani bajarish natijasiga mos kelmasa, default blokning operatorlari bajariladi.

Quyidagi misolda switch operatoridan foydalanishga misol keltirilgan(8.3-rasm).

The screenshot shows a browser window with two tabs. The left tab is titled 'index.html' and contains the following HTML code:

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>javaScriptda ishslash</title>  
</head>  
<body>  
    <div id="javob">imtixon  
    </div>  
    <script src="java.js"></script>  
</body>  
</html>
```

The right tab is titled 'natija' and displays the output of the script: 'Ikkinchchi bosqichdan utsak oxirgisi qoladi!'. Below the tabs, the browser's address bar shows the URL 'javaScriptda ishslash'.

On the right side of the image, there is a separate code block titled 'java.js' containing the following JavaScript code:

```
var xabar;  
var daraja = 3;  
switch (daraja) {  
    case 1:  
        xabar = 'Birinchi bosqichda omad!';  
        break;  
    case 2:  
        xabar = 'Ikkinchchi bosqichdan utsak oxirgisi qoladi!';  
        break;  
    case 3:  
        xabar = 'Yakuniy bosqichi!';  
        break;  
    default:  
        xabar = 'omad!';  
        break;  
}  
var el = document.getElementById('javob');  
el.textContent = xabar;
```

Ikkinchchi bosqichdan utsak oxirgisi qoladi!

8.3-rasm. Switch da ishslash

8.4. JavaScriptda shartni tekshirishda ishlataladigan operatorlar

1. Shartni tekshirish uchun quyidagi **taqqoslash** operatorlaridan foydalanish mumkin:

- ✓ == (tenglik), \$x==\$y ifodasi true, agar \$x qiymati \$y ga teng bo'lsa;
- ✓ != (tengsizlik), x!=y true, agar x y ga teng bo'lmasa;
- ✓ < (kamroq), x<y, agar x y dan kichik bo'lsa, true;
- ✓ > (katta), agar x y dan katta bo'lsa, x>y true;
- ✓ <= (kichik yoki teng), x<=y true, agar x y dan kichik yoki teng bo'lsa;
- ✓ >= (katta yoki teng), x>=y true, agar x y dan katta yoki teng bo'lsa.

Quyidagi misolda ikkita ifoda taqqoslash operatori yordamida ko'rsatilgan.

```
<!DOCTYPE html>
<html>
<head>
    <title>javaScriptda ishlash</title>
</head>
<body>
    <div id="javob">imtixon
    </div>
    <script src="java.js"></script>
</body>
</html>
```

```
var ball = 45, ball2=10, utish_balli=50;
var xabar;
if (ball+ball2 >= utish_balli+5) {
    xabar = 'Tabriklaymiz!';
    xabar += 'Siz utdingiz.';
} else{
    xabar = 'Qayta topshiring!';
}
var el = document.getElementById('javob');
el.textContent = xabar;
```

8.4-rasm. Taqqoslash operatori

2. Shartni tekshirish uchun quyidagi mantiqiy operatorlaridan foydalanish mumkin:

Ushbu operatorlar, shuningdek, dasturdagi hisob-kitoblar oqimini boshqarishga imkon beradi, bu esa uni iloji boricha moslashuvchan va intuitiv qiladi. Masalan, ular ikkita shartli operatorni bir-biriga bog'lash imkonini beradi:

```
if (mantiqiy_ifoda_1&&mantiqiy_ifoda_2)
{
    amallar
}
```

Bu yerda if dan keyingi operator mantiqiy ifoda_1 VA mantiqiy ifoda_2 rost bo'lsa bajariladi. Shunday qilib, mantiqiy ifodalar va operatorlarni birlashtirib, dasturning har qanday, eng murakkab mantiqini amalga oshiradigan juda murakkab tuzilmalarni qurish mumkin.

AND (mantiqiy “VA”), x VA y ifodasi to‘g‘ri, agar x va y ikkalasi ham to‘g‘ri bo‘lsa;

OR (mantiqiy “YOKI”), x OR y ifodasi to‘g‘ri, agar x yoki y rost bo‘lsa;

XOR (eksklyuziv OR), x XOR y rost, agar x yoki y rost bo‘lsa, lekin ikkalasi ham emas;

! (*inkor*), !x ifodasi to‘g‘ri, agar x noto‘g‘ri bo‘lsa;

&& (mantiqiy AND), x && y ifodasi, agar x va y ikkalasi ham to‘g‘ri bo‘lsa, rost;

//(mantiqiy “YOKI”), x yoki y ifodasi to‘g‘ri bo‘ladi, agar x yoki y rost bo‘lsa. Agar if operatoridagi mantiqiy ifoda rost bo‘lsa, u holda if konstruktsiyasidan keyingi operatorlar bajariladi. Agar bunday operatorlar bir nechta bo‘lsa, ular figurniy qavslarga joylashtiriladi.

NAZORAT SAVOLLARI:

- 8.1.Qanday shart operatorlarini bilasiz?
- 8.2.If shart operatori qanday holatlarda ishlatiladi?
- 8.3.If..else operatorini ishslash tamoyilini tushuntiring?
- 8.4.Switch operatori vazifasi?
- 8.5.Switch operatorida default qanday vazifani bajaradi?
- 8.6.Taqqoslash operatorlarini aytинг?
- 8.7. And va Or mantiqiy operatorlarining farqi nimada?

IX-BOB. JAVASCRIPTDA SIKL OPERATORLARI BILAN ISHLASH

Sikllar shartlardan keyingi eng muhim operatorlar toifasi hisoblanadi. Ular ma'lum bir shartga muvofiq ma'lum bir operatorlar to'plamining bajarilishini takrorlash uchun mo'ljallangan. Takroriy operatorlar sikl operatorining asosiy qismini tashkil qiladi va bu operatorlar orqali takroriyalar sikllar tashkil qilinadi.

Sikllar u yoki bu shartni tekshiradi: agar natija to‘g‘ri bo‘lsa, u holda kod bloki bajariladi. Keyin u yana tekshiriladi va agar u yana rost bo‘lsa, kod bloki yana bajariladi. Shart noto‘g‘ri natija berguncha sikl davom etadi.

9.1. JavaScriptda for takrorlash operatori

For operatori sikl tanasining operatorlarini ma'lum bir necha marta bajarishga imkon beradi. *For sikl operatorining sintaksisi quyidagicha:*

```
for (<ifoda1>; <ifoda2>; <ifoda3>)
    <operator yoki blok>;
```

Bu operatorda amal bajarish <ifoda1>dan boshlanadi. Keyin takrorlash qadamlari boshlanadi. Har bir qadamda <ifoda2> bajariladi, agar natija 0 dan farqli yoki rost (true) bo‘lsa, takrorlanish tanasi – <operator yoki blok> bajariladi va oxirida <ifoda3> bajariladi, aks holda boshqaruva takrorlash operatoridan keyingi operatorga o‘tiladi. Takrorlanish tanasi – <operator yoki blok> sifatida bitta operator, shu jumladan bo‘sh operator yoki operatorlar bloki kelishi mumkin. Takrorlanish takrorlanishi davomida bajarilishi lozim bo‘lgan operatorlar majmuasi takrorlanish tanasi deyiladi. Takrorlanish tanasi sifatida bir yoki bir nechta operatordan foydalanish mumkin.

Agar takrorlanish tanasida bir nechta operatordan foydalanmoqchi bo‘lsak bu operatorlarni blok { } orasiga olishimiz kerak.

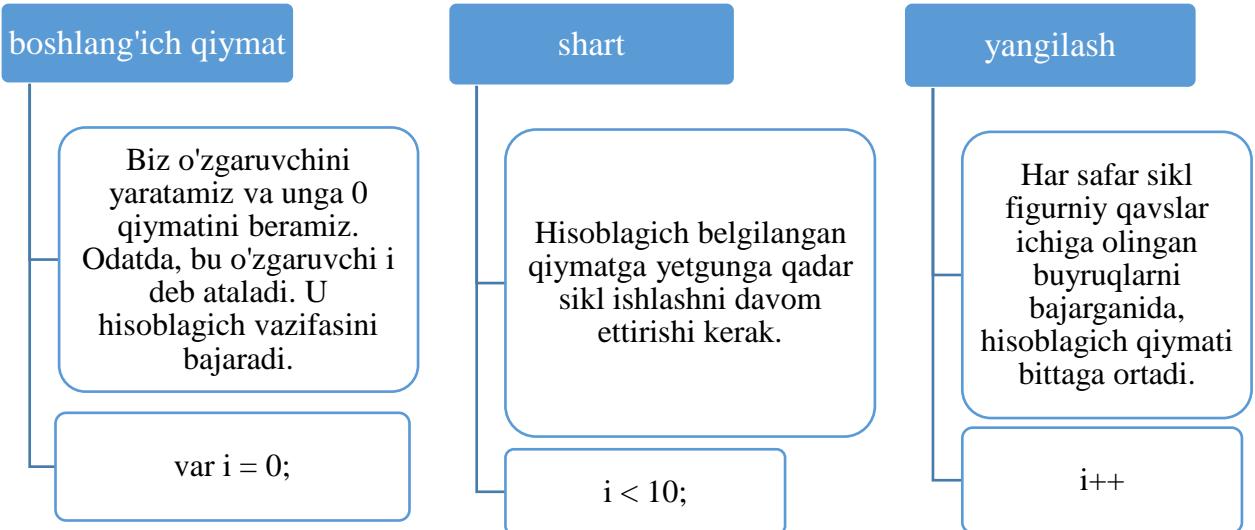
kalit so‘z takrorlash operatori

```
for (var i = 0; i < 10; i++) {
```

document.write(i); }

takrorlash sikli ichida
bajariladigan kod

Sikl hisoblagichi



O'zgaruvchi faqat siklning birinchi iteratsiyasida yaratiladi. Ba'zan u i emas, balki indeks deb ataladi. O'zgaruvchi shartdan oldin e'lon qilinadi. Quyidagi kod yuqoridagi bilan bir xil, tanlov faqat dasturchning xohishiga bog'liq.

```
var i;
for (i = 0; i < 10; i++) {
// bu yerda kod yoziladi
}
```

i o'zgaruvchisi dastlab 0 ga o'rnatildi, shuning uchun bu holda sikl 10 marta ishlaydi va keyin to'xtaydi. Shart raqamni o'z ichiga olgan o'zgaruvchi ham bo'lishi mumkin.

Agar rounds o'zgaruvchisi tekshirish bosqichlari sonini o'z ichiga olgan bo'lsa va sikl har bir bosqichda bir marta ishlashi kerak bo'lsa, shart quyidagicha shakllantiriladi:

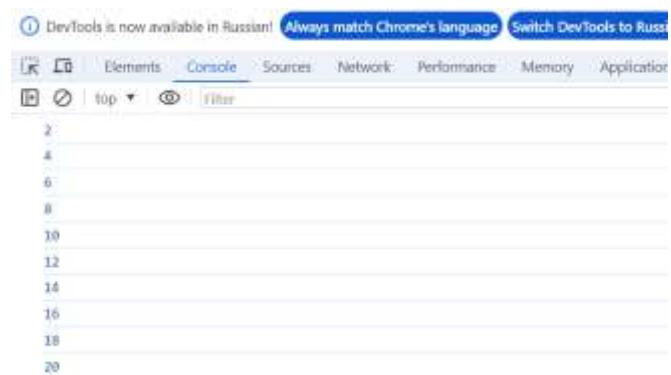
```
var rounds = 3;
i < (rounds);
```

Inkrement operatoridan foydalanish uchun “++” foydalaning.

Boshqacha qilib aytganda, bu kod: “i o'zgaruvchisini olishni va ++ operatori yordamida unga 1 qo'shishni” anglatadi. sikllarni teskarri yo'nalishda ham ishlatish mumkin, buning uchun dekrement operatsiyasi (--) ishlatiladi.

For sikliga misol qilib quyidagi n gacha bo'lgan sonlar ichida juft solarni chiqarishni keltirishimiz mumkin(natijani ko'rish uchun brauzer oynasi sichqonchaning o'ng tugmasi bosiladi va просмотреть код->console tanlanadi)

```
<!DOCTYPE html>
<html>
<head>
    <title>javaScriptda ishlash</title>
</head>
<body>
<script>
    var i, n=20;
    for(i=0; i<=n;i++){
        if (i%2==0) {
            console.log(i);
        }
    }
</script>
</body>
</html>
```



9.1-rasm. For takrorlanish operatoriga namuna

9.2. WHILE takrorlash operatori

Agar kodni necha marta bajarish kerakligini oldindan bilmasangiz, while siklidan foydalanishingiz mumkin. Bu yerda shart hisoblagich qiymati emas, balki boshqa omil. Shart to‘g‘ri bo‘lsa, kod bajariladi.

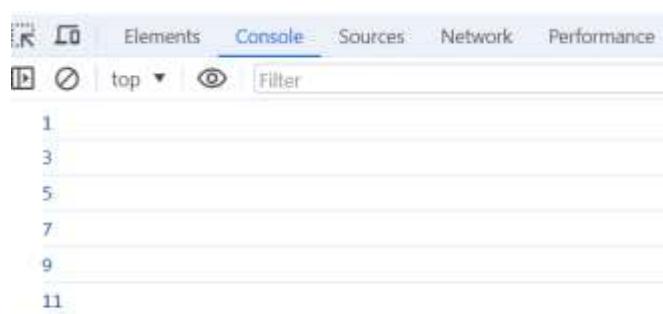
While sikli operatorining sintaksisi:

```
while (<ifoda>)
{
    <operator yoki blok>;
}
```

while sikl operatori quyidagicha ishlaydi:

1. Mantiqiy ifodaning qiymati hisoblanadi.
2. Agar qiymat rost bo‘lsa, sikl tanasi bajariladi, aks holda biz sikldan keyingi operatororga o‘tamiz.

Boshqacha qilib aytadigan bo‘lsak, mantiqiy shart sikl boshida tekshiriladi.



The screenshot shows a browser's developer tools open to the 'Console' tab. The code in the left panel is:

```
<!DOCTYPE html>
<html>
<head>
    <title>javaScriptda ishlash</title>
</head>
<body>
<script>
    var i=0;
    while (i++<12) {
        if (i%2!=0)
        {
            console.log(i);
        }
    }
</script>
</body>
</html>
```

The right panel shows the console output with the numbers 1, 3, 5, 7, 9, and 11 listed vertically.

9.2-rasm. While takrorlash operatoriga namuna

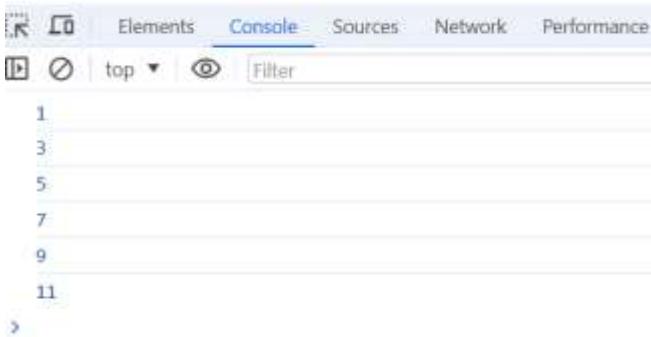
Postfiks ortishi bilan $i++ < 12$ shartlarni bajarish ketma-ketligini hisobga olish muhimdir. Birinchidan, shart tekshiriladi va shundan keyingina o‘zgaruvchining qiymati oshiriladi. Agar o‘stirish amali prefiks bo‘lsa, ya’ni $++i < 12$ shaklida yozilsa, u holda avval o‘zgaruvchi o‘sadi va shundan keyingina taqqoslash amalga oshiriladi.

9.3. DO WHILE takrorlash operatori

do-while takrorlash operatori *while* operatoridan farqli ravishda oldin operator yoki blokni bajaradi, keyin takrorlanish shartini tekshiradi. *do-while* takrorlash operatori quyidagi sintaksisiga ega:

```
do  
  <operator yoki blok>  
  while (<ifoda>);
```

Bunday takrorlash operatorining qo'llaniladigan holatlari takrorlanishni boshlamasdan turib, takrorlanish shartini tekshirishning iloji bo'limgan holatlar hisoblanadi.



The screenshot shows a browser's developer tools open to the 'Console' tab. It displays the following sequence of numbers: 1, 3, 5, 7, 9, 11. This output corresponds to the execution of a `do-while` loop in the provided JavaScript code.

```
<html>  
<head>  
  <title>javaScriptda ishlash</title>  
</head>  
<body>  
  <script>  
    var i=0;  
    do  
    {  
      if (i%2!=0)  
      {  
        console.log(i);  
      }  
      while (i++<12)  
    </script>  
</body>  
</html>
```

9.3-rasm. Do while takrorlash operatoriga namuna

NAZORAT SAVOLLARI:

- 9.1. Qanday takrorlash operatorlarini bilasiz?
- 9.2. For takrorlash operatorining sintaksisi qanday?
- 9.3. While takrorlash operatorining sintaksisi qanday?
- 9.4. Do While takrorlash operatorining sintaksisi qanday?
- 9.5. Takrorlaish operatorlariing farqli jihatlari?
- 9.6. Takrorlanish operatorlaridan foydalanib n gacha bo'lgan sonlar ichida 3 va 5 ga bo'linadigan sonlarni console rejimda ekranga chiqarish kodini yozing?

X-BOB. JAVASCRIPTDA FUNKSIYALAR VA MASSIVLAR BILAN ISHLASH

10.1. JavaScriptda funksiyalar bilan ishlash

Funksiya ma'lum bir harakatni bajarish uchun bir qator ko'rsatmalarni guruhlash imkonini beradi. Agar bir xil vazifa skriptning turli qismlarida qayta-qayta hal qilinsa, siz funksiyadan qayta foydalanishingiz mumkin (bir xil ko'rsatmalar to'plamini qayta yozish o'rni).

Muayyan muammoni hal qilish uchun zarur bo'lgan ko'rsatmalarni guruhlash kodingizni tartibga solishga yordam beradi. Bundan tashqari, funksiyadagi ko'rsatmalar har doim ham sahifani yuklash bosqichida bajarilmaydi, shuning uchun funksiya ma'lum bir tarzda muammoni hal qilish uchun bajarilishi kerak bo'lgan bosqichlarni saqlashga imkon beradi. Keyin skript dasturda kerak bo'lganda barcha shu kabi amallarni bajarish uchun funksiyani chaqirishi mumkin. Masalan, foydalanuvchi ma'lum bir sahifa elementini bosgan taqdirdagina bajarilishi uchun dasturlashtirilgan vazifaga ega bo'lishingiz mumkin.

Agar siz funksiyaga berilgan vazifani keyinroq bajarishini xoxlasangiz, unda siz funksiyaga nom berishingiz kerak. U funksiya tomonidan bajariladigan vazifani tavsiflashi kerak. Dasturchi funksiyaga vazifani bajarishni buyursa, u funksiyani chaqirish deyiladi. Funksiya vazifani bajarish uchun o'tishi kerak bo'lgan bosqichlar kodlar blokiga to'planadi. Kod bloki jingalak qavslar ichiga olingan bir yoki bir nechta ko'rsatmalardan iborat bo'lib, yopuvchi qavsdan keyin nuqtali vergul qo'yish shart emas, bu nuqtali vergul faqat ko'rsatmalardan keyin qo'yiladi.

Ba'zi funksiyalar berilgan vazifani bajarishi uchun ma'lumotlar bilan ta'minlanishi kerak. Misol uchun, agar funksiya to'rtburchaklarning maydonini hisoblashi kerak bo'lsa, biz funksiyaga uning balandligi va kengligini aytishimiz kerak. Funksiyaga uzatiladigan ma'lumotlar bo'laklari **parametrlar** deb ataladi.

Sizga javob berishi kerak bo'lgan funksiyani yozganingizda, u qaytariladigan qiymat deb nomlanadi. Quyida JavaScript faylidan funksiya misoli keltirilgan. U updateMessage() deb ataladi.

Misolda updateMessage nomli funksiya va qiymat – ko‘rsatmalardan iborat kod bloki mavjud. Funksiyani nomi bilan chaqirganingizda, funksiya tarkibidagi ko‘rsatmalar bajariladi.

Function.html

```
<html>
<head>
<title>simple example</title>
</head>
<body>
<h1>Funksiyalar bilan ishslash</h1>
<div id="message">Welcome our
site!</div>
<script src="javafunk.js"></script>
</body>
</html>
```

javafunk.js

```
var msg = 'Bizning yangiliklarimizga obuna boling va 10% chegirmaga ega boling!';
function updateMessage() {
    var el = document.getElementById('message');
    el.textContent = msg;
}
updateMessage();
```

Natija

Funksiyalar bilan ishslash

Bizning yangiliklarimizga obuna boling va 10% chegirmaga ega boling!

10.1-rasm. JavaScriptda funksiyalar bilan ishslashga namuna

Ushbu misolda biz sahifaning yuqori qismida joylashgan xabarni ko‘rsatamiz. Xabar identifikatori(id) 'message' bo‘lgan HTML elementida joylashgan. Biz bu xabarni JavaScript kodi yordamida o‘zgartiramiz. </body> tegini yopilishdan oldin JavaScript fayliga havola joylashgan. JavaScript fayli yangi xabarni o‘z ichiga olgan o‘zgaruvchidan boshlanadi. Keyin- updateMessage() funksiyasi joylashgan.

Funksiyani e’lon qilish

Funksiyani yaratish uchun biz unga nom beramiz va keyin muammoni hal qilish uchun zarur bo‘lgan ko‘rsatmalarni figurali qavslarga yozamiz. Bu jarayon funksiyani e’lon qilish deb ataladi.

Funksiya function kalit so‘zi yordamida e’lon qilinadi. Funksiyaga nom beriladi (ba’zan identifikator deb ataladi), keyin qavslar qo‘yiladi.

Vazifani bajaradigan ko‘rsatmalar kod blokida joylashgan (ular figurali qavslarda yozilgan).



10.2-rasm. Funksiyani e’lon qilishga namuna

Bu misol juda oddiy namuna bo‘lib, (figurali qavslarda faqat bitta ko‘rsatma bor), u funksiyalar qanday e’lon qilinishini tushunishga yordam beradi. Siz yozadigan va o‘qiydigan funksiyalarning aksariyati bir necha yoki bir nechta ko‘rsatmalardan iborat bo‘ladi.

Bu yerda funksiyalar ma’lum bir muammoni hal qilish uchun zarur bo‘lgan kodni saqlashini yodda tutish kerak. Skript har doim bunday vazifani bajarishi kerak bo‘lganda ushbu funksiyani chaqirishi mumkin.

Agar skriptning turli qismlarida ma’lum bir muammoni hal qilish kerak bo‘lsa, unda siz bir xil ko‘rsatmalarni bir necha marta takrorlashingiz shart emas.

Har safar muammo yechilganida kodi qaytadan foydalilanligan funksiyani yozish kifoya.

Funksiyani chaqirish

Funksiya e’lon qilingandan so‘ng, uning figurali qavslaridagi barcha ko‘rsatmalarni faqat bitta kod qatori bilan bajarishingiz mumkin. Ushbu jarayon **funksiya chaqiruvি** deb ataladi.

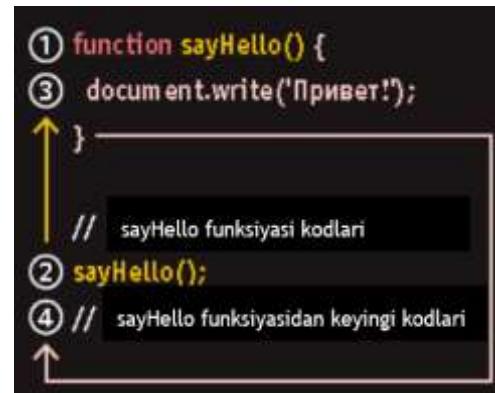


10.3-rasm. Funksiyani chaqirish

Funksiyadagi kodni bajarish uchun biz uning nomini, so‘ngra ikkita qavsnini yozamiz. Bunday holda, dasturchilar kod funksiyani chaqirishini aytishadi.

Bitta JavaScript faylida siz funksiyani xohlagancha, ko‘p marta chaqirishingiz mumkin.

1. Ushbu funksiya muayyan vazifani hal qilish uchun ko‘rsatmalarini saqlashi mumkin.
2. Agar siz skript ushbu vazifani bajarishini istasangiz, undagi funksiyani chaqirasiz.
3. Funksiya kod blokida joylashgan buyruqlarni bajaradi.
4. Funksiya tugagach, kodni bajarish funksiya chaqiruviga qilingan joydan davom etadi.



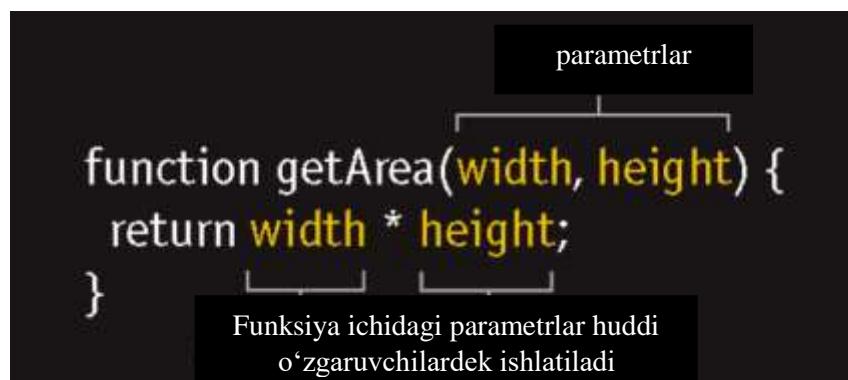
Ba’zi hollarda funksiya e’lon qilinishidan oldin chaqiriladi. Chunki tarjimon avval butun skriptni ko‘rib chiqadi va keyin ko‘rsatmalarini bajarishni boshlaydi va dastur skriptda keyinroq funksiya e’lon qilinishini bilib oladi.

Ta’lab qilinadigan ma’lumotlar uchun funksiyani e’lon qilish

Ba’zan funksiya o‘ziga yuklangan vazifani bajarish uchun u yoki bu ma’lumotga muhtoj bo‘ladi. Bunday hollarda funksiyani e’lon qilishda biz unga parametrlarni beramiz. Funksiya ichida parametrlar o‘zgaruvchilar kabi ishlaydi.

Agar funksianing ishlashi uchun ma’lumot kerak bo‘lsa, u funksiya nomidan keyin qavs ichida yoziladi.

Qavslar ichidagi ma’lumotlar funksiya parametrlari deb ataladi. Funksiya ichida parametrlar o‘zgaruvchilar kabi ishlaydi.



10.4-rasm. Funksiya parametrlarini e’lon qilish

Ushbu namunada funksiya to‘rtburchakning maydonini hisoblab chiqadi va natijada olingan qiymatni qaytaradi. Buning uchun biz funksiyaga to‘rtburchakning balandligi(height) va kengligini(width) berishimiz kerak. Har safar funksiya chaqirilganda, unga berilgan qiymatlar boshqacha bo‘lishi mumkin.

Shunday qilib, stsenariyni loyihalashda, vazifani hal qilish uchun funksiyaga kerak bo‘lgan ma’lumotlarni aniqlash kerak. Funksiyani diqqat bilan ko‘rib chiqsangiz, parametrlar o‘zgaruvchilar bilan aynan bir xil ishlatalishini ko‘rishingiz mumkin. Demak, parametrlar va funksiya ichidagi o‘zgaruvchilar bir xil nomda bo‘lishi kerak. Bunday holda, width va height deb nomlangan parametrlar devorning width va heightga mos keladi.

Ta’lab qilinadigan ma’lumotlar uchun funksiyani chaqirish

Parametrlari bo‘lgan funksiyani chaqirganda, uning nomidan keyin qavslar ichida bajarish paytida foydalanishi kerak bo‘lgan qiymatlarni ko‘rsatish kerak. Ushbu qiymatlar argumentlar deb ataladi, ular aniq raqamlar, satrlar va boshqalar yoki o‘zgaruvchilar bo‘lishi mumkin.

Argumentlarni qiymatlar

Quyidagi funksiyani chaqirganingizda, width 3 ga, height esa 5 ga teng bo‘ladi.

```
getArea(3, 5);
```

Argumentlar-o‘zgaruvchilar

Funksiyani chaqirishda siz aniq qiymatlarni ko‘rsata olmasangiz, uning o‘rniga o‘zgaruvchilardan foydalanishingiz mumkin.

Quyidagi misol avvalgisi bilan bir xil vazifani bajaradi.

```
width = 3;
```

```
height= 5;
```

```
getArea(width, height);
```

Parametrlar va argumentlarni taqqoslash

Parametr va argument atamalari ko‘pincha bir-birining o‘rnida ishlataladi, ammo ular orasida farqlar mavjud. Oldingi sahifada funksiya deklaratsiyasida width

va height (birinchi qatordagi qavslar ichida) kalit so‘zлari ishlataligani. Funksiyadagi figurali qavslarda bu so‘zлar o‘zgaruvchi vazifasini bajaradi. Ularning nomlari parametrdir.

Bu sahifa getArea() funksiyasini chaqiradi va kod hisoblashda foydalilaniladigan aniq raqamlarni belgilaydi (raqamlarni o‘z ichiga olgan o‘zgaruvchilar).

Kodga aytgan qiymatlar (ya’ni, ushbu devorning maydonini hisoblash uchun zarur bo‘lgan ma’lumotlar) argumentlardir.

Funksiya yordamida bitta qiymatni olish

Ba’zi funksiyalar ma’lumotni ularni chaqirgan kodga qaytaradi. Masalan, funksiya hisoblashni amalga oshirishi va natijasini qaytarishi mumkin.

calculateArea()	Area o‘zgaruvchisi	return kalit so‘zi
funksiyasi to‘rtburchaklar	funksiya ichida yaratiladi.	funksiyani chaqirgan
maydonini uni chaqirgan	U to‘rtburchakning	kodga qiymatni qaytarish
kodga qaytaradi.	hisoblangan maydonini	uchun ishlataladi.
	o‘z ichiga oladi.	

```
<html>
<head>
    <title>javaScriptda ishlash</title>
</head>
<body>
<script>
    function calculateArea(width, height) {
        var area = width * height;
        return area;
    }
    var wallOne = calculateArea(3, 5);
    var wallTwo = calculateArea(8, 5);
    console.log("1-maydon qiymati = " + wallOne);
    console.log("2-maydon qiymati = " + wallTwo);
</script>
</body>
</html>
```



10.5-rasm. Funksiyalar bilan ishlash

10.2. JavaScriptda massivlar bilan ishlash

Massiv o‘zgaruvchilarning maxsus turidir. Massiv faqat bitta qiymatni emas, balki ularning butun ro‘yxatini saqlaydi.

Agar siz ro‘yxat yoki boshqa o‘zaro bog‘langan qiymatlar to‘plami bilan ishlayotgan bo‘lsangiz, ularni massivda saqlash mantiqan to‘g‘ri keladi.

Ro‘yxatda qancha element bo‘lishini oldindan bilmasangiz, massivlar bilan ishslash foydalidir. Gap shundaki, massivni yaratishda unga qancha qiymatlarni joylashtirish rejalashtirilganligi belgilanmagan.

Agar qancha element bilan ishlappingizni bilmasangiz, ko‘p o‘zgaruvchilar yaratish o‘rniga massivdan foydalangan ma’qul (chunki amalda ulardan faqat bir nechtasi kerak bo‘lishi mumkin).

Misol uchun, massiv xaridlar ro‘yxatidagi alohida elementlarni saqlash uchun foydalidir, chunki ular bir-biriga bog‘liq.

Bundan tashqari, har safar xaridlar ro‘yxatini yaratganingizda, undagi narsalar soni avvalgi holatdan farq qilishi mumkin.

Massivdagi qiymatlar vergul bilan ajratiladi.

```
<html>
<head>
    <title>javaScriptda ishlash</title>
</head>
<body>
    <h1 id="colors">Ranglar</h1>
    <script>
        var colors;
        colors = ['oq', 'qora', 'yashil'];
        var el = document.getElementById('colors');
        el.textContent = colors[0];
    </script>
</body>
</html>
```

10.6-rasm. Massivlar bilan ishslash

Biz massivni yaratamiz va unga oddiy o‘zgaruvchiga o‘xshab nom beramiz, yuqoridagi misolda unga colors deb nom berilgan. Massiv elementlariga tayinlangan qiymatlar vergul bilan ajratilgan kvadrat qavs ichida keltirilgan. Massiv bir vaqtning o‘zida har xil turdagи ma’lumotlarni o‘z ichiga olishi mumkin: satrli, raqamli va mantiqiy qiymatlar.

Birinchi misolda keltirilgan variant literal(harfli) deb ataladi. Qoida tariqasida massivlar shu usul yordamida yaratiladi. Har bir qiymatni alohida satrga yozishingiz mumkin:

```
colors = ['oq',
          'qora',
          'yashil'];
```

Keyingi misol boshqa yo‘lni ko‘rsatadi: massiv konstruktori. Bunda new kalit so‘zdan keyin Array() ishlatiladi. Qiymatlar vergul bilan qavs ichida (kvadrat emas)

ko'rsatiladi. Massivdan ma'lumotlarni chiqarish uchun item() usulidan foydalilaniladi. Elementning raqami (indeks) qavs ichida ko'rsatiladi.

Massivdagi qiymatlar

Massivdagi elementlarni nomerlash

Massivdagi har bir elementga avtomatik ravishda indeks deb ataladigan raqam beriladi. Indeks massivdagi muayyan elementlarga kirish uchun ishlataladi. Uch rangni o'z ichiga olgan quyidagi massivni ko'rib chiqing:

```
var colors;
colors = ['oq',
'qora',
'yashil'];
```

Bir qarashda, indeks noldan boshlanishi (birdan emas) g'ayrioddiy ko'rinishi. Quyidagi jadvalda massiv elementlari va ularning tegishli indeks qiymatlari ko'rsatilgan:

indeks	qiymat
0	'oq'
1	'qora'
2	'yashil'

Massiv elementlariga murojat qilish

Ro'yxatdagi uchinchi elementni chiqarish uchun biz ushbu element nomini va uning indeksini kvadrat qavs ichida ko'rsatamiz. Quyida itemThree o'zgaruvchisining deklaratasiysi keltirilgan. Uning qiymati ranglar qatoridan uchinchi rangga o'rnatiladi.

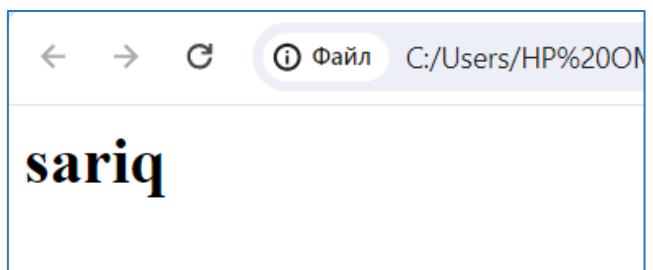
```
var itemThree;
itemThree = colors[2];
```

Massivdagi elementlar soni

Har bir massiv uzunlik xususiyatiga ega bo'lib, u massivdagi elementlar sonini o'z ichiga oladi. Ushbu xususiyat bilan ishlashni boshlashdan oldin numColors o'zgaruvchisini e'lon qilaylik. Uning qiymati massivdagi elementlar soniga teng. Massiv nomidan keyin nuqta, so'ngra length kalit so'zi qo'yiladi.

```
var numColors;
numColors=colors.length;
```

```
<html>
<head>
    <title>javaScriptda ishlash</title>
</head>
<body>
    <h3 id="colors">Colors</h3>
    <script>
        var colors = ['oq',
        'qora',
        'yashil'];
        colors[2] = 'sariq';
        var el = document.getElementById('colors');
        el.textContent = colors[2];
    </script>
</body>
</html>
```



10.7-rasm. Massiv elementlariga murajat qilish va ularning qimmatlarini o'zgartirish

Yuqoridagi misolda uchta rang ro‘yxatini o‘z ichiga olgan colors nomli massiv yaratilgan. Massivning qiymatlari bitta satrga yoki alohida satrlarga yozilishi mumkin.

Massivni yaratgandan so‘ng, biz uning uchinchi elementi “yashil” ni “sariq” qiymati bilan almashtiryapmiz.

Massivdagi qiymatiga murajat qilish uchun massiv nomidan keyin kvadrat qavs ichida kerakli element indeksini belgilaymiz.

Massivdagi elementning qiymatini o‘zgartirish uchun siz o‘sha elementni tanlab, teng belgisi yordamida unga yangi qiymat berishingiz kerak.

NAZORAT SAVOLLARI:

- 10.1. Funksiya qanday e’lon qilinadi?
- 10.2. Funksiya parametrlari nima?
- 10.3. Funksiyada argument qanday vazifani bajaradi?
- 10.4. return ning vazifasi nima?
- 10.5. Massiv nima?
- 10.6. Massivlardan qanday maqsadda foydalanish mumkin?
- 10.7. Massiv indeksi nima va u nechidan boshlanadi?
- 10.8. Massivning uzunligi qanday aniqlanadi?

XI-BOB. JAVASCRIPTDA SATRLAR BILAN ISHLOVCHI FUNKSIYALAR BILAN ISHLASH

11.1. JavaScriptda obektlar

Ob'ekt o'zgaruvchilar va funksiyalar to'plamini birlashtiradi. Ob'ektlarda o'zgaruvchilar va funksiyalar turlicha chaqiriladi.

Obyektda o'zgarchilar xususiyatlar deyiladi.

Agar o'zgaruvchi obektda bo'lsa, u holda u xususiyat deyiladi. Xususiyatlar obektning tafsilotlarini tavsiflaydi, masalan, mehmonxona nomi va undagi xonalar soni. Har bir mehmonxonaning o'z nomi bor, turli mehmonxonalardagi xonalar soni ham odatda farq qiladi.

Obektda funksiyalar usullar deb ataladi.

Agar funksiya obektda bo'lsa, u usul deb ataladi. Usullar obekt bilan bog'liq vazifalardir. Masalan, mehmonxonada qancha xona mavjudligini tekshirish mumkin. Buning uchun bron qilingan xonalar sonini ularning umumiyligi sonidan chiqarib tashlashingiz kerak.

Quyidagi misolda mexmonxona nomli obekt yaratilgan bo'lib, uning bitta usuli va beshta xususiyati bor. Obekt kodi figurniy qavslar ichida beriladi. Ob'ekt mexmonxona o'zgaruvchisida saqlanadi.

Xususiyat va metodlar o'zgaruvchilar va nomlangan funksiyalar kabi nom va qiymatga ega. Obektda bu nom kalit deb ataladi. Obektda bir nomli ikkita kalit bo'lishi mumkin emas. Gap shundaki, kalitlar ular bilan bog'langan qiymatlarga murajat qilish uchun ishlatiladi. Xususiyat qiymati satr, raqam, mantiqiy, massiv yoki hatto boshqa obekt bo'lishi mumkin. Usulning qiymati har doim funksiyadir.

Yuqorida mehmonxona obekti yaratilgan bo'lib, u o'z ichiga quyidagi kalit/qiymat juftliklarini oladi:

Xususiyat: kalit	qiymatlari
nomi	satr
xonalar_soni	raqam
bron_qilingan_raqam	raqam
sport_zal	mantiqiy
xonalar_tipi	massiv tipi
usul mavjud_xonalarni_tekshirish	funksiya

Var mexmonxona ={

nomi: ‘Toshkent’,

xonalar_soni: 56,

bron_qilingan: 25,

sport_zal: true,

xonalar_tipi:[‘kup_xonali’, ‘ikki_kishilik’, ‘bir_xonali’],

}

● Kalit so’zlar

● qiymat

O‘zgaruvchining
xossalari

mavjud_xonalarni_tekshirish: function(){

return this.xonalar_soni-this.bron_qilingan;

}

Funksiyaning
usullari

Keyingi bir nechta sahifalar bu ob’ektni yaratishning mumkin bo‘lgan usullaridan biri ekanligini ko‘rsatadi.

Dasturchilar odatda, nomi/qiymat juftliklaridan keng foydalanadilar:

- HTML atribut nomlari va qiymatlaridan foydalanadi;
- CSS xususiyat nomlari va qiymatlaridan foydalanadi.

JavaScriptda:

- o‘zgaruvchilar nomiga ega bo‘lib, ular qatorli, sonli va mantiqiy qiymatlarga ega bo‘lishi mumkin;

- massiv nomga ega va qiymatlar guruhini o‘z ichiga oladi; har bir massiv elementi nom/qiymat juftligidir, chunki element indeks va qiymatga ega;
- nomlangan funksiyalar nomi va qiymatiga ega bo‘lib, ularning qiymati shunday funksiya chaqirilganda bajarilishi kerak bo‘lgan buyruqlar to‘plamidir.

Obektlar nom/qiymat juftliklaridan iborat (obektlarda nomlar kalitlar deb ataladi).

Harfli belgilar - obektlarni yaratishning eng oddiy va eng mashhur usuli hisoblanadi. Umuman olganda, bir nechta bunday usullar mavjud.

Obektfigurali qavslardan va qavslar ichidagi tarkibdan iborat. Obekt mehmonxona o‘zgaruvchisida saqlanadi, shuning uchun biz uni mehmonxona obekti deb ataymiz.

Ikki nuqta kalit va qiymat o‘rtasida ajratuvchi sifatida ishlatiladi.

Vergul xususiyatlari va usullar orasiga qo‘yiladi (lekin oxirgi qiymatdan keyin vergul qo‘yilmaydi).

```

Var mexmonxona ={
    nomi: 'Toshkent',
    xonalar_soni: 56,
    bron_qilingan: 25,
}
mavjud_xonalarni_tekshirish: function(){
    return this.xonalar_soni-this.bron_qilingan;
}

```

mavjud_xonalarni_tekshirish() usuli ushbu obektning xonalari va bron qilingan xususiyatlaridan foydalanishini ko‘rsatish uchun this kalit so‘zidan foydalanadi.

Xususiyatlarni o‘rnatishda biz ularning qiymatlari bilan o‘zgaruvchilar bilan ishlaganimizdek ishlaymiz: satrlar qo’shtirmoq ichiga, massivlar esa kvadrat qavs ichiga olinadi.

Obektning xossalari yoki usullariga nuqta belgisi yordamida murajat qilish mumkin. Bundan tashqari, xususiyatlarga kvadrat qavslar yordamida kirish mumkin.

Obekt xususiyat nomi/usul

```
var mexmonxona_nomi = mexmonxona.nomi;
```

```
var bosh_xona = mexmonxona.mavjud_xonalarni_tekshirish
```

Kvadrat qavs sintaksi obekt xususiyatlariga murajat qilish uchun ham ishlataladi (lekin uning usullariga emas).

```
var mexmonxona_nomi = mexmonxona['nomi'];
```

Harfli belgilardan foydalanib obektlarni yaratish

Ushbu misolning birinchi qismida biz harfli(literal) yozuvdan foydalangan holda obekt yaratamiz.

Obekt mexmonxona deb ataladi va Toshkent mehmonxonasini ifodalaydi, unda 75 ta xona mavjud (va ulardan 42 tasi bron qilingan).

Keyinchalik, sahifa tarkibini ushbu obekt ma'lumotlari bilan yangilaymiz: biz mexmonxona nomini va mavjud xonalar sonini ko'rsatamiz.

Mexmonxona nomini bilish uchun biz nomi xususiyatiga murajat qilamiz va **mavjud_xonalarni_tekshirish()** usuli yordamida mavjud xonalar sonini aniqlaymiz.

Ushbu obektning xususiyatiga murojat qilish uchun uning nomidan keyin nuqta qo'yiladi, keyin esa xususiyat nomi yoziladi. Xuddi shunday usuldan foydalanish uchun ham obekt nomini olish, undan keyin nuqta qo'yish va keyin usul nomini yozish kerak.

```
mexmonxona.mavjud_xonalarni_tekshirish()
```

Agar usul parametrlarni talab qilsa, ular qavslar ichida ko'rsatilishi kerak (funksiya argumentlari kabi).

Misol.

HTML	CSS	JavaScript
<pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <title></title> </head> <body> <div class="wrap"> <div class="content"> <p>mexmonhana ma'lumoti</p> <p id='nomi'></p> <p id='xona'></p> <p>ta
 Xonalar qoldi</p> </div> </div> </body> </html></pre>	<pre>* { color: white; font-family: sans-serif; padding: 5px; text-align: center; } .wrap { background: black; height: 250px; width: 200px; } #name { border-top: 1px solid #fff; border-bottom: 1px solid #fff; text-transform: uppercase; } #room { color: #ff9933; font-size: 50px; margin: 0px; padding: 0px; }</pre>	<pre><script type="text/javascript"> var mexmonxona = { nomi: 'Toshkent', xonalar_soni: 75, qolgan_xonalar_soni: 42, mavjud_xonalarni_tekshirish: function() { return this.xonalar_soni - this.qolgan_xonalar_soni; } }; var nomi = document.getElementById('nomi'); nomi.textContent = mexmonxona.nomi; var xona = document.getElementById('xona'); xona.textContent = mexmonxona.mavjud_xonalarni_teks_hirish(); </script></pre>

Obekt yaratishda konstruktorda foydalanish

new kalit so‘zi va obekt konstruktori bo‘sh obektni yaratadi. Keyin unga usullar va xususiyatlar qo‘shiladi.

Birinchidan, yangi obekt yaratiladi: bu **new** kalit so‘zi va **Object()** konstruktor funksiyasi yordamida amalga oshiriladi. U JavaScript tilining bir qismidir va obektlarni yaratish uchun ishlataladi. Obekt yaratilgandan so‘ng, siz nuqta belgilaridan foydalanib, unga xususiyatlar va usullarni qo‘shishingiz mumkin. Obektga usul yoki xususiyat qo‘shadigan har bir ko‘rsatma nuqta-vergul bilan tugaydi.

Ushbu sintaksis siz yaratgan har qanday obektga xususiyatlar va usullarni qo‘shish uchun ishlatalishi mumkin (u qanday belgi(notatsiya) bilan yaratilganligidan qat’iy nazar).

Bo‘sh obekt quyidagi kabi harfiy belgilar yordamida yaratiladi:

```
var mexmonxona = {}
```

Figurali qavslar ichida bo‘sh obekt yaratiladi.

```

var mexmonxona = new Object();
mexmonxona.nomi: 'Toshkent',
mexmonxona.xonalar_soni: 56,
mexmonxona.bron_qilingan: 25,
}
mexmonxona.mavjud_xonalarni_tekshirish: function(){
return this.xonalar_soni-this.bron_qilingan;
}

```

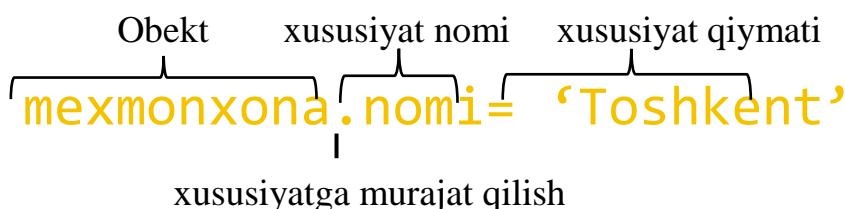
The code snippet defines an object named `mexmonxona`. It contains three data properties: `nomi`, `xonalar_soni`, and `bron_qilingan`. It also contains one method `mavjud_xonalarni_tekshirish`. Braces on the right side of the code group the properties into two categories: `xossalari` (values) and `usullari` (methods).

11.2. JavaScript obekt xususiyatlari

Xususiyat qiymatlarini yangilash uchun nuqta belgisi yoki kvadrat qavslardan foydalilanadi. Ushbu sintaksis bilan obekt ustida ishlash belgili yoki konstruktor belgisi yordamida amalga oshiriladi. Xususiyatlarni o‘chirish uchun delete kalit so‘zidan foydalaning.

Xususiyatni yangilash uchun biz oldingi sahifada xususiyatlarni qo‘shtganda ishlatgan texnikadan foydalanamiz. Biroq, xususiyat yangilanganda, u yangi qiymatga ega bo‘ladi.

Agar obekt siz yangilamoqchi bo‘lgan xususiyatga ega bo‘lmasa, u obektga qo‘shiladi.



Kvadrat qavs sintaksisi, obekt xususiyatlarini yangilash imkonini beradi (lekin usullarni yangilashga ruxsat bermaydi). Obyekt nomidan keyin kvadrat qavslar ishlatiladi va bu qavslar ichida xususiyat nomi joylashgan.

Yangi mulk qiymati tayinlash operatoridan(=) keyin joylashtiriladi.

`mexmonxona['nomi'] = 'Toshkent'`

Demak. obektlarga murajat qilish quyidagi sintaksislар asosida amalga oshiriladi:

```
// obekt nomi.xususiyati  
let yosh = shaxs.yosh;
```

yoki

```
//obekt nomi["xususiyat"]  
let yosh = shaxs["yosh"];
```

yoki

```
//obekt nomi[ifodalar]  
let yosh = shaxs[x];
```

Obektga murajat qilishning 1-usuli:

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript xususiyatlari</h1>  
<p id="demo"></p>  
<script>  
const shaxs = {  
    ismi: "Aziz",  
    familiya: "Karimov",  
    yosh: 35,  
};  
document.getElementById("demo").innerHTML = shaxs.ismi + " " +  
shaxs.familiya + " " + shaxs.yosh + " yosh.";  
</script>  
</body>  
</html>
```

index.html

Natija

JavaScript xususiyatlari

Aziz Karimov 35 yosh.

1.1-rasm. obektga murajat qilish(1-usul)

Obektga murajat qilishning 2-usuli:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Object xususiyatlari</h1>
<p id="demo"></p>
<script>
const shaxs = {
    ism: "Malika",
    familiya: "Abduhalilova",
    yosh: 28,
};
document.getElementById("demo").innerHTML = shaxs["ism"] + " is " +
shaxs["yosh"] + " yosh.";
</script>
</body>
</html>
```

index.html

Natija

JavaScript Object xususiyatlari

Malika is 28 yosh.

1.2-rasm. obektga murajat qilish(2-usul)

Obektga murajat qilishning 3-usuli:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScriptda obekt xususiyatlari</h1>
<p id="demo"></p>
<script>
const shaxs = {
    ism: "Aziz",
    familiya : "Karimov",
    yosh : 26
};
let x = "ism";
let y = "yosh";
document.getElementById("demo").innerHTML = shaxs[x] + " is " + shaxs[y]
+ " yosh.";
</script>
</body>
</html>
```

index.html

JavaScriptda obekt xususiyatlari

Aziza is 26 yosh.

1.3-rasm. obektga murajat qilish(3-usul)

Xususiyatlarni o‘zgartirish, qo‘shish, o‘chirish va ba’zilari faqat o‘qish mumkin.

Xususiyatni o‘chirish

Xususiyatni o‘chirish uchun **delete** kalit so‘zidan keyin obyekt nomi va xususiyat nomidan foydalaniladi. Delete operatori xususiyat qiymatini ham, xususiyatning o‘zini ham o‘chiradi. O‘chirilgandan so‘ng, xususiyat qayta qo‘shilishidan oldin foydalanilmaydi.

O‘chirish operatori obekt xususiyatlarida foydalanish uchun mo‘ljallangan. Bu o‘zgaruvchilar yoki funksiyalarga ta’sir qilmaydi.

Misol:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Obekt xususiyatlarini o'chirish</h1>
<p id="demo"></p>
<script>
const shaxs = {
    ism: "Maxbuba",
    familiya: "Shoraimova",
    yosh: 50,
};
delete shaxs.yosh;
document.getElementById("demo").innerHTML =
shaxs.ism + " " + shaxs.yosh + " yosh.";
</script>
</body>
</html>
```

index.html

NATIJA

JavaScript Obekt xususiyatlarini o‘chirish

Maxbuba undefined yosh.

1.4-rasm. Obekt xususiyatlarini o‘chirish

Yangi xususiyatlarni qo‘shish

Siz shunchaki qiymat berish orqali mavjud obektga yangi xususiyatlar qo‘shishingiz mumkin:

```
shaxs.millati = "Uzbek";  
  
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScriptda Obektga xususiyatlarni  
qo‘shish</h1>  
<p id="demo">dd</p>  
<script>  
const shaxs = {  
    ism: "Maxbuba",  
    familiya: "Shoraimova",  
    yosh: 50,  
};  
shaxs.millat = "uzbek";  
  
document.getElementById("demo").innerHTML =  
shaxs.ism + " " + shaxs.yosh + " yoshda millati  
" + shaxs.millat;  
</script>
```

index.html

natija

JavaScriptda Obektga xususiyatlarni qo‘shish

Maxbuba 50 yoshida millati uzbek

1.5-rasm. Obekt xususiyatlarini o‘chirish

11.3. Ko‘p obektlarni yaratish. Konstruktur belgisi

Bazan bir nechta obektlar o‘xshash narsalarni ifodalashini xohlaysiz. Obekt konstruktorlari obektlarni yaratish uchun funksiyadan shablon sifatida foydalanishlari mumkin. Avval obyekt xossalari va usullari bilan shablon yaratamiz. Mehmonxona funksiyasidan har biri mehmonxonaga mos keladigan obyektlarni yaratish uchun shablon sifatida foydalaniladi. Barcha funksiyalar singari, u ko‘rsatmalarni o‘z ichiga oladi.

Bunday holda, ko‘rsatmalar obektga xususiyatlar va usullarni qo‘shadi. Bu funksiya uchta parametrga ega. Har bir parametr obekt xususiyati uchun qiymatni belgilaydi. Ushbu funksiya yordamida yaratilgan har bir obekt uchun usullar bir xil bo‘ladi.

```
function mexmonxona(nomi, xonalar_soni, bron_qilingan)
```

```
{
    this.nomi: 'Toshkent',
    this.xonalar_soni: 56,
    this.bron_qilingan: 25,
}

this.mavjud_xonalarni_tekshirish = function()
{
    return this.xonalar_soni-this.bron_qilingan;
};}
```

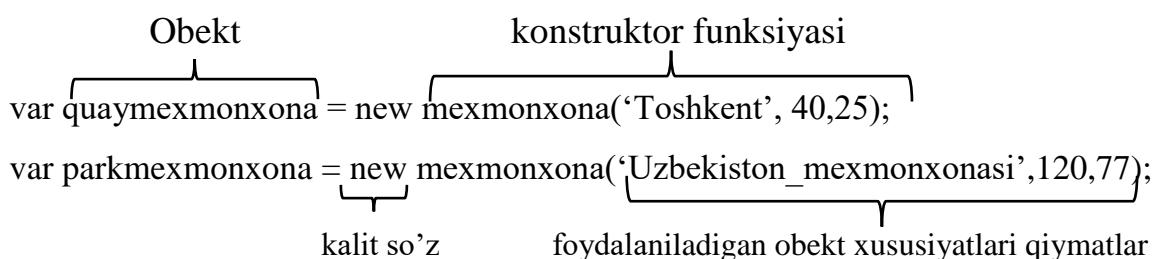
this kalit so‘z obekt nomi o‘rniga xususiyat yoki usulning ushbu aniq funksiya tomonidan yaratilgan obektga tegishli ekanligini ko‘rsatish uchun ishlataladi.

Berilgan obektda yangi xususiyat yoki usul yaratuvchi har bir ko‘rsatma nuqtali vergul bilan tugaydi.

Konstruktor funksiyasining nomi odatda bosh harf bilan boshlanadi (boshqa funksiya nomlari odatda kichik harflar bilan yoziladi).

Quyidagi misolda obekt namunalari konstruktor funksiyasi yordamida yaratiladi. Yangi obekt yaratish uchun new kalit so‘zidan keyin funksiya chaqiriladi. Har bir obektning xususiyatlari funksiyaga argument sifatida uzatiladi, bu yerda ikkita obekt ikkita mehmonxonaga mos keladi. new kalit so‘zi konstruktor funksiyasini chaqirganda (oldingi sahifada belgilangan), yangi obekt yaratiladi.

Har safar ushbu funksiya chaqirilganda, uning argumentlari har xil bo‘ladi, chunki ular har bir mehmonxonaning xususiyat qiymatlarini ifodalaydi. Ikkala mehmonxona obekti avtomatik ravishda konstruktor funksiyasida belgilangan bir xil usullarni oladi.



Birinchi obekt **quaymexmonxona** deb nomlangan. Ushbu mehmonxonaning nomi “Toshkent” bo’lib, unda 40 ta xona mavjud, ulardan 25 tasi bron qilingan. Bitta konstruktor funksiyasi yordamida bir nechta ob’ektlar yaratilgan bo’lsa ham, usullar o’zgarmaydi, chunki ular xususiyatlarda saqlangan ma'lumotlarga kirishadi, yangilaydilar yoki hisob-kitoblarni bajaradilar.

Ikkinci obekt parkmexmonxona deb nomlangan. Ushbu mehmonxonaning nomi “Uzbekiston_mexnomxonasi” bo’lib, unda 120 ta xona mavjud, ulardan 77 tasi bron qilingan. Ushbu usul skriptingiz mavjud bo’lishi kerak bo’lgan, ammo ishlatilmasligi mumkin bo’lgan juda murakkab obektni o’z ichiga olganida foydalidir. Obekt funksiyada aniqlanadi, lekin faqat kerak bo’lganda yaratiladi.

Global javascript obektlari

Ular tilda yaratilishi kerak bo’lgan obektlarni ifodalash uchun xizmat qiladi. Misol uchun, sana va vaqt qiymatlari bilan ishlash uchun maxsus mo’ljallangan obekt mavjud.

Ma'lumki, obekt mavjud ma'lumotlarga asoslanib, ma'lum bir real dunyo hodisasining modelini yaratish uchun ishlatilishi mumkin.

Global JavaScript obektlari bitta modelni yaratmaydi. Ular tilning turli qismlariga tegishli bo’lgan alohida obektlar to‘plamidir.

Global obektlarning nomlari odatda katta harf bilan boshlanadi — masalan, String va Date.

Asosiy obektlarni ifodalovchi obektlarig ma'lumot turlari:

STRING	String qiymatlari bilan ishlash uchun
NUMBER	Raqamli qiymatlari bilan ishlash uchun
BOOLEAN	Mantiqiy qiymatlari bilan ishlash uchun

Hodisalar bilan ishlashga yordam beradigan obektlar

DATA	Sanalarni taqdim etish va ular bilan ishlash uchun
MATH	Math - raqamlari va hisob-kitoblari bilan ishlash uchun
REGEX	Matn satrlarini shablonlarga moslashtirish uchun

String obektining toUpperCase() usuli quyidagi o‘zgaruvchidagi barcha harflarni bosh harf bilan yozadi:

```
hotel.toUpperCase();
```

Math obyektining PI xususiyati p sonini qaytaradi:

```
Math.PI();
```

11.4. O‘rnatilgan obektlar

Brauzer oyna yoki veb-sahifa kabi obektlarni ifodalovchi o‘rnatilgan obektlar to‘plamidan foydalanadi. O‘rnatilgan obektlarni interaktiv veb-sahifalarni yaratish vositalari sifatida ko‘rish mumkin.

Odatda, siz yaratgan obektlar aynan siz duch kelgan muammolarni hal qilish uchun dasturlashtiriladi. Ular skriptingizda ishlataladigan ma’lumotlarni modellashtiradi yoki uni ishlashi uchun zarur bo‘lgan funksiyalarni o‘z ichiga oladi.

O‘z navbatida, o‘rnatilgan obektlar ko‘plab senariylarning ishlashi bilan bog‘liq bo‘lgan funksionallikni o‘z ichiga oladi.

Brauzerde veb-sahifa yuklangandan so‘ng, siz ushbu obektlarni skriptlaringizda ishlatishingiz mumkin.

O‘rnatilgan obektlar sizga turli xil ma’lumotlarni aniqlash imkonini beradi, masalan: brauzer oynasining kengligi, sahifadagi asosiy sarlavha mazmuni, foydalanuvchi tomonidan shakl maydoniga kiritilgan matnning uzunligi va boshqalar.

Siz o‘zingiz yozgan obektning xususiyatlari va usullariga murajat qilishda bo‘lgani kabi, nuqta belgilaridan foydalanib, o‘rnatilgan obektlarning xususiyatlari va usullariga kirasisiz.

Obekt modeli nima

Obekt modeli - bu obektlar guruhi bo‘lib, ularning har biri bizni o‘rab turgan dunyoda mavjud bo‘lgan o‘zaro bog‘liq narsalarning modellaridan birini ifodalaydi. Bu kichik modellarning barchasi birgalikda katta tizimning modelini tashkil qiladi.

Boshqa obekt obektning mulki sifatida harakat qilishi mumkin. Agar bir obekt boshqa obekt ichida joylashgan bo'lsa, unda bunday ichki obekt bola yoki avlod obekti deb ataladi.

O'rnatilgan obektlardan foydalanish

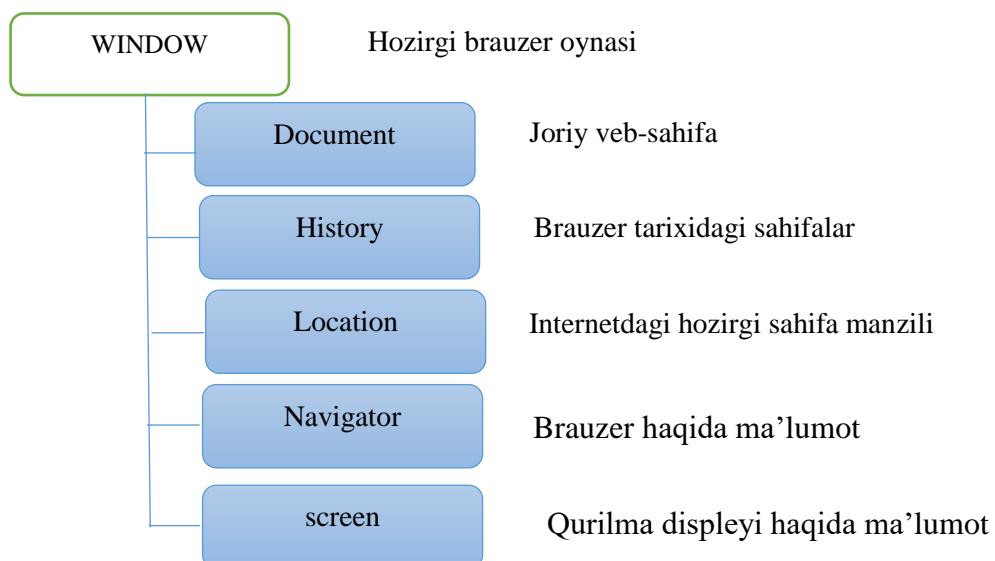
O'rnatilgan obektlarning uchta guruhi mavjud bo'lib, ularning har biri veb-sahifalar uchun skriptlarni yozishda yordam beradigan o'z vositalari to'plamiga ega.

Hujjat ob'ekt modelini sahifaga kirish va uning mazmunini yangilash uchun kerak.

Brauzer obekti modeli

Brauzer obekti modeli brauzer yorlig'i yoki oynasining ko'rinishini yaratadi.

Ushbu modelning yuqori qismida joriy oyna yoki brauzer yorlig'ini ifodalovchi window obekti joylashgan.



Misollar:

Oyna obektining `print()` usuli brauzerde chop etish dialog oynasini ko'rsatadi:

```
window.print();
```

Ekran obektining `width` xususiyati qurilma ekranining kengligini piksellarda belgilaydi:

```
window.screen.width;
```

Brauzer obekti modeli: window obekti

Window obekti hozirda ochilgan brauzer oynasini yoki yorlig‘ini anglatadi. U hujjatning brauzer obekti modelining yuqori qismida joylashgan va brauzerde ishslash bilan bog‘liq bo‘lgan boshqa obektlarni o‘z ichiga oladi.

Quyidagi jadvalda window obektining xususiyatlari va usullari keltirilgan. Bundan tashqari, unda ekran va boshqa obektlarining ba’zi xususiyatlari mavjud (ular window obektining avlodlari).

Xususiyat	Tavsifi
window.innerHeight	Oyna balandligi
window.innerWidth	Oyna kengligi
window.pageXOffset	hujjatni gorizontal ravishda aylantirilgan masofasi(px)
window.pageYOffset	hujjatni vertikal ravishda aylantirilgan masofasi(px)
window.screenX	X o‘qi bo‘yicha ko‘rsatgich koordinatasi, (ekranning yuqori chap burchagidan (piksellarda))
window.screenY	Y o‘qi bo‘yicha ko‘rsatgich koordinatasi, (ekranning yuqori chap burchagidan (piksellarda))
window.location	Oyna ob'ektining joriy URL manzili (yoki mahalliy fayl yo'li)
window.document	Hujjat obektiga havola, oynada joylashgan sahifani ko‘rsatish uchun ishlataladi.
window.history	Ushbu oynada yoki yorliqda ko‘rilgan sahifalar haqidagi ma'lumotlarni o‘z ichiga olgan brauzer oynasi yoki yorlig‘ining tarixi obektiga havola
window.screen	Ekran ob'ektiga havola
window.screen.width	Kenglik(piksel) xususiyatini topish uchun ekran obektiga murojaat qilish
window.screen.height	Balandlik (piksellarda) xususiyatini topish uchun ekran obektiga murojaat qilish

Metodlar	Tavsifi
window.alert()	Xabarli dialog oynasini yaratish (foydanuvchi ushbu Oynani yopish uchun “OK” tugmasini bosishi kerak)
window.open()	Parametr sifatida ko‘rsatilgan URL bilan yangi brauzer oynasini ochish. Agar brauzerde oynalarni blokirovka qiluvchi dastur o‘rnatilgan bo‘lsa, unda bu usul ishlamasligi mumkin.
window.print()	Brauzerga foydalanuvchi joriy sahifaning mazmunini chop etishni xohlayotgani haqida

xabar berish (brauzerning foydalanuvchi interfeysidagi chop etish buyrug‘iga o‘xshab ishlaydi)

Brauzer obyekti modelidan foydalanish

Brauzer ma’lumotlar window obektida va uning avlodlaridan to‘planadi, msg o‘zgaruvchisida saqlanadi va sahifada ko‘rsatiladi.

+ = operatsiyasi msg o‘zgaruvchisi oxirida yangi ma’lumotlarni qo‘shadi.

1. Windows obektining ikkita xususiyati, innerWidth va innerHeight, brauzer oynasining kengligi va balandligini ko‘rsatadi.

Doimo obektning xususiyatiga kirish uchun uning nomi va xususiyat nomi orasiga boshqa nuqta qo‘ying, masalan: window.history.length.

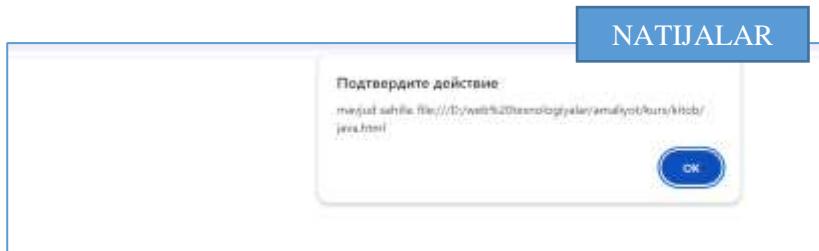
2. Avlod obektlari xususiyatlari ota-onada obektining xususiyatlari sifatida saqlanadi, shuning uchun ularga murajat qilish uchun nuqta belgisi ishlataladi.

3. Id atributi info qiymatiga ega bo‘lgan element ajratiladi va shu paytgacha shakllangan xabar hujjatga yoziladi.

Quyidagi misolda windows obekti xususiyatlari keltirilgan:

HTML

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title></title>
</head>
<body>
<p id="info"></p>
<script type="text/javascript">
    Var msg = '<h2>brauzer oyasi</h2><p>kengligi: ' +
    window.innerWidth + '</p>'
    Msg += '<p>Balandligi: ' + window.innerHeight + '</p>'
    Msg += '<h2>Elementlar </h2><p>tarixi: ' + window.history.length +
    '</p>'
    Msg += '<h2>Ekran</h2><p>kegligi: ' + window.screen.width + '</p>'
    Msg += '<p>baladlik: ' + window.screen.height + '</p>'
    Var el = document.getElementById('info')
    El.innerHTML = msg
    Alert('mavjud sahifa: ' + window.location)
</script>
</body>
</html>
```



brauzer oyasi

kengligi: 1536
Balandligi: 730

Elementlar

tarixi: 1

Ekran

kengligi: 1536
baladlik: 864

NAZORAT SAVOLLARI:

- 11.1. Obektlar qanday yaratiladi?
- 11.2. Obektga murtajat qanday amalga oshiriladi?
- 11.3. Brauzer obekti modelidan qanday foydalaniladi?
- 11.4. window.innerHeight xususiyatining vazifasi?
- 11.5. window.print() metodining vazifasi?
- 11.6. window.alert() metodining vazifasi?
- 11.7. Obekt modeli nima?
- 11.8. Xususiyatni o‘chirish qanday amalga oshiriladi?

XII BOB. DOM ELEMENTLARI. QUERY SELECTORLAR

12.1. Hujjat obekti modeli

DOM (Document Object Model), HTML, XHTML yoki XML hujjatlarini strukturaviy va ierarxik ravishda ta’riflaydigan platformaga mo’ljallangan. DOM, veb-sahifalarda joylashgan har bir elementni, ularning atrubutlarini va ular orqali HTML strukturasi bo‘yicha murojat qilish imkonini beradi. DOM veb-brauzerlarning bu hujjatlarga murajat qilish va ular bilan ishlashga ruxsat beradi.

DOM, HTML va XML hujjatlarni obyekt modeli sifatida ko‘rsatadi. Bu model brauzerlar va skript tillari, masalan JavaScript yordamida veb-saytlarning elementlariga, ularning xususiyatlariga va tarkibiga murojaat qilishga imkon beradi.

Masalan, agar HTML hujjatda bir “div” elementi bo‘lsa, DOM uni “div” elementi obyektiga aylantiradi. Bunday obyektlar orqali JavaScript yoki boshqa skript tillari HTML elementlariga murojaat qilish, ularni o‘zgartirish va yangi elementlar qo‘sish imkoniyatiga ega bo‘ladi.

DOM, veb-sahifalarni dinamik ravishda o‘zgartirish va boshqarish, animatsiyalar yaratish, veb-sahifalarda interaktivlik yaratish va ko‘p boshqa funksiyalarni amalga oshirishga yordam beradi.

Modelning yuqori qismida butun sahifani ifodalovchi hujjat obyekti joylashgan.

Misollar.

Document obyektining getElementById() usuli elementni id atributining qiymati bo‘yicha oladi:

```
document.getElementById('bir');
```

Document obektining lastModified xususiyati sahifaning so‘nggi yangilanish sanasini bildiradi:

```
document.lastModified;
```

Demak, Document Object model (DOM) brauzerlar HTML sahifasini qanday modellashtirishi kerakligini va JavaScript tili uning tarkibiga qanday murajat qilishi va brauzer oynasida bo‘lganida uni qanday yangilashi mumkinligini tasvirlaydi.

DOM HTML yoki JavaScriptning bir qismi emas. Bu alohida qoidalar to‘plami. Hujjat obekt modeli barcha asosiy brauzerlarda qo‘llaniladi va ikkita asosiy muammolar to‘plamini hal qiladi.

Html sahifa modelini yaratish

Brauzer veb-sahifani yuklaganida, xotirada uning modelini yaratadi.

DOM brauzer dom daraxti deb ataladigan model yordamida ushbu modelni tuzish usulini belgilaydi.

DOM obekt modeli deb ataladi, chunki u (DOM daraxti) obektlardan iborat.

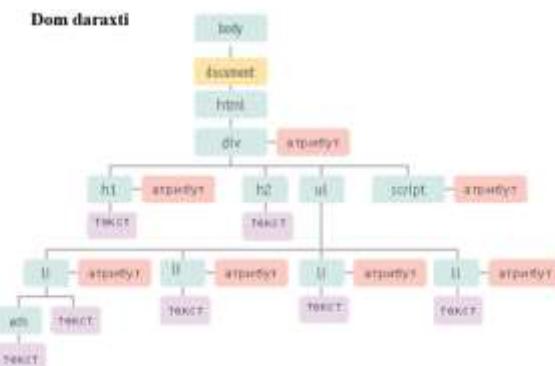
Har bir obekt brauzer oynasida yuklangan sahifaning alohida komponentiga mos keladi.

Html sahifaga murojat qilish va o‘zgartirish

DOM shuningdek, individual model obektlariga murajat qilish va yangilash imkonini beruvchi usullar va xususiyatlarni belgilaydi. O‘z navbatida, bu o‘zgarishlar foydalanuvchi brauzer oynasida ko‘radigan tarkibda aks etadi.

Dasturchilar ba’zan DOMni amaliy dasturlash interfeysi (API) deb atashadi. Foydalanuvchi interfeyslari odamlarga dasturlar bilan muloqot qilish imkonini beradi, API esa dasturlar va skriptlarga bir-biri bilan ma’lumot almashish imkonini beradi. DOM sizning skriptingiz brauzerdan joriy sahifa haqida qanday ma’lumotlarni so‘rashi mumkinligini va u tarkibga qanday o‘zgartirishlar kiritish kerakligini qanday aytib berishini boshqaradi.

Dom daraxti veb-sahifa modelidir. Brauzer veb-sahifani yuklaganida, uning modelini yaratadi. Ushbu model dom daraxti deb ataladi, u brauzer xotirasida saqlanadi. DOM daraxti to‘rtta asosiy tugun turidan iborat.



12.1-rasm. DOM daraxti tuzilishi

1.Hujjat tuguni

HTML hujjatidagi har bir element, atribut va matn qismi alohida DOM tuguni bilan ifodalanadi. Daraxtning yuqori qismida hujjat tugunlari joylashgan.

Har qanday element, atribut yoki matn tuguniga murajat qilganingizda, unga hujjat tugunlari orqali murajat qilasiz. Hujjat tuguni DOM daraxtining barcha murojaatlari uchun boshlang‘ich nuqtadir.

2.Element tugunlari

HTML elementlari veb-sahifaning tuzilishini tavsiflaydi. Masalan, <h1>–<h6> elementlarida sarlavhalar, <p> elementlari matndagi paragraflarning boshi va oxirini belgilaydi va hokazo.

DOM daraxtiga murajat qilish uchun biz yangi elementlarni qidiramiz. Kerakli element topilgach, uning matn tugunlari va atribut tugunlariga kirish mumkin.

Keyin matn va atributlarga qanday murajat qilish va o‘zgartirish haqida gaplashamiz.

3.Matn tugunlari

Element tuguniga murojat qilish orqali siz uning ichida joylashgan matnga murojat qilishingiz mumkin. U alohida matn tugunida saqlanadi. Matn tugunlarining avlodlari yo‘q.

4.Atribut tugunlari

HTML elementlarining ochilish teglari atributlar bilan birga bo‘lishi mumkin. Har bir atribut DOM daraxtida alohida tugun bilan ifodalanadi.

Elementga murajat qilganingizda, element atributlarini o‘qish yoki o‘zgartirish imkonini beruvchi maxsus JavaScript usullari va xususiyatlaridan foydalilanadi. Masalan, sahifadagi materialning taqdimotiga ta’sir qiluvchi yangi CSS qoidalarini o‘rnatish uchun siz tez-tez sinf atributlarining qiymatlarini o‘zgartirishingiz kerak bo‘ladi.

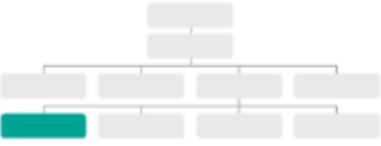
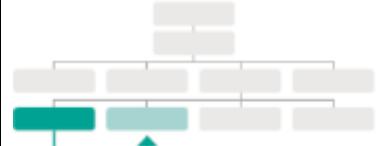
Dom daraxti bilan ishslash

DOM daraxtiga murajat qilish va uning tarkibini yangilash ikki bosqichda amalga oshiriladi:

- 1) biz ishlamoqchi bo‘lgan elementga mos keladigan tugunni topamiz;
- 2) biz matnli tarkib, element va atributlardan foydalanamiz.

1-bosqich: elementlarga murajat qilish

Quyida elementlarga murajat qilish uchun ishlataladigan usullar va xususiyatlar haqida umumiy ma’lumot berilgan. Ushbu material 198-217-betlarda ko‘rib chiqilgan. Dastlabki ikkita ustunda DOM so‘rovlari ro‘yxati keltirilgan. Oxirgi ustunda domni chetlab o‘tish (elementlar o‘rtasida harakatlanish) uchun mo‘ljallangan usullar va xususiyatlar mavjud.

Alohibda element tugunini tanlash	Bir nechta elementlarni tanlash (tugunlar ro‘yxati)	Elementlar tugunlari orasidagi daraxtga o‘tish
 <p><i>getElementById()</i> Usul kerakli element identifikatorini o‘qish belgisidan foydalanadi (barcha identifikatorlar — ID atributlari — sahifada unikal bo‘lishi kerak).</p> <p><i>querySelector()</i> Usul CSS selektoridan foydalanadi va uni o‘z ichiga olgan birinchi elementni qaytaradi.</p>	 <p>Bir nechta elementlarni tanlashning uchta umumiy usuli mavjud.</p> <p><i>getElementsByClassName()</i> Usul ma’lum bir sınıf atributti qiymatiga ega bo‘lgan barcha elementlarni tanlaydi.</p> <p><i>getElementsByTagName()</i> Usul belgilangan teg nomiga ega bo‘lgan barcha elementlarni tanlaydi.</p> <p><i>querySelectorAll()</i> Usul o‘z ichiga olgan barcha elementlarni tanlash uchun CSS selektoridan foydalanadi.</p>	 <p>Siz bir element tugunidan ubilan bog‘langan boshqasiga o‘tishingiz mumkin.</p> <p><i>parentNode</i> Ushbu element tugunining asosiy tugunini tanlaydi (mos ravishda faqat bitta element qaytariladi).</p> <p><i>previousSibling / nextSibling</i> B Dom daraxtidagi ma’lumotlarga ulashgan oldingi yoki keyingi elementni tanlaydi.</p> <p><i>firstChild / lastChild</i> Berilgan elementning birinchi yoki oxirgi bola elementini tanlaydi</p>

DOM so‘rovi bitta elementni yoki tugunlar to‘plami bo‘lgan NodeList obektini qaytarishi mumkin.

Ba’zan siz faqat bitta aniq elementga murajat qilishingiz. Boshqa hollarda, siz elementlar guruhini ajratib ko‘rsatishingiz kerak, masalan, sahifadagi barcha h1 sarlavhalari yoki ma’lum bir ro‘yxatdagi barcha li elementlari.

Element tugunlari guruhlari

Agar usul bir nechta tugunni qaytarishi mumkin bo'lsa, u har doim tugunlar to'plami bo'lgan NodeList elementini qaytaradi (to'plamda so'rovga mos keladigan faqat bitta element bo'lsa ham). Ro'yxatdan o'zingizni qiziqtirgan elementni indeks bo'yicha tanlashingiz kerak (raqamlash massivdagi kabi 0 dan boshlanadi).

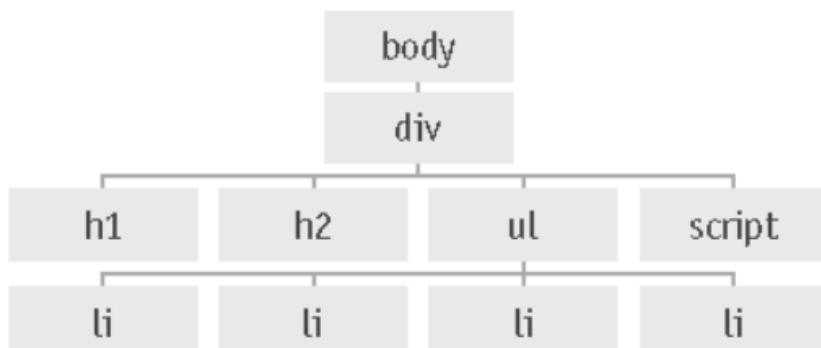
Misol uchun, bir nechta elementlar bir xil teg nomiga ega bo'lishi mumkin, shuning uchun getElementsByTagName() usuli har doim NodeList obyektini qaytaradi.

Eng qisqa marshurt

Agar siz sahifangizdagi elementga kirishning eng qisqa yo'lini topishni odat qilib qo'ysangiz, butun sayt tezroq va samaraliroq ko'rindi.

Odatda, bu siz ishlamoqchi bo'lgan elementga boradigan yo'lida tugunlarning minimal sonini taxmin qilishni talab qiladi. Misol uchun, getElementById() usuli elementni tezda qaytaradi, chunki sahifalarda ma'lum bir id atribut qiymatiga ega bo'lgan ko'pi bilan bitta element mavjud. Biroq, bu usuldan faqat sizni qiziqtirgan element id atributiga ega bo'lsagina foydalanish mumkin.

Esingizda bo'lsa, tugunlar DOM daraxtidagi elementlarning ko'rinishidir.



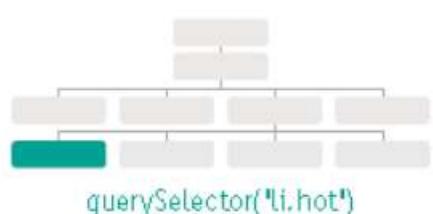
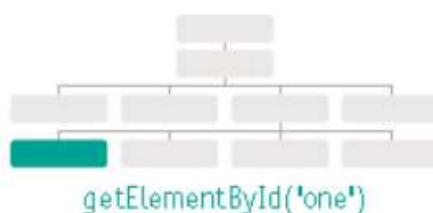
Faqat bitta element tuguni qaytarish usullari

getElementById('identifikator')

Muayyan elementni uning id atributining qiymatiga qarab tanlaydi. Siz izlayotgan HTML kontenti ushbu identifikatorga ega bo'lishi kerak. Brauzerni qo'llab-quvvatlashi: Internet Explorer 5.5 yoki undan yuqori, Opera 7 yoki undan yuqori, Chrome, Firefox, Safari barcha versiyalari.

querySelector('CSS selektori')

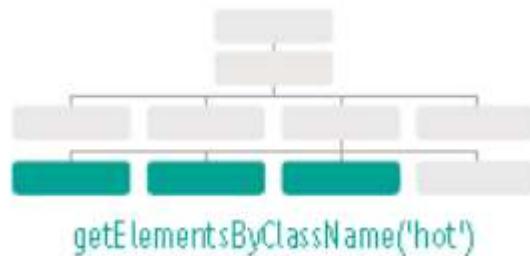
Bir yoki bir nechta elementni tanlash uchun CSS selektor sintaksisidan foydalanadi. Bu usul so'rovga mos keladigan natijalarning faqat birinchisini qaytaradi. Brauzerni qo'llab-quvvatlashi: Internet Explorer 8, Firefox 3.5, Safari 4, Chrome 4, Opera 10 yoki undan yuqori.



Bir yoki ko‘p elementlarni qaytarish usullari (tugun ro‘yxati ob’yeklari sifatida)

getElementsByClassName(klass)

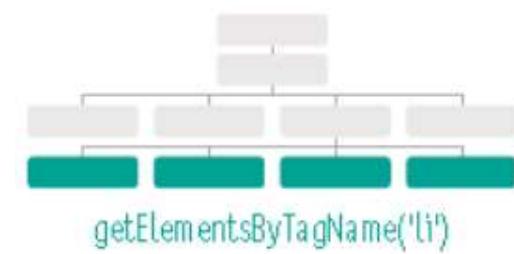
Bir yoki bir nechta elementni klas atributining qiymati bo‘yicha tanlaydi. Siz izlayotgan HTML kontenti sinf atributiga ega bo‘lishi kerak. Bu usul querySelectorAll() dan tezroq ishlaydi. Brauzerni qo‘llab-quvvatlashi: Internet Explorer 9, Firefox 3, Safari 4, Chrome 4, Opera 10 yoki undan yuqori versiyalar (ba’zi brauzerlarning oldingi versiyalarida bu xususiyat qisman yoki noto‘g’ri qo‘llab-quvvatlanadi).



getElementsByTagName('teg nomi')

Belgilangan teg nomiga ega sahifadagi barcha elementlarni tanlaydi. Bu usul querySelectorAll() dan tezroq ishlaydi..

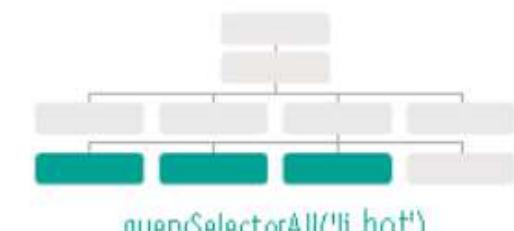
Brauzerni qo‘llab-quvvatlashi: Internet Explorer 6, Firefox 3, Safari 4, Chrome 4, Opera 10 yoki undan yuqori versiyalar (ba’zi brauzerlarning oldingi versiyalarida bu xususiyat qisman yoki noto‘g’ri qo‘llab-quvvatlanadi).



querySelectorAll('teg nomi')

CSS selektor sintaksisidan foydalanadi, bir yoki bir nechta elementlarni tanlaydi va so‘rovga mos keladigan barcha elementlarni qaytaradi.

Brauzerni qo‘llab-quvvatlashi: Internet Explorer 8, Firefox 3.5, Safari 4, Chrome 4, Opera 10 yoki undan yuqori.



Ikki usul — getElementById () va querySelector () — butun hujjatni qidirishi va alohida elementlarni qaytarishi mumkin. Ikkalasi ham o‘xshash sintaksisdan foydalanadi.

getElementById() usuli elementga murojat qilishning eng tezkor va samarali usuli hisoblanadi, chunki ikkita element bir xil identifikator qiymatiga ega bo‘la olmaydi.

Ushbu elementning sintaksisi quyida ko‘rsatilgan:



`getElementById()` usuli elementni uning identifikatori (ID atributi) qiymati bo'yicha topmoqchi ekanligingizni bildiradi.

`Queryselector ()` usullari keyinchalik hujjatning obekt modelida paydo bo'ldi, shuning uchun nisbatan eski brauzerlarda qo'llab-quvvatlanmaydi. Ammo bu juda moslashuvchan, chunki uning parametri CSS selektoridir.

`querySelector()` JavaScriptning DOM (Document Object Model) elementlarini topish uchun ishlataladigan bir metoddir. Bu metod orqali, HTML hujjatida barcha elementlardan birini tanlash mumkin.

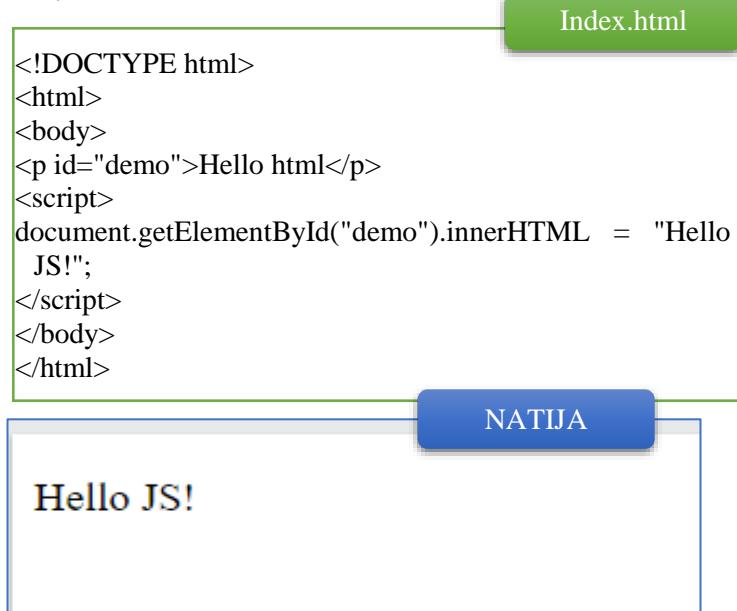
Metod sintaksisi quyidagicha:

document.querySelector(selector)

Selector - CSS sintaksisiga mos tushadigan elementni tanlash xususiyatlari. Bu, bir element nomi, sinf nomi, ID nomi, yoki boshqa tarkibiy CSS selektor bo'lishi mumkin.

"Document" so'zi hujjat obektiga ishora qiladi. Alovida elementlarga har doim document obekti orqali murajat qilish kerak.

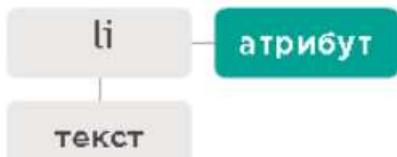
Shunday qilib, ushbu usul ko'plab elementlarni ta'kidlash uchun ishlatalishi mumkin.



12.2-rasm. "Document" obektiga namuna

2-bosqich: Elementlar bilan ishlash

Quyida elementlar bilan ishlashda qo'llaniladigan usullar va xususiyatlarning umumiy ko'rinishi keltirilgan.

Atribut qimmatlarga murajat qilish va yangilash	Html kontentlari bilan ishlash	Matn tugunlarini murajat qilish va yangilash
 <p>Har qanday element ichidagi matn matn tugunida joylashgan. Yuqorida ko'rsatilgan matn tuguniga murojat qilish uchun quyidagilarni bajaring:</p> <ol style="list-style-type: none"> 1. li elementini tanlang. 2. firstChild xossasidan foydalanib, matn tugunini oling. 3. Ushbu elementdan matn olish uchun matn tuguningning yagona xususiyatidan (nodeValue) foydalaning. 	 <p>Bola elementlari va matn tarkibiga kirish imkonini beruvchi bitta xususiyat mavjud:</p> <ul style="list-style-type: none"> innerHTML Boshqa xususiyat faqat matnli tarkibga kirish imkonini beradi: textContent Bir nechta usullar sizga yangi tugunlar yaratish, daraxtga tugunlar qo'shish va ularni olib tashlash imkonini beradi: createElement() createTextNode() appendChild() / removeChild() Bunday operatsiyalar DOM manipulyatsiyasi deb ataladi. 	 <p>Quyida atributlar bilan ishlashda foydalanish mumkin bo'lgan ba'zi xususiyatlar va usullar keltirilgan.</p> <ul style="list-style-type: none"> className/id class va id atributlarining qiymatlarini olish va o'zgartirish imkonini beradi. hasAttribute() getAttribute() setAttribute() removeAttribute() Birinchi usul atribut mayjudligini tekshiradi. Ikkinchisi o'z qiymatini oladi. Uchinchisi bu qiymatni yangilaydi. To'rtinchisi atributni olib tashlaydi.

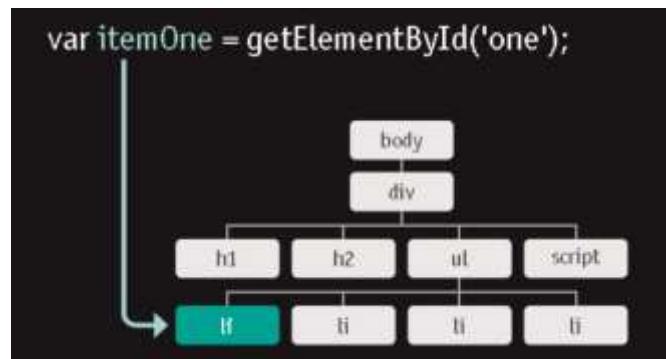
DOM so'rovlarini keshlash

DOM daraxtidagi elementlarni topadigan usullar DOM so'rovlari deb ataladi. Agar element bilan bir necha marta biror narsa qilish kerak bo'lsa, unda bunday so'rovning natijasi o'zgaruvchida saqlanishi kerak.

Elementlarni o'zgaruvchilarda saqlash haqida gapirganda, aslida dom daraxtidagi elementning holati o'zgaruvchiga o'tkazilishini anglatadi. Element tuguningning xususiyatlari va usullari ushbu o'zgaruvchi bilan ishlaydi.

Agar sizning skriptingiz bir xil elementlardan bir necha marta foydalanishni talab qilsa, ularning joylashuvi o'zgaruvchilarda saqlanishi mumkin.

Shunday qilib, brauzer vaqtini tejaydi, chunki u DOM daraxtini qayta ko‘rib chiqishi va allaqachon ishlatilgan elementlarni qayta topishga vaqt sarflamaydi. Ushbu amaliyot namunaviy keshlash deb ataladi.



itemOne o‘zgaruvchisi li elementini saqlamaydi, lekin bu tugun DOM daraxtida joylashgan pozitsiyaga havolani anglatadi. Berilgan elementning matn kontentiga murajat qilish o‘zgaruvchi nomidan foydalanishingiz mumkin:

itemOne.textContent

Misollar:

1-misol. Html elementini Id isi bo‘yicha topish

```
<body>
<h2>JavaScript HTML DOM</h2>
<p id="intro">Id bo'yicha HTML elementlarini topish</p>
<p>Ushbu misol
<b>getElementsById</b> usulini ko'rsatadi.</p>
<p id="demo">Salom</p>
<script>
const element =
document.getElementById("intro");
document.getElementById("demo").inn
erHTML =
"Kirish paragrafidagi matn: " +
element.innerHTML;
</script>
</body>
```

NATIJA

JavaScript HTML DOM

Id bo'yicha HTML elementlarini topish

Ushbu misol **getElementsById** usulini ko'rsatadi.

Kirish paragrafidagi matn: Id bo'yicha HTML elementlarini topish

12.3-rasm. Html elementini Id isi bo‘yicha topishga namuna

2-misol: Teg nomi bo‘yicha HTML elementlarini topish

Bu misol barcha <p> elementlarni topadi:

```

<!DOCTYPE html>
<html>
<body>
<h2>JavaScript HTML DOM</h2>
<p>Teg nomi bo'yicha HTML elementlarini topish.</p>
<p>Ushbu misol
<b>getElementsByTagName</b> usulini ko'rsatadi.</p>
<p id="demo"></p>
<script>
const element =
document.getElementsByTagName("p");
document.getElementById("demo").innerHTML
= 'Birinchi xatboshidagi matn (indeks 0): ' +
element[0].innerHTML;
</script>
</body>
</html>

```

Index.html

NATIJA

JavaScript HTML DOM

Teg nomi bo'yicha HTML elementlarini topish.

Ushbu misol **getElementsByTagName** usulini ko'rsatadi.

Birinchi xatboshidagi matn (indeks 0): Teg nomi bo'yicha HTML elementlarini topish.

12.4-rasm. Teg nomi bo'yicha HTML elementlarini topishga namuna

3-misol: Bu misol id=“main” li elementni va keyin “main” ichidagi barcha <p> elementlarni topadi:

```

<body>
<h2>JavaScript HTML DOM</h2>
<div id="main">
<p>Teg nomi bo'yicha HTML elementlarini topish</p>
<p>Ushbu misol <b>getElementsByTagName</b> usulini ko'rsatadi.</p>
</div>
<p id="demo"></p>
<script>
const x = document.getElementById("asosiy");
const y = x.getElementsByTagName("p");
document.getElementById("demo").innerHTML = 'Main ichidagi birinchi
xatboshi (indeks 0):' + y[0].innerHTML;
</script>
</body>

```

Index.html

JavaScript HTML DOM

Teg nomi bo‘yicha HTML elementlarini topish

Ushbu misol **getElementsByName** usulini ko'rsatadi.

12.5-rasm. Teg nomi bo‘yicha HTML elementlarini topishga namuna

4-misol. sinf nomi bo‘yicha HTML elementlarini topish

Agar siz bir xil sinf nomiga ega barcha HTML elementlarini topmoqchi bo‘lsangiz, **getElementsByClassName()** dan foydalaning.

Bu misol class=“intro” bilan barcha elementlar ro‘yxatini qaytaradi.

index.html

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript HTML DOM</h2>
<p>Sinf nomi bo‘yicha HTML elementlarini topish.</p>
<p class="intro">Salom dunyo!</p>
<p class="intro">Ushbu misol
<b>getElementsByClassName</b> usulini ko'rsatadi.</p>
<p id="demo"></p>
<script>
const x = document.getElementsByClassName("intro");
document.getElementById("demo").innerHTML =
'Birinchi paragraf (indeks 0) class= intro bilan: ' +
x[0].innerHTML;
</script>
</body>
</html>
```

JavaScript HTML DOM

Sinf nomi bo‘yicha HTML elementlarini topish.

Salom dunyo!

Ushbu misol **getElementsByClassName** usulini ko'rsatadi.

Birinchi paragraf (indeks 0) class= intro bilan: Salom dunyo!

12.6-rasm. sinf nomi bo‘yicha HTML elementlarini topishga namuna

12.2. HTML elementlarini o‘zgartirish va ularga namunalar

Xususiyat	Tavsifi
element.innerHTML = new html content	Elementning HTML-ni o‘zgartirish
element.attribute = new value	HTML elementining atribut qiymatini o‘zgartirish
element.style.property = new style	HTML elementining uslubini o‘zgartirish
Metodlar	Tavsifi
element.setAttribute(attribute, value)	HTML elementining atribut qiymatini o‘zgartirish

Elementlarni qo‘shish va o‘chirish

document.createElement(element)	Html elementini yaratish
document.removeChild(element)	Html elementini o‘chirish
document.appendChild(element)	Html elementini qo‘shish
document.replaceChild(new, old)	HTML elementini almashtirish
document.write(text)	HTML chiqish oqimiga yozish

Xodisalarni qo‘shish

Metod	Tavsifi
document.getElementById(id).onclick = function(){code}	Onclick xodisasiga kod yozish
element.attribute = new value	HTML elementining atribut qiymatini o‘zgartirish

Misollar:

5-misol. HTML tarkibini o‘zgartirish. HTML elementi tarkibini o‘zgartirishning eng oson yo‘li innerHTML xususiyatidan foydalanishdir.

```
document.getElementById(id).innerHTML = new HTML
```

Ushbu misol <p> elementining tarkibini o‘zgartiradi:

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript HTML-ni o'zgartirishi mumkin</h2>
<p id="p1">Salom dunyo!</p>
<script>
document.getElementById("p1").innerHTML = "Yangi matn!";
</script>
<p>Yuqoridagi paragraf skript bilan o'zgartirildi.</p>
</body>
</html>
```

NATIJA

JavaScript HTML-ni o'zgartirishi mumkin

Yangi matn!

Yuqoridagi paragraf skript bilan o'zgartirildi.

12.7-rasm. HTML tarkibini o'zgartirishga namuna

Yuqoridagi HTML hujjatida id= “p1” li <p> element mavjud. Biz id = “p1” elementini olish uchun HTML DOM dan foydalanamiz.

JavaScript ushbu elementning tarkibini (innerHTML) “Yangi matn!” ga o'zgartiradi.

Atribut qiymatini o'zgartirish

HTML atributining qiymatini o'zgartirish uchun ushbu sintaksidan foydalilanadi:

```
document.getElementById(id).attribute = yangi qiymat
```

Ushbu misol elementining src atributining qiymatini o'zgartiradi:

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript HTML DOM</h2>

<script>
document.getElementById("image").src = "gul.png";
</script>
<p>Asl rasm rasm.jpg edi, lekin skript uni gul.png qilib o'zgartirdi</p>
</body>
</html>
```

JavaScript HTML DOM



Asl rasm rasm.jpg edi, lekin skript uni gul.png qilib o'zgartirdi

12.8-rasm. Atribut qiymatini o'zgartirishga namuna

JavaScriptda document.write() to‘g‘ridan-to‘g‘ri HTML chiqish oqimiga yozish uchun ishlatalishi mumkin:

```
index.html
```

```
<body>
<script>
document.write(Date());
</script>
</body>
```

NATIJA

Tue Jun 04 2024 11:45:07 GMT+0500 (Узбекистан, стандартное время)

12.9-rasm. document.write()dan foydalanish bo‘yicha namuna

6-misol. JavaScript kiritilgan sonlarni raqamli ekanligini tekshirishi mumkin.

index.html

```
<body>
<h2>JavaScript-ni tekshirish</h2>
<p>Iltimos, 1 dan 10 gacha raqam kriting:</p>
<input id="numb">
<button type="button" onclick="myFunction()">Yuborish</button>
<p id="demo"></p>
<script>
function myFunction() {
// id="numb" bilan kirish maydonining qiymatini oling
let x = document.getElementById("numb").value;
// Agar x raqam bo'lmasa yoki bittadan kichik yoki 10 dan katta bo'lsa
let text;
if (isNaN(x) || x < 1 || x > 10)
{
text = "Kirish noto'g'ri"; }
else
{
text = "OK kriting"; }
document.getElementById("demo").innerHTML = text; }
</script>
</body>
```

NATIJA

JavaScript-ni tekshirish

Iltimos, 1 dan 10 gacha raqam kriting:

0

Yuborish

Kirish noto'g'ri

12.10-rasm. JavaScriptda namuna

HTML uslubini o'zgartirish

HTML elementining uslubini o'zgartirish uchun ushbu sintaksisdan foydalaning:

```
document.getElementById(id).style.property = new style
```

Misol-1. Quyidagi misol <p> elementining uslubini o'zgartiradi:

index.html

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript HTML DOM</h2>
<p> HTML stilini o'zgartirish</p>
<p id="p1">Hello World!</p>
<p id="p2">Hello World!</p>
<script>
document.getElementById("p2").style.color = "red";
document.getElementById("p2").style.fontFamily =
"Arial";
document.getElementById("p2").style.fontSize =
"larger";
</script>
</body>
</html>
```

NATIJA

JavaScript HTML DOM

HTML stilini o'zgartirish

Hello World!

Hello World!

12.11-rasm. HTML uslubini o'zgartirishga namuna

Xodisalardan foydalanish

HTML DOM xodisalari sodir bo'lganda kodni bajarishga imkon beradi.

Ushbu misolda, foydalanuvchi tugmani bosganida, HTML elementining uslubini id = “id1” bilan o‘zgartiradi:

The diagram illustrates the effect of changing an HTML element's ID. On the left, the `index.html` file contains the following code:

```
<!DOCTYPE html>
<html>
<body>
<h1 id="id1">Birinchi sarlavha</h1>
<button type="button"
onclick="document.getElementById('id1').style.color = 'red'">
Click Me!</button>
</body>
</html>
```

On the right, under the heading "NATIJALAR", two screenshots show the result. The top screenshot shows the heading "My Heading 1" in black text. The bottom screenshot shows the heading "My Heading 1" in red text, indicating that the CSS style has been applied due to the button's click event.

12.12-rasm. DOM hodisalaridan foydalanishga namuna

NAZORAT SAVOLLARI:

- 12.1.DOM ning vazifasi nima?
- 12.2.HTML elementlarini o‘zgartirish qanday amalga oshiriladi?
- 12.3.getElementsByClassName(klass) ninng vazifasi?
- 12.4.getElementsByTagName('teg nomi') ning vazifasi?
- 12.5.querySelectorAll('teg nomi')ning vazifasi?
- 12.6.getElementById('identifikator')ning vazifasi?
- 12.7.querySelector('CSS selektori')ning vazifasi?
- 12.8.Html atribut qiymatini o‘zgartirish qanday amalga oshiriladi?

XIII-BOB. WEB SAYTLARNI YARATISHDA CMS

TEXNOLOGIYASIDAN FOYDALANISH

13.1. CMS. Asosiy tushunchalar. CMS ning maqsad va vazifalari

CMS – bu Content Management System dan olingen kengaytma, u so‘zmaso‘z tarjima qilinganda “*Sayt kontentini boshqarish tizimi*” yoki “Saytni Boshqarish Tizimi” larni anglatadi. Ba’zida CMS ni saytning “Harakatlantiruvchisi” deb atashadi.

CMS – bu dinamik ma’lumotli web-sahifalarni ishlab chiqarish va ta’minalashga imkon beruvchi dasturiy ta’mindir. Har xil CMS har xil murakkablikdagi saytlarni proyektlashga imkon beradi. Internet magazinlardan to axborotli portallargacha, hammasidan ham, CMS axborotli va kontentli saytlarni yaratish uchun mos keladi.

CMS (inglizcha “Content Management System”) – sayt boshqaruv tizimi. Bu saytni tez yaratish ishini soddallashtirish uchun hizmat qiladi. Ko`pchilik buni “motor” yoki “sayt yuragi” deb ataydi. CMS bu oddiy qilib aytilganda – sayt ochishni va saytni tahrirlash, ma`lumotlar kiritishni yengillashtiruvchi yaxlit bir sistema ya`ni web dastur. Bu dastur php, html, mysql dasturlash tillarini to‘liq bilmasdan turib sayt ochish imkon beradi.

1. Saytni boshqarish tizimi buyurtmachiga saytni mustaqil expluatatsiya qilishda, ya’ni bizning mutaxassis bilan emas, balki o‘zingizning kompaniya xizmatchingiz (menedjer, kotib, marketolog) bilan saytdagi ixtiyoriy ma’lumotni va uning strukturasini o‘zgartirishingizga imkon beradi.
2. Saytni boshqarish tizimi yordamida saytni rivojlanishi sezilarli soddalashadi va jonlanadi. Saytda yangi bo‘lim yaratish, ixtiyoriy sahifada reklama bloklarini, ichki bannerlarni yoki tovarlarning maxsus takliflarini joylashtirishni mustaqil bajarishi mumkin. Bundan tashqari, Buyurtmachi mustaqil, bizning yordamimizsiz, CMS yordamida saytni yangi interfaol bo‘limlarini yaratishi (masalan, forum) yoki u yoki bu

sahifalarni parol bilan yopib va ularga faqat tanlangan foydalanuvchilargagina kirishga imkon berishi mumkin.

CMS ning ikkita asosiy turi bor:

1. Internetda ishlovchi va joylashuvchi CMS.
2. Kompyuterga o‘rnatilgan va sayt bilan avtomatik yoki fayllarni yangilash interfeysi orqali ulangan cms.

CMS ning sayt bilan integratsiyasidan keyin, kompaniyangizning xizmatchisi sayt tarkibi va strukturasini boshqarishlari, yangiliklarni chop qilishlari, matnlarni tahrir qilishlari va grafikani joylashtirishlari, katalogdagi mahsulotni o‘zgartirishlari va mahsulotlarni qo‘sishlari yangi sahifalarni, bo‘limlarni yaratishlari va ko‘pgina boshqa ishlarni bajarishlari mumkin. CMS asosida qurilgan dinamik saytlarning ustunligi dizaynning axborotli to‘ldirishdan ajratilganida bu xujjat aylanmasini, biznes jarayonlarni shaxsiy lashtirish maxanizmlarini avtomatlashtirishga imkon beradi. Bundan tashqari, CMS sayt tarkibini oson, sodda va tez boshqarishga imkon beradi..

CMS ning maqsad va vazifalari – mijoz tomonidan ma’lumot va grafikalarni boshqarish, to‘ldirish va tahrir qilishdir. Soddarоq qilib aytganda, bu tizim – sizga bo‘limlarni o‘zgartirish va to‘ldirish, sayt tarkibini tahrir qilish, sayt grafika va dizaynnini o‘zgartirishga imkon beradi.

CMS sayt kontentni boshqarish tizimi saytlarni yaratish va ta’minalash narxini pasaytiradi, shuningdek materiallarni yangilash tezligini oshiradi, ya’ni buyurtmachida material paydo bo‘lganidan to saytda chop etilganigacha ketgan vaqt minimal bo‘ladi. Buning ustiga CMS da ma’lumotlarni chop etishga tayyorlash jarayoni juda ham oddiy. CMS kontentni boshqarish tizimi sizga saytda ihtiiyoriy hajmdagi ma’lumotlarni joylashtirishga imkon beradi.

Odatda CMS ga asoslangan sayt sahifalari butunligicha saqlanmaydi, balki ularga murojat qilishda juda tez shakllanadi, bu sahifalarni yuklashdagi tezlikni oshirishga imkon beradi, chunki sahifa shabloni faqat bir marta yuklanadi va keyinchalik har bir yangi sahifa so‘rovida foydalanuvchi kompyuteridan ochilaveradi.

Saytni boshqarish tizimida odatda keshlashtirish mexanizmi amalga oshiriladi, bu qurilmalarga tushadigan yuklamalarni kamaytirishga imkon beradi va ishslashning va saytning kiritish qobiliyatining tezligi oshishiga imkon beradi. Bu ishlar ma'lumotlar bazasi va statik materiallarga so'rov natijalarini saqlash evaziga sodir bo'ladi.

CMS ning tekinlari ham mavjud, ular sayt tarkibini boshqarish uchun yetaricha katta imkoniyatlarni namoyish qiladi. Tekin CMSlar ko'p foydalanuvchilarga ega, ular tekin kontentni boshqarish tizimi yordamida saytni boshqarish muammolarini hal qilishga yordam beradi. Tekin CMS larga katta miqdordagi kengaytma va qo'shimchalar mavjud, ular saytni boshqarish jarayonini ancha qulay va funksional qilishga imkon beradi. Ammo tekin CMS larning salbiy tomoni rasmiy texnik qo'llab quvvatlashning yo'qligi va sayt tarkibini boshqarish tizimida kritik xatolarni to'g'rilash, tezligining paytligidadir.

CMSning afzalliklari va kamchiliklari mavjud.

Afzalliklari:

- Oddiylik va tezlik:** CMS, veb-tarkibni yaratish, o'zgartirish va boshqarishni osonlashtiradi. Bu, HTML yoki CSS bilan ishslashga nisbatan juda oson.
- Ko'plab funksiyalar:** Ko'p CMSlar, foydalanuvchilarga qulayliklar qo'shish uchun qo'shimcha modullar va pluginlar orqali keng qamrovli funksiyalar bilan ta'minlanadi.
- Ko'p foydalanuvchilar uchun qo'llab-quvvatlash:** Ko'plab odamlar birgalikda veb-sayt yoki veb-ilovalarni boshqarish uchun CMS ishlatishadi.
- Dinamik sahifalar:** CMS, dinamik veb-saytlarni yaratish imkonini beradi. Bular, maqbul saytlar, bloglar, forumlar va h.k. kabi dinamik tarkibni o'z ichiga oladi.

Kamchiliklari:

- Narxi:** Ba'zi CMSlar pulli bo'lishi mumkin. Shuningdek, qo'shimcha modullar va pluginlar uchun to'lov talab qilinishi mumkin.

2. **Ishonchli emaslik:** Ba’zi CMSlar to‘g‘ri tahrirlash va ushbu tahrirlarni saqlashga qodir emas. Shu sababli, xavfsizlik muammolari paydo bo‘lishi mumkin.
3. **Cheklanish:** Ba’zi CMSlar chet tillar va ushbu tildagi modullarni yuklashga mo‘ljallangan. Bu, ba’zi foydalanuvchilar uchun cheklangan yoki qiyinchilik yaratadigan CMSlar tanlashni talab qiladi.

Har bir CMS, xususan, joylashtirish, o‘rnatish, foydalanish va ma'lumotlarni boshqarishga qaratilgan maqsadlarga qarab taqdim etilgan afzallik va kamchiliklarga ega bo‘ladi.

13.2. CMS turlari

Quyida eng keng tarqalgan va bugungi kunda ko‘pchilik saytlarning asosi hisoblangan CMS dasturlarini keltirib o‘tiladi. Bu yerda faqat eng ko‘p foydalaniladigan, asosiy dasturlargina keltirilgan.

1. **WordPress.** Ushbu CMS asosan blog (biror shaxs, joy yoki alohida mavzuga bag‘ishlangan, tez-tez yangilab turiladigan, qisqa maqolalar va ma'lumotlar kiritib boriladigan o‘rta hajmdagi sayt) uchun mo‘ljallangan. Ammo bugungi kunga kelib bu CMS shu qadar mukammallahib ketdiki, hozirda uni hatto portal saytlar uchun ham ishlatishmoqda.

WordPress – foydalanish uchun juda oson, qulay va eng asosiysi 100% bepul bo‘lgan dastur. Uni **WordPress**ning rasmiy sayti www.wordpress.org orqali yuklab olishingiz va foydalanishingiz mumkin. Asosiy CMS dasturdan tashqari **WordPress** uchun millionlab shablonlar (sayt ko‘rinishlari) va pluginlar (dastur ichida ishlovchi kichik qo‘shimchalar) yaratilgan.

Endi **WordPress** CMSni o‘zbek tilida ham ishlatishingiz mumkin. Uning O‘zbekiston uchun maxsus rasmiy sahifasi ham bor: uz.wordpress.org

2. **Joomla!** Bu CMS asosan katta saytlar va portallar uchun mo‘ljallangan. Ommaviyligi bo‘yicha **WordPress** dan keyingi o‘rinda tursa ham, ishlatish juda qulay, ishonchli va bepul CMS hisoblanadi. Dasturning rasmiy sayti www.joomla.org orqali yuklab olishingiz mumkin.

Joomla! CMS uchun ham ko‘plab shablonlar, “extension” (kengaytma, pligin)lar yaratilgan. Joomla! uchun ham saytni o‘zbek tiliga o‘tkazish imkoniyati bor. Buning uchun dasturni o‘rnatib, o‘zbek tiliga o‘girish uchun maxsus tayyorlangan kichik dasturni qo‘siningiz kifoya!

3. Simple Machines. Forum tashkil etish uchun juda qulay CMS. Unda foydalanuvchilar ro‘yxatdan o‘tishlari, mavzular yaratishlari, mavjud mavzularda o‘z fikrlarini bildirish orqali qatnashishlari va boshqalarning fikrlarini to‘ldirishlari mumkin. Bu CMS vBulletin kabi katta imkoniyat va kuchga ega bo‘lmasa ham, oddiy darajadan yuqoriq bo‘lgan katta forum saytlarni bemalol boshqara oladi. Forum uchun o‘zbek tilini o‘rnatish imkoniyati mavjud, ammo hozircha o‘zbek tilidagi rasmiy CMS ishlab chiqilmagan. Tilni faqat dasturga qo‘sishcha kiritish orqali o‘zgartirish imkoni bor.

Bugungi kunda **Simple Machines** forum saytlar uchun yengil ishlashi, o‘rnatilishi uchun ko‘p joy talab qilmasligi va eng asosiysi bepul bo‘lganli sababli ham ko‘p sonli saytlarning tanloviga aylangan.

Simple Machines CMSni dasturning rasmiy
sayti www.simplemachines.org orqali yuklab olishingiz mumkin.

PHP Nuke. Eng birinchi CMS lardan biri bo‘lib, afsuski, ancha vaqtlardan buyon yangilanmay kelmoqda. Biroq, uning kodi ko‘plab adashlari uchun asos bo‘lib xizmat qilmoqda (masalan, DotNetNuke, PostNuke). Bu bir-biriga nom jixatdan yaqin bo‘lgan tizimlar foydalanuvchi diqqatiga navbatma-navbat chiqib turadi. Funksionallik jixatidan tizim ancha mukammal bo‘lib, uning uchun ko‘plab qo‘sishcha modullar yaratilgan. Masalan, xabarlar lentasi, forum, maqolalar, so‘rovlardan tashkil qilish, statistika olib borish va boshqalar. Ammo, saytlarning tuzilishi jixatidan bir xillik kuzatiladi. SHu bilan birga, bu tizim yuklanishi qiyin bo‘ladi, ya’ni qo‘sishcha modullar saxifa yuklanishini sekinlashtiradi. Bu esa ko‘plab foydalanuvchilarga yokmasligi tayin. Yuklanish nafaqat saxifaga, shuningdek, serverga ham tushadi, bu esa xosting xizmatini taqdim etuvchilarga yoqmaydi.

Drupal. Turli xil o‘zgartirishlar orqali o‘zimizga moslab olish bo‘yicha eng mukammal CMS deb aytish mumkin. Shu yerda shuni aytish kerakki, “Tomchi” (bu tizim nomi ingliz tilidan aynan hhunday tarjima qilinadi) aslida, CMS emas, balki “CMS ni tuzuvchi tizim”. Uning yordamida siz tayyor bloklardan kerakli CMS ni yig‘asiz. Shuni aytish mumkinki, funksianallik va dizayn jixatidan o‘zingizni qanoatlantiruvchi CMS ni xosil qilasiz. Standart modullar to‘plamiga blog, forum, xabarlar lentasi, fayllarni yuklash, ovoz berish, qidiruv kabilalar kiradi. Bunda ko‘rinadigan kamchilik bitta: PHP va MySQL ni mukammal bilish lozim.

Slead CMS. Bugungi kunda ancha ommabop CMS bo‘lib, turli xil saytlarda ishlatiladi. Iffi xil varianti mavjud: pullik va tekin (OpenSlead). Ikkovi ham bitta yadroda qurilgan (bu degani xavfsizlik darajasi bir xil) va faqatgina funksionalligiga tasir qiluvchi modullar soni bilan farqlanadi. Bepul variantida biror mavzudagi nisbatan kichik portalni yaratishingiz mumkin. Ammo, bundan ortig‘ini emas. Dizayni ham unchalik katta qiyinchiliklarsiz o‘zgartirilishi mumkin.

13.3. CMS texnologiyasi – Joomlada sayt yaratish

Joomla veb saytlarni yaratish va boshqarish uchun yuqori darajadagi CMS (Content Management System) platformalardan biridir. Bu, WordPress va Drupal bilan birga, ommaviy va professional foydalanuvchilar uchun moslashuvga ega bo‘lgan tizim hisoblanadi. Joomla orqali veb saytlarni yaratish quyidagi bosqichlar orqali amalga oshiriladi:

1. Joomlani o‘rnatish: Birinchi qadam - Joomlani o‘rnatish. Joomla veb-saytlarni yaratish va boshqarish uchun rasmiy veb-sayt orqali Joomlani yuklab olishingiz mumkin. Keyin, uni o‘rnatilish va sozlash juda oson.

2. Dizayn tanlash: Joomlani o‘rnatgandan so‘ng, sizga mos dizayn tanlash imkoniyati beriladi. Buning uchun Joomla temalari yoki shablonlari mavjud bo‘lgan o‘z tuzilishga ega ko‘plab veb-resurslar mavjud. Sizning saytingizga mos temani tanlash va uni o‘rnatish orqali saytingizni shakllantirishingiz mumkin.

3. Tarkibni yaratish va tahrirlash: Joomla sayt tuzishda veb-sahifalarni yaratish va tahrirlash juda oson. Joomlani o‘rganish orqali, matn, rasmlar, videolar va boshqa kontentlar bilan to‘ldirilgan sahifalarni yaratishni osonlashtirish mumkin.

4. Modullar va pluginlar: Joomla modullari va pluginlari orqali saytingizga qo‘shimcha funksiyalarni qo’shishingiz mumkin. Masalan, kontakt formalar, kalit so’zlar, qo‘shimcha tarmoq havolalari, ijtimoiy tarmoq integratsiyasi va boshqalar kabi modullar va pluginlar mavjud.

5. Xususiyatlar va foydalanuvchilarga kirish: Joomla orqali, foydalanuvchilarga saytga kirish, ularga ro’yxatdan o’tish va ularning profilini boshqarish imkoniyatlarini taqdim etish mumkin. Shuningdek, Joomla-ni o‘rganish orqali, xususiyatlar qo‘sish va sayt tarkibini dinamik ravishda boshqarishni osonlashtirish mumkin.

CMS texnologiyalaridan biri bo‘lgan Joomlada sayt yaratish uchun dastlab virtual server Denwer ni o‘rnatamiz. Undan so‘ng xosil bo‘lgan W diskka kirib ichidagi home papkasiga sayt nomi ostida papka ochamiz, uning ichiga WWW papka ochamiz va uning ichiga Joomla saytini tashlaymiz va brauzerdan saytimizni chaqiramiz. Bu oynada o‘rnatish tilini tanlab oldinga tugmasi bosiladi. Keyingi xosil bo‘lgan oynamizda esa tekshirish ishlari boshlanadi. Tanishib bo‘lgach —далее tugmasi bosiladi.



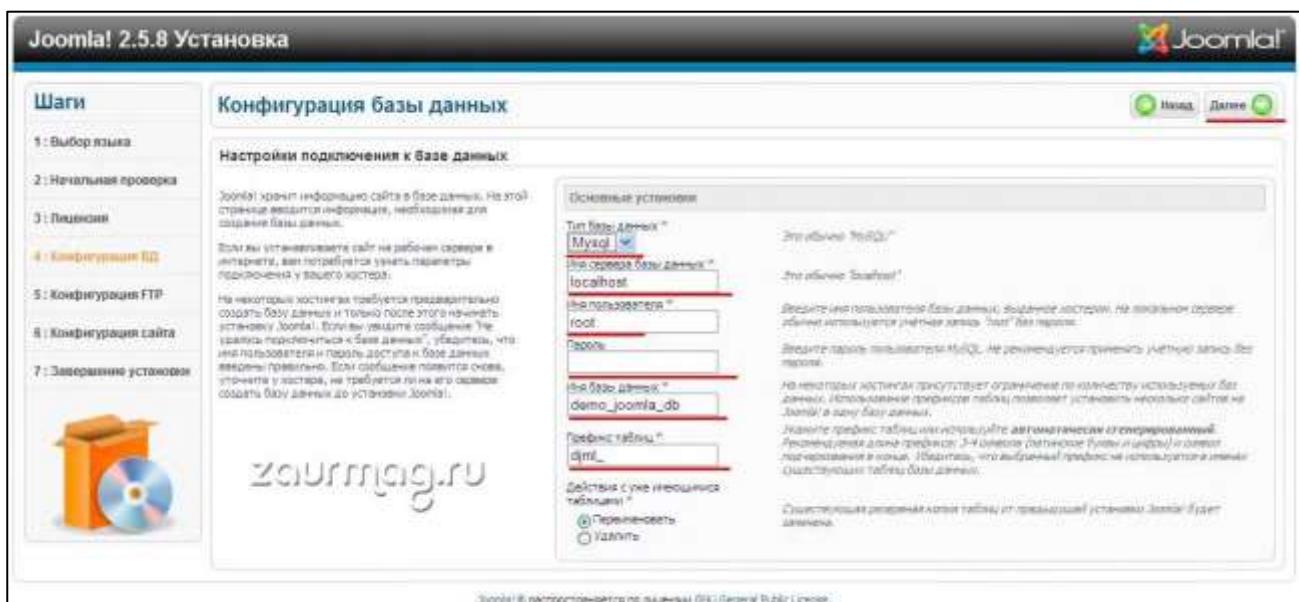
13.1-rasm. Joomlani o‘rnatishda dastlabki qadam

Ushbu oynada esa litzenziya haqida ma'lumot beriladi. So'ngra yana далее tugmasi bosiladi.



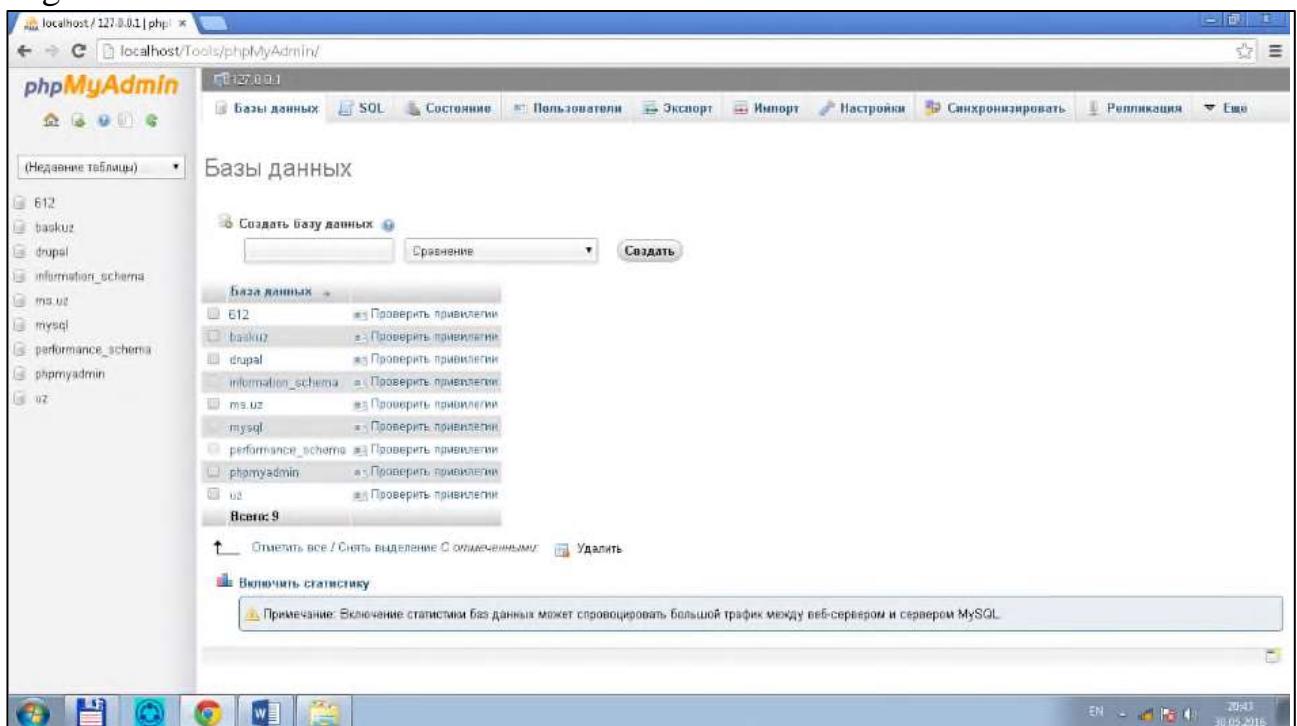
13.2-rasm. Litzenziya haqida ma'lumot beriladi

Quyidagi oynada esa ma'lumotlar bazasining tuzilishi kiritiladi. Bizni holatimizda u MySQL bazasi bo'lib xostning va foydalanuvchining nomi berilgan. Oxirgi qismda esa baza nomi kiritiladi. Shuning uchun yangi baza yaratamiz:



13.3-rasm. MySQL bazasi bo'lib xostning va foydalanuvchining nomi beriladi

Navbatdagi oynada MySQL bazasining oynasi berilgan bo‘lib yangi baza nomini yozamiz va yaratish tugmasi bosiladi. So‘ngra yuqoridagi oynada oldinga tugmasi bosiladi.



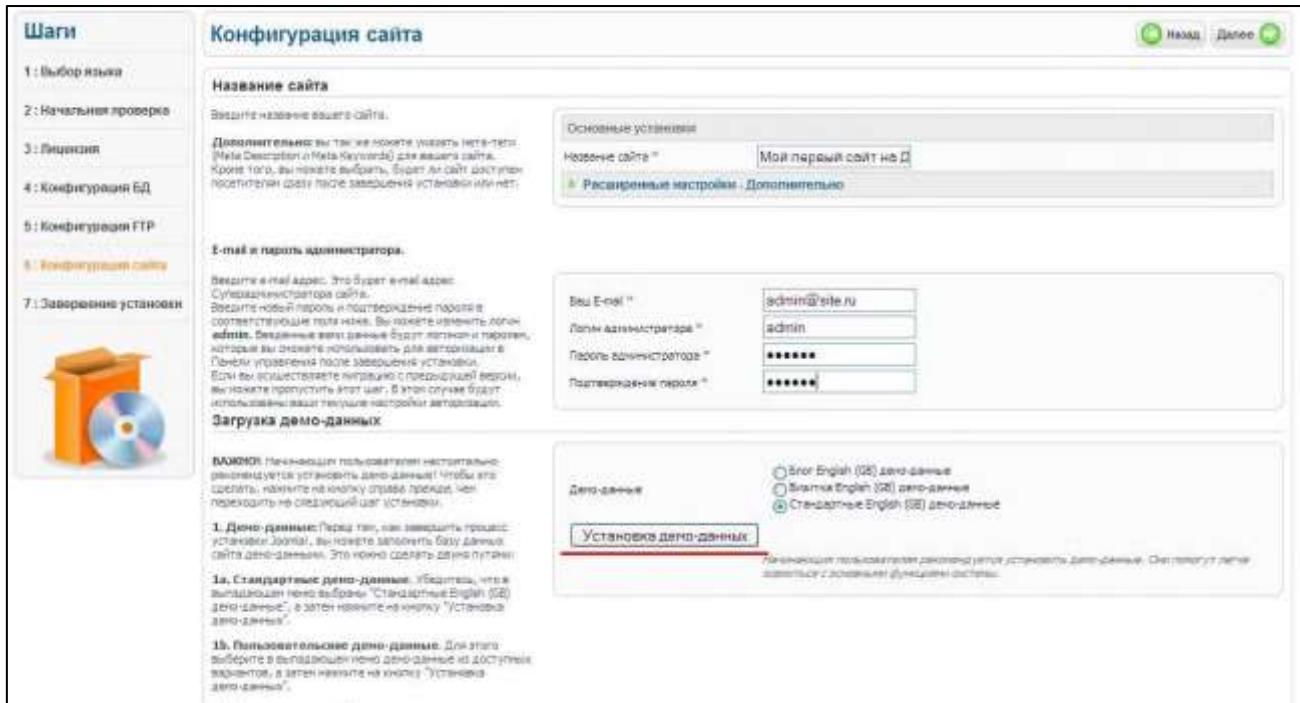
13.4-rasm. Yangi baza yaratish

Ushbu oyna esa o‘zgarishsiz qoldirilib —далее tugmasi bosiladi.



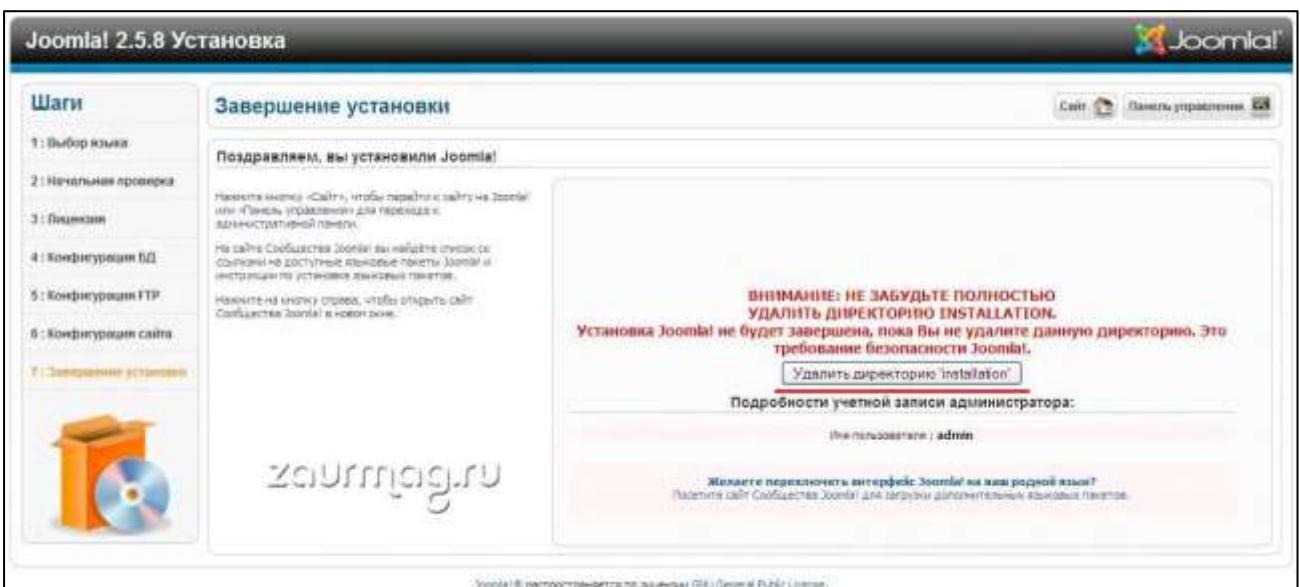
13.5-rasm. Konfiguratsiya FTP ko‘riladi

Quyidagi oynada esa saytning nomi, pochta manzili va parol takroran kiritiladi. So‘ngra Demo variantini o‘rnatish uchun —daleel tugmasi bosiladi.



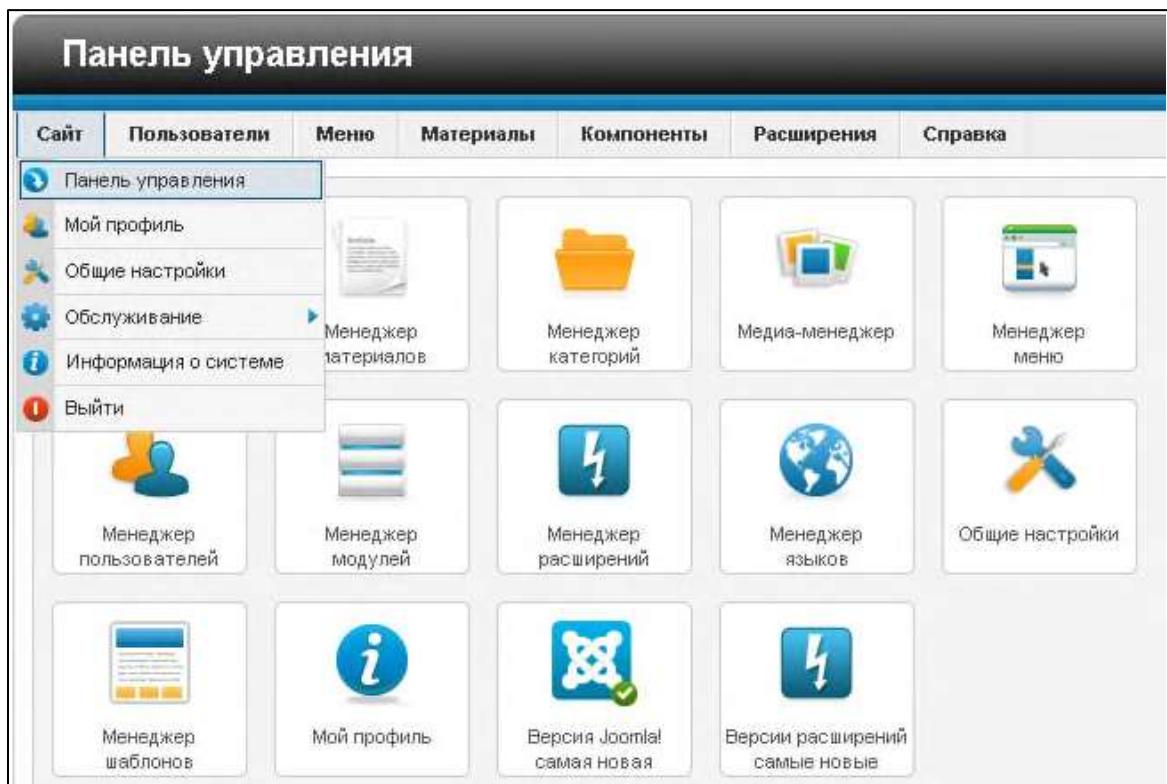
13.6-rasm. Pochta manzili, login, parol kiritiladi

So‘ngi oyna quyidagicha bo‘lib unda installition papkasini o‘chirib tashlash kerakligi ko‘rsatilgan. Aytilgan papkani o‘chirib bo‘lgach sajt tugmasi bosladi va jarayon yakullanadi.



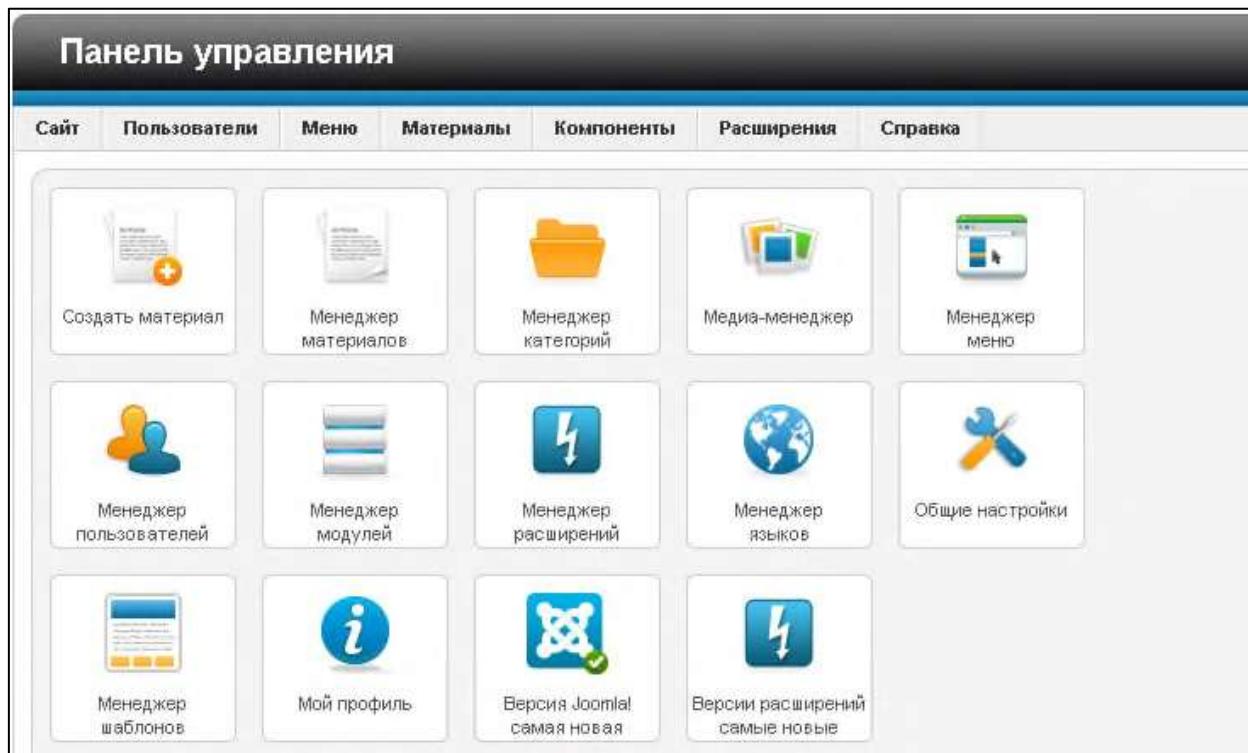
13.7-rasm. Yakunlanuvchi oyna

Saytning administrator oynasi quyidagicha bo‘lib u 7 bo‘limdan iborat.



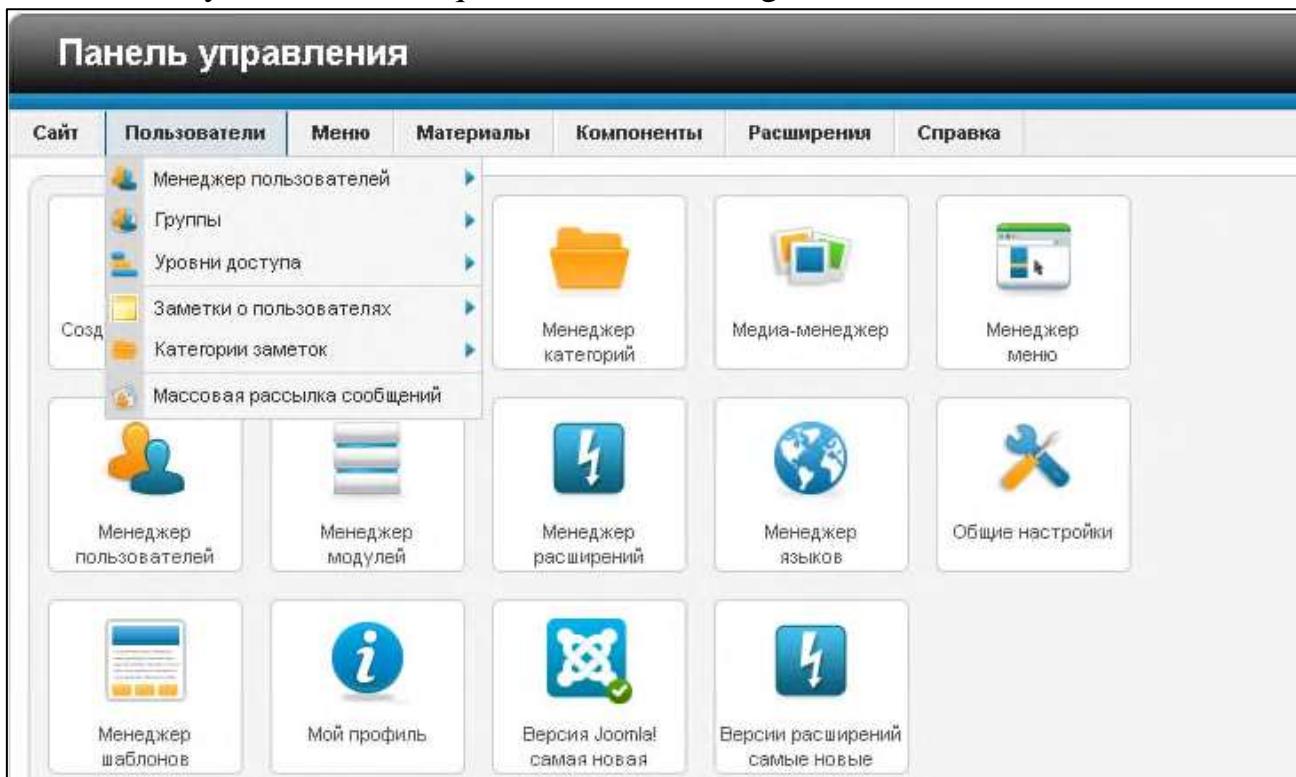
13.8-rasm. Boshqaruv paneli oynasi

Сайт bo‘limida boshqaruv paneli, foydalanuvchilarni boshqarish, global konfiguratsiya va paneldan chiqish bo‘linmalari joylashgan.



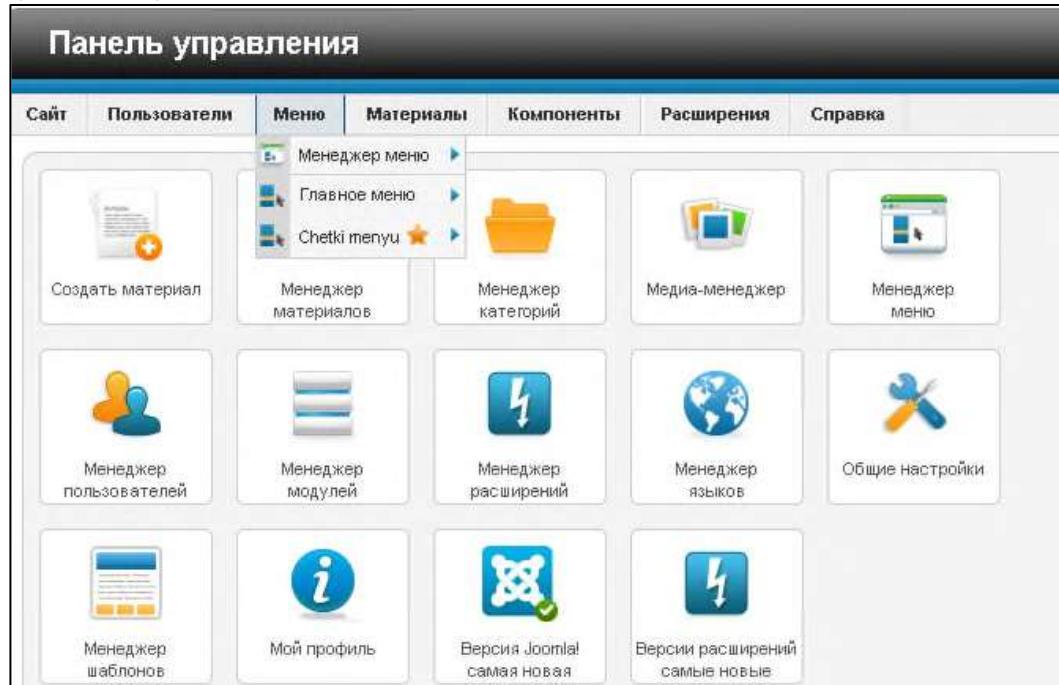
13.9-rasm. Sayt bo‘limi imkoniyatlari

Пользователи бу бо‘limda foydalanuvchi yaratish, guruhlar yaratish, eslatmalarini yaratish va boshqa narsalarni o‘z ichiga oladi.



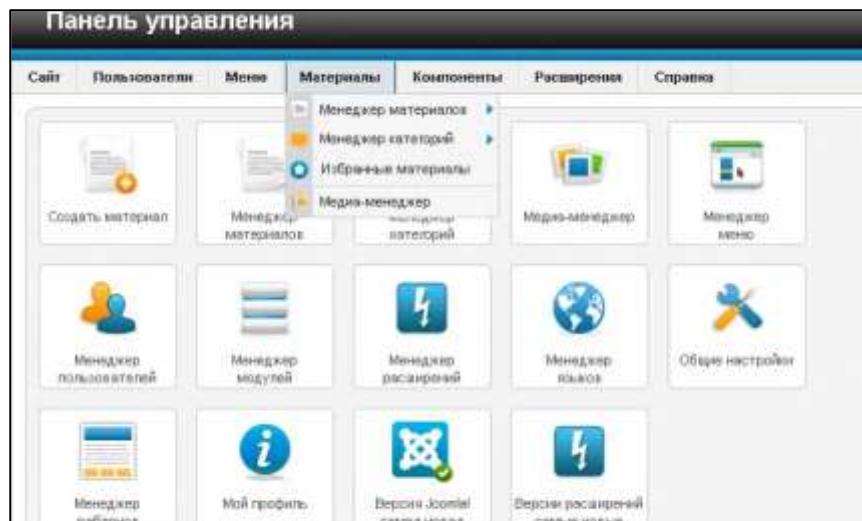
13.10-rasm. Пользователи бо‘limi imkoniyatlar oynasi

Меню бо‘limida saytning barcha qismidagi menyo‘larni boshqarish imkoniyali mavjud:



13.11-rasm. Меню бо‘limi imkoniyatlar oynasi

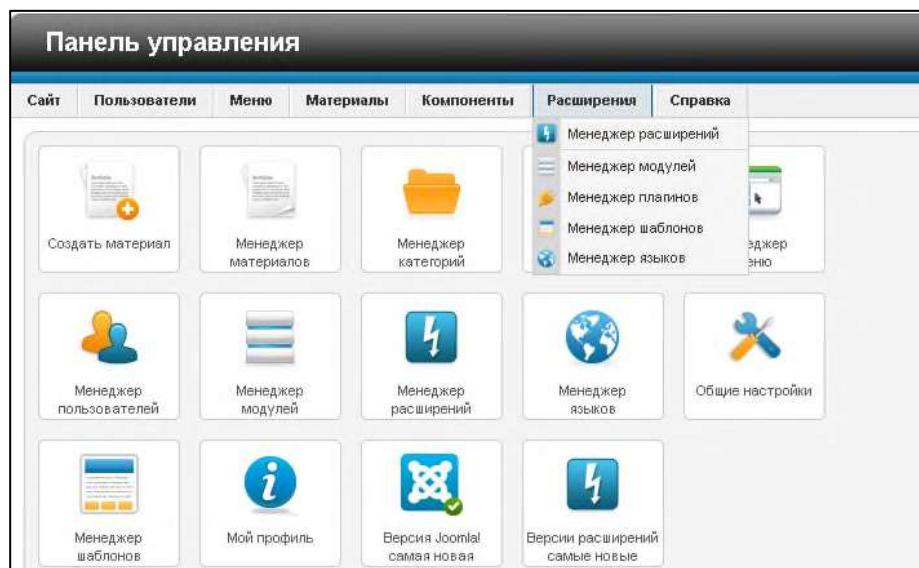
Материалы – bo‘limida biz qilayotgan web saytimizni kategoriysi va materiallarini kiritib chiqamiz. Biz yana asosiy sahifada ko‘rinib turishi uchun – избранны материалы qilib qo‘sghan ma’lumotlarni asosiy qilib qo‘yishimiz mumkin.



13.12-rasm. Материалы bo‘limi imkoniyatlari

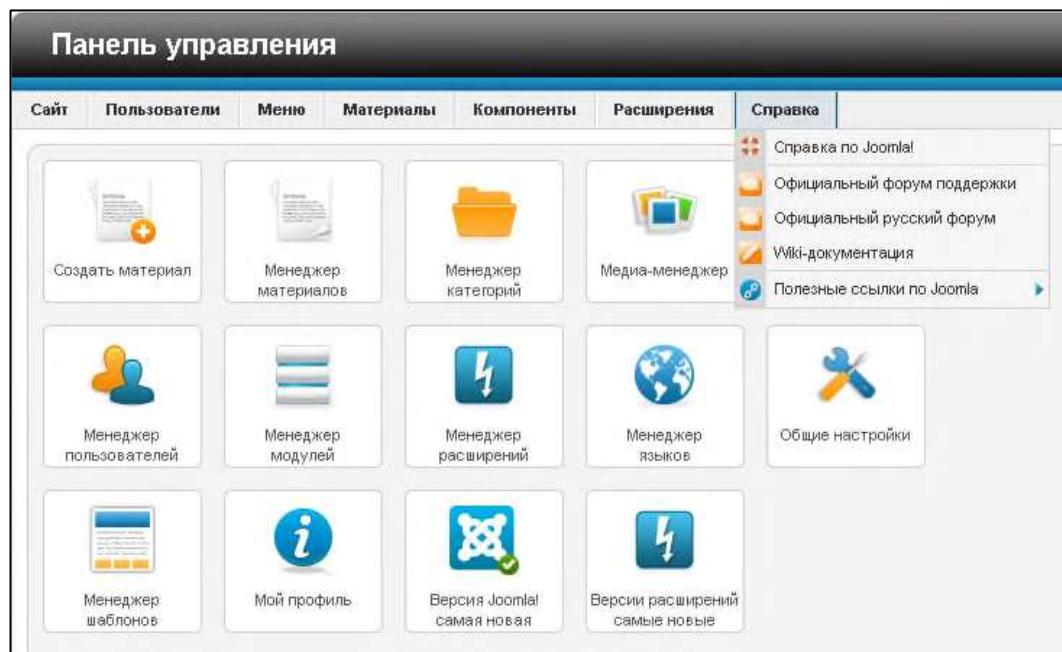
Компоненты bo‘limida biz qilayotgan saytimizda barcha komponentlarni boshqarish imkoniyatiga ega bo‘lamiz.

Расширения bu bo‘limda biz saytimiz uchun kerak bo‘ladigon barcha modul, pluginlar, shablonlarni yuklab olish imkoniyatiga ega bo‘lamiz va bu yerda biz yaratayotgan saytimizni tilini ham shu yerda o‘zgartirishimiz mumkin bo‘ladi.



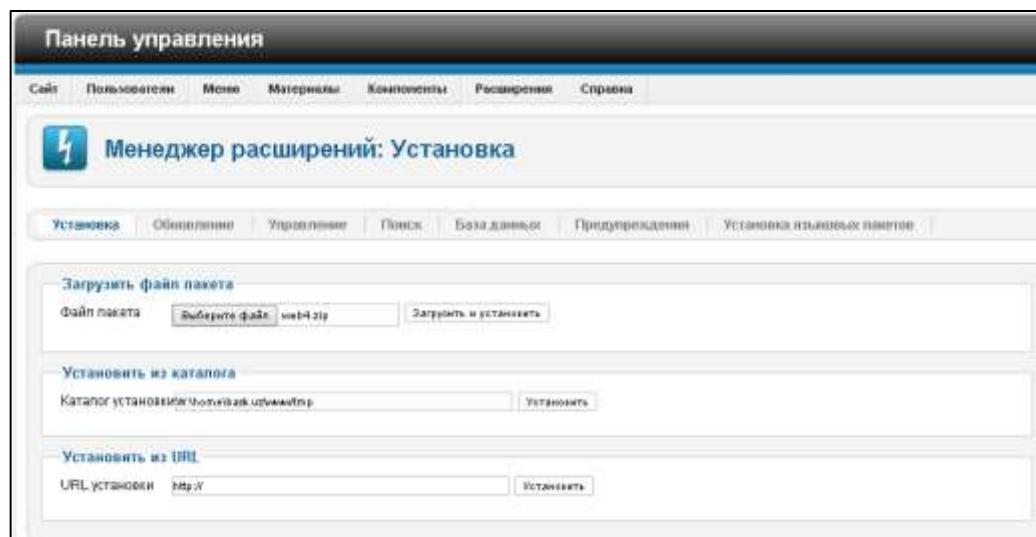
13.13-rasm. Расширения bo‘limi imkoniyatlar oynasi

Справка|| bo‘limida joomla haqida so‘rovlarga javoblar bo‘ladi ma’lumotlarga ega bo‘lamiz.



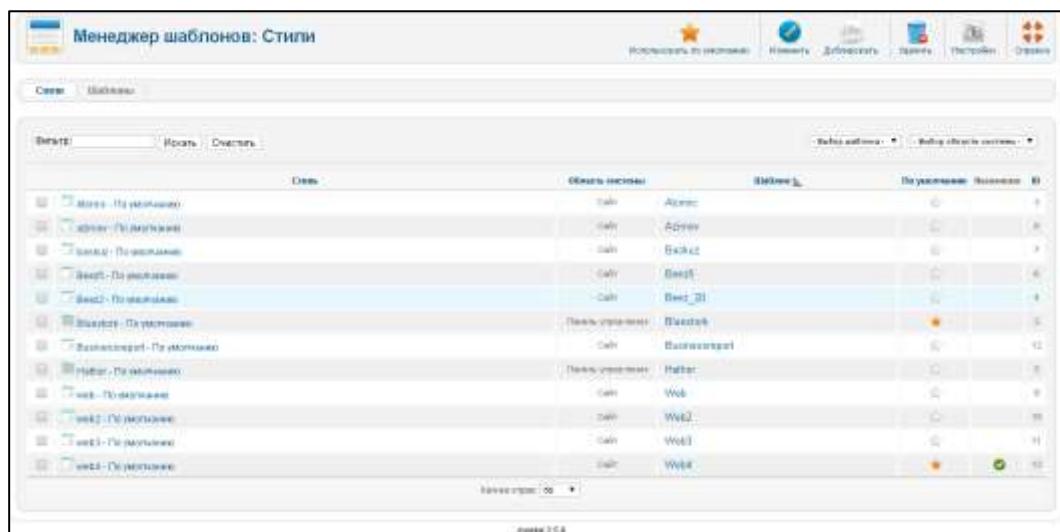
13.14-rasm. Справка bo‘limi imkoniyatlar oynasi

Boshqaruvi paneli bilan tanishib chiqdik endi biz saytni yaratish uchun CMS Joomladan foydalanib ish jarayonini olib boramiz. Joomlani ishga tushurganimizda u standart holatda boladi. Biz ta’lim sohasiga tegishli sayt qilganligimiz uchun o‘zimizga mos keladigon shablon o‘rnatamiz. Shablonni iloji boricha ochiq ranglardan foydalanganimiz maqsadga muofiq bo‘ladi. Yorqin ranglar esa saytda odamni ko‘zini ham charchatib qo‘yadi.



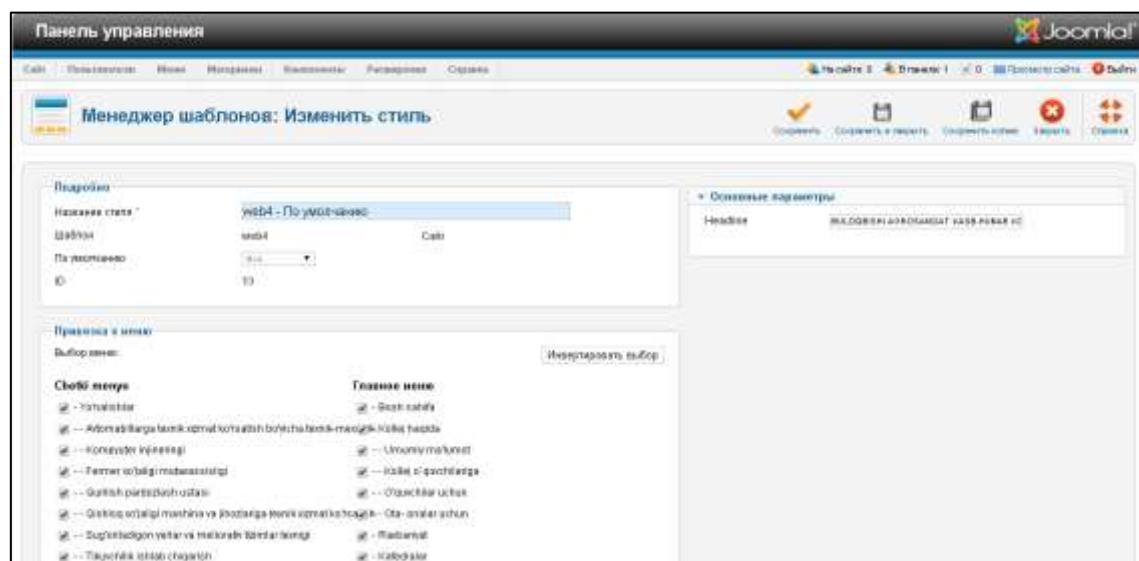
13.15-rasm. Shablon o‘rnatish

Yuklab olingan shablonni faol holatga o‘tkazish uchun —Панел управления bo‘limidan —Расширения bo‘limidan —Менеджер шаблонов bo‘limini tanlanadi va keyin esa yuklab olingan shablonni tanlanadi.



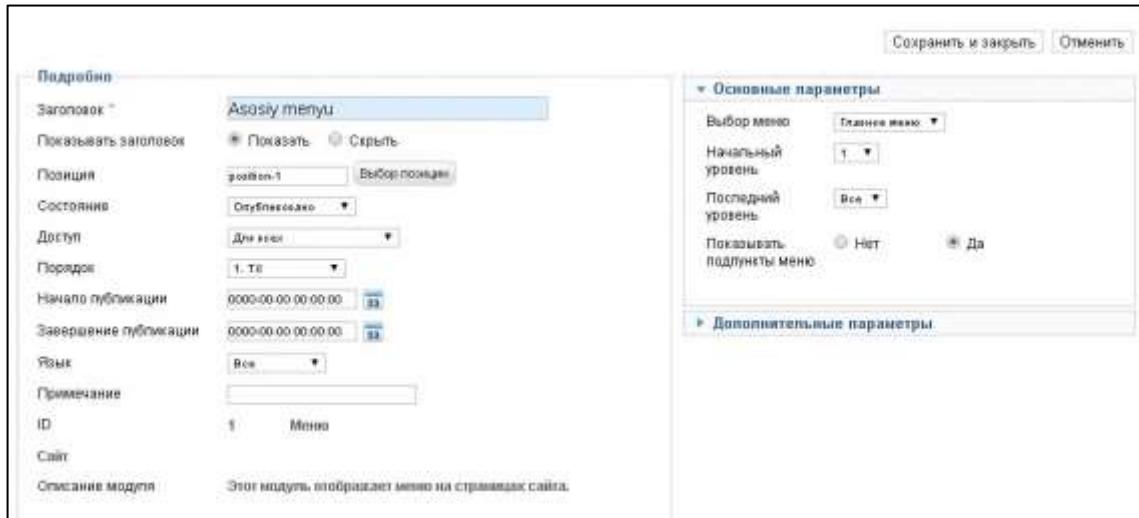
13.16-rasm. Shablonni faol holatga keltirish

Endi biz o‘rnatilgan shablonimizga yaratgan bo‘limlarimiz va o‘rnatgan modullarimiz ko‘rinishi uchun yuklangan shablonga kursorni olib borib chap tugmasini bir marta bosiladi, quyidagi —инвертировать выбор тугмаси bosiladi va —Сохранить и закрыть тугмаси bosilib chiqib ketiladi.



13.17-rasm. O‘rnatgan shablonimizda barcha bo‘limlar ko‘rinishi uchun

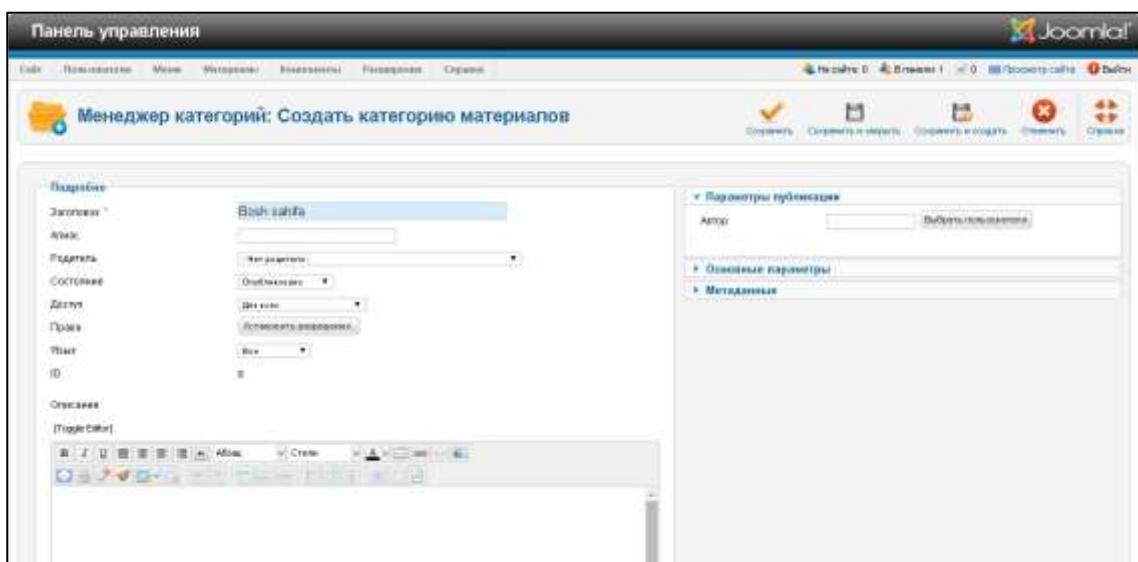
Asosiy menyu bo‘limini yaratamiz va qolgan bo‘limlar shu yaratgan menyumiz ichida joylashadi. Shablonimizni qaysi joyida turish keraligini ya’ni (position) ni ko‘rsatib qo‘yamiz. Buning uchun biz —Менеджер менюю bo‘limiga kirib yaratgan menyumizni joylashish o‘rnini ko‘rsatib qo‘yamiz.



13.18-rasm. Asosiy menyuni joylashish o‘rni

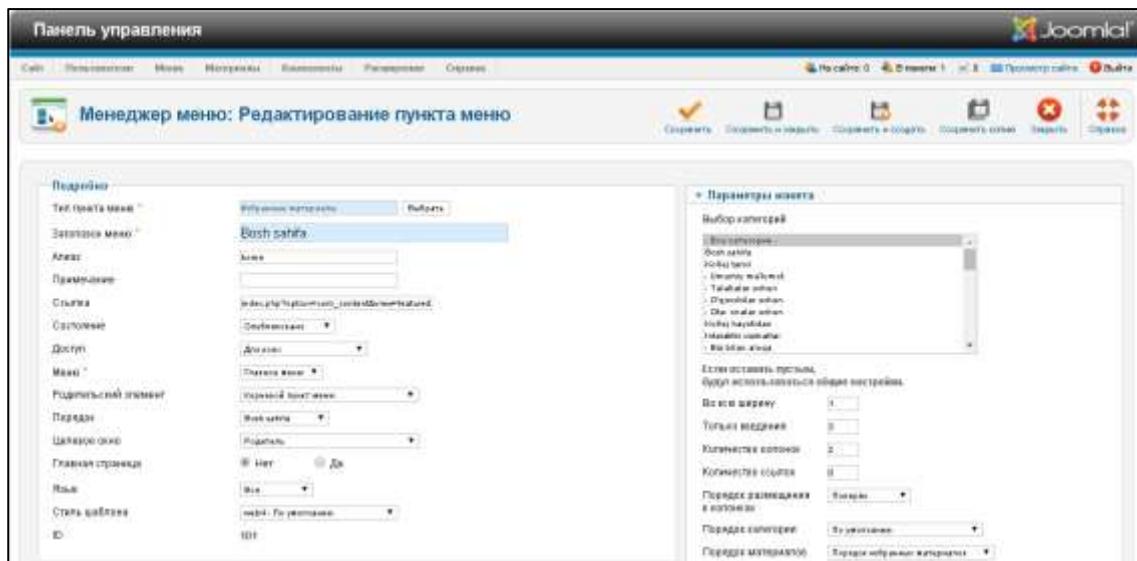
Saytimizga Bosh sahifa qismini yaratishni ko‘rib chiqamiz. Buning uchun biz bosh sahifani — категория sini yaratamiz.

Материалы bo‘limidagi — Менеджер категорий bo‘limidan — Создать категорий tanlanadi va Сахранить и закрыть tugmasi bosiladi.



13.19-rasm Bosh sahifasini katigoriyasini yaratish

Biz bosh sahifa menyusini yaratish uchun —Меню bo‘limidan —> Менеджер меню—> Создать меню bo‘limidan bosh sahifa bo‘limini yaratamiz va — Сохранить и закрыть tugmasi bosiladi.



13.20-rasm. Bosh sahifa menyusini yaratish

Bosh menyuga ma’lumot kiritilmaydi unga qo‘yilgan materiallar asosiy qilib chiqiladi va bosh sahifamizda ko‘rinib turadi.

NAZORAT SAVOLLARI

- 13.1. CMS larning qanday turlari bor?
- 13.2. CMS texologiyalarining afzalliklari?
- 13.3. CMS texnologiyalarining kamchiliklari?
- 13.4. CMS texnologiyasi – Joomlani yuklash bosqichlari?
- 13.5. CMS texnnologiyasining maqsadi?
- 13.6. CMS texnologiyasining vazifalari?
- 13.7. WordPress va Joomla. Qaysi cms texnologiyasini afzal ko‘rasiz?

TEST

<p>1. Web brauzerlar ...</p> <p>a. Internet Explorer, Power Point, Google Chrome</p> <p>b. Google Chrome, Mozilla Firefox, Opera</p> <p>c. Google Chrome, Modella Firefox. Word</p> <p>d. Download master, Modella Firefox. Opera</p> <p>2. Gipermurojaat bu ...</p> <p>a. butunjahon o‘rgimchak to‘ridan hujjatning URL deb nomlanuvchi manzilini qidirishda ishlataladi</p> <p>b. sahifada boshqa Web sahifalar bilan bog‘lovchi chekli gipermurojaat</p> <p>c. Dasturlar asosida yaratilgan sahifani tasvirlovchi dastur</p> <p>d. To‘g’ri javob yo‘q</p> <p>3. TEG lar deb nimaga aytildi</p> <p>a. HTML ning konstruksiyasi TEG lar deyiladi</p> <p>b. HTML tilida boshqa kompyuter tillaridagi kabi izoh berish TEG lar deyiladi</p> <p>c. sarlavha qismi va asosiy qism TEG lar deyiladi</p> <p>d. fayl oddiy matnli fayl bo‘lib uni maxsus brauzer TEG lar deyiladi</p> <p>4. HTML tilida boshqa kompyuter tillaridagi kabi izoh berish imkoniyati mavjud. Izoh quyidagi qaysi belgilar orsiga joylashtiriladi?</p> <p>a. #- - # va # - #</p> <p>b. < - - va < - - ></p> <p>c. //</p> <p>d. < - - > va < - - ></p> <p>5. Sarlavha qismi quyidagi ochiluvchi va yopiluvchi teglari orasida joylashadi.</p> <p>a. <head></head></p> <p>b. <Body></body></p> <p>c. < title ></ title ></p> <p>d. < Link ></ Link ></p> <p>6. Web sahifaning asosiy qismi ... teglari orasida joylashadi</p> <p>a. < Link > va </ Link > teglari orasida joylashadi.</p> <p>b. < title > va </ title > teglari orasida joylashadi.</p> <p>c. < body > va </ body > teglari orasida joylashadi.</p> <p>d. < get > </ get > teglari orasida joylashadi.</p> <p>7. Alink - foydalanuvchi tomonidan ...</p> <p>a. tanlangan gipermurojaat rangi.</p> <p>b. web sahifa matni yozilgan tilni aniqlash</p>	<p>23. Elektron pochta uzatishning oddiy qaydnomasi–</p> <p>a. SMTP</p> <p>b. TCP/IP</p> <p>c. DNS</p> <p>d. HTTPS</p> <p>24. IP manzillar va kompyuterlar domen nomlarini aniqlovchi server bu –</p> <p>a. Proxy server</p> <p>b. DNS server</p> <p>c. Gateway</p> <p>d. URL server</p> <p>25. Internetga murojaat qilishning eng oddiy va qulay usuli bo‘lib, u manzilni ifodalaydi.</p> <p>a. URL</p> <p>b. DNS</p> <p>c. Proxy</p> <p>d. FTP</p> <p>26. Dastur ta’minati va protokollari asosida tashkil etilgan, hamda ma‘lumotlar bazasi va elektron hujjatlar bilan kollektiv ravishda ishlash imkonini beruvchi korxona yoki konsern miqyosidagi yagona informatsion muhitni tashkil etuvchi kompyuter tarmog‘i</p> <p>a. Intranet</p> <p>b. Korporativ tarmoq</p> <p>c. Server</p> <p>d. Internet</p> <p>27. Internet uchun katta bo‘lmagan klient va server ilovalarni yaratishga mo‘ljallangan nisbatan oddiy obyektga yo‘naltirilgan til</p> <p>a. HTML</p> <p>b. Javascript</p> <p>c. PHP</p> <p>d. XML</p> <p>28. SCRIPT konteyneri qaysi 2 ta asosiy funksiyani bajaradi?</p> <p>a. HTML-dokument ichiga kodni joylashtirish, brauzer tomonidan HTML-belgilarni shartli hosil qilish.</p> <p>b. server tomonidan dokumentlarning sahifalarini hosil qilish, brauzer tomonidan HTML-belgilarni shartli hosil qilish.</p> <p>c. HTML-dokument ichiga kodni joylashtirish, HTML-dokumentni tahlil</p>
--	---

<p>c. foydalanuvchi tomonidan oldin murojat qilingan gipermurojaat rangi.</p> <p>d. tasvirlanayotgan matn rangi.</p>	<p>qilish vaqtida interpretator boshqaruvini o'z qo'liga oladi</p>
<p>8. Web sahifadagi avtor haqida ma'lumot yozmoqchi bo'lsak uni quyidagi ko'rinishda yozish mumkin:</p>	<p>d. boshqaruvni interpretatoriga beradi, server tomonidan dokumentlarning sahifalarini hosil qiladi</p>
<p>a. <Body name = "Auther" content= "Bu meniki!!!"></p> <p>b. <Lang name = "Auther" content= "Bu meniki!!!"></p> <p>c. <meta name = "Lang" content= "Bu meniki!!!"></p> <p>d. <meta name = "Auther" content= "Bu meniki!!!"></p>	<p>29. PHPda ssenariy boshlanish va tugash teglarini ko'rsating</p>
<p>9. Gipermatnlar tili (HTML)...</p>	<p>a. <?e ?></p>
<p>a. web-brauzerlar ekraniga ma'lumotlarni standart kodlar orqali chop etish imkoniyatini beradi.</p>	<p>b. <p></p></p>
<p>b. teglari asosiy matnni o'rta ga olib turibdi</p>	<p>c. #PH #ph</p>
<p>c. so'ralayotgan xujyatning aniq adresini ko'rsatish imkoniyatini berish</p>	<p>d. <?echo ?></p>
<p>d. xujyatni yaratishga mo'ljallangan maxsus programmalar</p>	<p>30. Windows muhitida web server tashkil qiluvchi dasturlar</p>
<p>10. DNS server bu...</p>	<p>a. Java, OpenServer, HTML</p>
<p>a. IP manzillar va kompyuterlar domen nomlarini aniqlovchi server.</p>	<p>b. XAMP, OpenServer</p>
<p>b. Internet da ba'zi bir ma'lumotlarga ko'pchilik murojaat qilgani uchun bu ma'lumotlarga oid serverga ularish</p>	<p>c. PHP, Download</p>
<p>c. Internet aloqa kanallari orqali uzatilgan ma'lumotlar oqimi hajmi.</p>	<p>d. OpenGL, Unserved</p>
<p>d. ma'lumotlarni uzatishning turli qaydnoma (protokol)larini internet foydalanadigan elektron pochtaning oddiy qaydnomasi</p>	<p>31. HTML Attributelar keltirilgan qatorni belgilang</p>
<p>11. Quyidagilardan qaysi shakl elementi sifatida ishlatalidi?</p>	<p>a. Bg, H1, tr</p>
<p>a. <hr></p>	<p>b. Size, color, width, align</p>
<p>b. <form></p>	<p>c. Body, head, aling</p>
<p>c. <button></p>	<p>d. Width, ul, p</p>
<p>d. <fieldset></p>	<p>32. Qidirish mashinalari Web sahifalar haqidagi ma'lumotlarni qaysi tegdan oladi?</p>
<p>12. Quyidagi qaysi biri rang uslubini aniqlash uchun to'g'ri?</p>	<p>a. Meta ma'lumotlar (<meta> tegida keymards va description)</p>
<p>a. H1 {color: FF-00-88}</p>	<p>b. Web sahifadagi matnli gipermurojat</p>
<p>b. H1 {color: red}</p>	<p>c. <body> tegi bir qancha qo'shimcha parametrlar</p>
<p>c. H1 {font-color: red}</p>	<p>d. Web sahifaning asosiy qismi <body></p>
<p>d. H1 {color: rgb(#D46A11)}</p>	<p>33. O'zarobog'liq tarzda kodlangan matn, rasm va boshqa shakldagi axborotlardan iborat o'zgarmas veb sahifalar majmui-</p>
<p>13. CSS Imkoniyatlari bu</p>	<p>a. Dinamik sayt</p>
<p>a. Turli muhitlar, qurilmalar uchun ma'lumotlarni taqdim etishni boshqarish</p>	<p>b. Statik sayt</p>
<p>b. Web-sahifaning HTML kodini o'zgartirish</p>	<p>c. Axborot sayt</p>
<p>c. taqdimot tarkibni vizual boshqarish</p>	<p>d. Kungilochar sayt</p>
<p>d. taqdimot tarkibini o'zgartirish</p>	<p>34. Foydalanuvchi so'rovini bajarish jarayonida axborotlarni bir-biri bilan bog'lanmagan xolda o'zgarib turuvchi web sahifalardan iborat sayt-</p>
<p>14. Keyingi qatorga o'tishni qaysi teg belgilaydi?</p>	<p>a. Dinamik web-sayt</p>
<p>a.
</p>	<p>b. Statik web-sayt</p>
	<p>c. Axborot web-sayt</p>
	<p>d. Kungilochar web-sayt</p>
	<p>35. <table border="#"></p>
	<p>a. Jadval ramkasi qalinligini berish.</p>

<p>b. <a> c. <div> d. </p> <p>15. Brauzer keshidan foydalanishning asosiy kamchiligi nimada?</p> <p>a. Ma'lumotlar yangilanmasligi mumkin. b. Qayta ishlashga sabab bo'ldi. c. Yuklash vaqtini oshiradi. d. Shifrlashni ishlatib bo'lmaydi. Ba'zi Internet-provayderlar buni qo'llab-quvvatlamaydi.</p> <p>16. Web-sahifalarni joylashtirish usullari qanday?</p> <p>a. blokli b. jadvalli c. ierarxik d. aloqador</p> <p>17. Web-sahifa qaysi noyob xislati bilan ajratidi?</p> <p>a. rasmlar b. tarkibi c. css fayli d. URL manzili</p> <p>18. HTMLda ismingizni yozish. Alisher Qodirov shunday yozgan: <p> Alisher Qodirov </p> Ushbu matnni brauzer qanday ko'rsatadi?</p> <p>a. ikki qatorda; b. bitta qatorda ikkita bo'shliq bilan; c. bitta bo'shliq bilan bitta qatorda; d. umuman ko'rsatmaydi.</p> <p>19. Ikkala to'g'ri variantni tekshiring. Qanday teglar shrift rangini o'zgartirishi mumkin?</p> <p>a. <HTML> ... </HTML> b. <BODY> ... </BODY> c. ... d. <P> ... </P></p> <p>20. Jadval satrini belgilaydigan teg nomi</p> <p>a. table; b. tr; c. td; d. cell;</p> <p>21. Faylni serverga yuborish uchun forma tugmasi ...</p> <p>a. <input type="submit"> b. <input type="button"> c. <input type="image"> d. <input type="file"></p> <p>22. CSS fayli HTML –...</p> <p>a. hujjatdagi matn, jadval va boshqa elementlarni formatlash uchun ishlatiladigan shablon b. sahifaga kaskad jadvallar usuli c. murakkab usullarini qo'llash d. faylning hajmi juda kattalashib boradi</p>	<p>b. Jadval yacheyskalari orasidagi masofani berish. c. jadval qiymatlari va ramkasi orasidagi masofani berish. d. jadval enini piksellarda yoki hujjat eniga nisbatan foizlarda berish.</p> <p>36. Fayl jo'natish bayonnomasi</p> <p>a. HTTP b. FTP c. PFT d. HTTPS</p> <p>37. Formada kiritish maydonini hosil qilish teglari</p> <p>a. <input></input> b. <form></form> c. <frame></frame> d. <forms></forms></p> <p>38. Matn kiritish uchun ko'p satrli maydon yaratish</p> <p>a. <input type="radio" name="NAME" value="x"> b. <textarea name="NAME" cols=40 rows=8></textarea> c. <select name="NAME"></select> d. <text name="NAME" cols=40 rows=8></text></p> <p>39. Internet aloqa kanallari orqali uzatilgan ma'lumotlar oqimi hajmi nima deb ataladi?</p> <p>a. Proxy b. Trafik c. Getway d. Usertraf</p> <p>40. URIga ega va umumiy interfeysi XML-da belgilangan dasturiy ta'minot tizimi bu -</p> <p>a. Web server b. Web servis c. Pochta tizimi d. Ximoya tizimi</p> <p>41. HTML qaysi teglar gipermatnli havolani yaratish uchun ishlatiladi?</p> <p>a. <A> b. c. <I> </I> d. <S> </S></p>
---	--

QISQARTMALAR VA ATAMALAR

Atamalar va qisqartmalar	Ingliz tilidagi sharhi	O‘zbek tilidagi sharhi
TCP/IP	Transmisson control protocol/internet protocol	uzatishni boshqarish protokoli / internet protokoli
WWW	World Wide Web	Umumjahon o‘rgamchak to‘ri
URL	Uniform Resource Locator	Yagona manba identifikatori
CMS	Content Management System	Kontentlarni boshqarish tizimi
CSS	Cascading Style Sheets	Kaskadli jadval stillari
DB	Database	Ma’lumotlar bazasi
HTML	HyperText Markup Language	Gipermatnli belgili til
HTTP	HyperText Transfer Protocol	Gipermantlarni uzatuvchi protokol
PHP	Hypertext Preprocessor	Gipermatn prosessori
W3C	World Wide Web Consortium	Butun dunyo o‘rgimchak to‘ri konsorsiumi
XHTML	Extensible HyperText Markup Language	Gipermatnli kengaytirilgan belgili til
XML	Extensible Markup Language	Kengaytirilgan belgili til
SQL	Structured Query Language	Strukturalangan so‘rovlar tili
ASCII	American Standart Code for Information Interchange	Axborot almashishning Amerika standarta
SGML	Standard Generalized Markup Language	Standart umumlashtirilgan belgilash tili
FTP	<i>File Transfer Protocol</i>	Fayl uzatish protokoli
WML	English Wireless Markup Language	Simsiz qurilmalar uchun belgilash tili

“WEB TEKNOLOGIYALAR” FANI BO‘YICHA GLOSSARIY

So‘z	ingl	рус	Ma’nosi
CSS-dizayn	CSS-design	CSS дизайн	Hujjatni joylashtirish usuli, tuzilish va taqdimotning tushunchasini aks ettiradi
Kaskadli jadval stillari	Cascading Style Sheets	Каскадные таблицы стилей	HTML, XHTML va XML-hujjatlarning dizayni uchun mo‘ljallangan, stil xususiyatlarining standart tili
Ma’lumotlar bazasi	Database	База данных	Bitta yoki bir nechta mahsus fayl shakllarida mayjud bo‘lgan, tuzilmaviy va tashkiliy ma’lumotlar to‘plami
Gipermatnli belgili til	HyperText Markup Language	Язык гипертекстовой разметки	SGML ilovasi hisoblanuvchi va gipermatnli hujjatlar tuzilmasini yozish uchun mo‘ljallangan, standart belgili tillardan biri
Web-sahifa	Web page	Веб-страница	Standart belgili tillardan biri yordamida tashkil etilgan, gipermatnli hujjat
Web-server	webserver	Веб сервер	Web-sayt fuksionalligini ta’minlovchi, mahsus dasturiy ta’milot. Web-serverning asosiy vazifasi – kontentning dinamik generasiyasi uchun skriptlarni ishga tushirish va brauzer orqali
Web-sayt	Website	Интернет сайт	Web-saytning qat’iy belgilangan atamasi mavjud emas. Web-saytni Internet tarmog‘ida o‘zining unikal manziliga ega bo‘lgan joy sifatida tavsiflash mumkin
Gipermurojaat	Hypertext	Гиперссылка	Web-sahifa gipermurojaat deb nomlanuvchi, matnli satrlarning boshqa sahifalariga yo‘naltiriladi
MySQL	MySQL	MySQL	Ochiq kodli ma’lumotlar bazasining dasturiy ta’miloti
ARPANet	ARPANet	ARPANet	buzilgan aloqa bo‘g‘inlarni avtomatik ravishda aylanib o‘tishga va tarmoqdagi kompyuterlarning ma’lumot

			almashishiga imkon yaratuvchi kommunikatsiyalar paketidir
Internet	Internet	Internet	dunyo bo‘ylab joylashgan va yagona tarmoqqa birlashtirilgan minglab kompyuter tarmoqlarining majmuidir
web brauzerlar	веб-браузер	web browser	Internetning WWW xizmatidan foydalanish uchun maxsus dasturlar
Elektron pochta	Электронная почта	Email	kompyuter tarmoqlari bo‘ylab elektron xatlar bilan almashish (qabul qilish va yuborish) imkonini beruvchi texnologiyadir
Jadval	Table	Таблица	Ma’lumotlar bazasi obyekti. Satr va ustunlar ichiga tashkil etilgan tegishli ma’lumotlarni saqlaydi
Asosiy kalit	Primary key	Первичный ключ	Birlamchi kalit hisoblanadi
Atribut	attribute	атрибут	Xususiyat, sifat yoki miqdor belgisi. U makondagi obyektni ta’riflovchi (biroq uning qaerda joylashganligini ko‘rsatish bilan bog‘liq bo‘lmagan) va uning noyob soni ya’ni aniqlovchisi bilan bog‘liqlikda tasavvur qilinadi. Atribut ko‘rsatkichlari majmuasi odatda ma’lumotlar bazalarini relyasion boshqarish tizimi vositalari yordamida jadvallar shaklida taqdim etiladi.
Kalit	key	ключ	Ramzlar majmui. U obyektlarni o‘xhash obyektlar to‘plamidan ajratib olish, ularni qidirish yoki maxfiylashtirish uchun ishlataladi.
TEG	TEG	TEG	HTML ning konstruksiyasi
So‘rov	query	запрос	Ma’lumotlarni uzatishda bosh kompyuter tomonidan bo‘ysunuvchi kompyuterdan uning holatini (ish rejimini) aniqlash va oydinlashtirish maqsadida so‘rov qilish jarayoni

FOYDALANILGAN ADABIYOTLAR

Asosiy adabiyotlar

1. A.X.Boymurodov, Sh.Sh.Allamova. / Web texologiyalari / O‘quv qo‘llama – Toshket “City of book”, 2023. - 224 bet
2. Jon Duckett. Javascript & jquery: interactive front-end web development© by John Wiley & Sons, Inc., Indianapolis, Indiana. All Rights Reserved. Published Under License with the original publisher John Wiley & Sons, Inc.
3. X.N.Zayniddinov va boshqalar. Web ilovalarni yaratish // Darslik. - T.:Aloqachi, 2020.–348 b
4. Жадаев А. Г. PHP для начинающих. — СПб.: Питер, 2014. — 288 с.
5. Кириченко А. В., Хрусталев А.А. HTML5+CSS3. ОСНОВЫ СОВРЕМЕННОГО WEB-ДИЗАЙНА — СПб.: «Наука и Техника», 2018 г. – 352 с.
6. Jon Duckett. HTML & CSS. Design and build Websites. Published by John Wiley & Sons, Inc.2011
7. Sh.Sh.Allamova. / Oliy ta’lim muassasalarida o‘quv jarayoni samaradorligini aralash ta’lim asosida takomillashtirish metodikasi: Ped.fan.fal.dok. avtoreferati – Chirchiq 2024. -54 bet.

Qo‘srimcha adabiyotlar

1. Mirziyoyev Sh.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. 2017.
2. Mirziyoyev Sh.M. Qonun ustuvorligi va inson manfaatlarini ta’minalash – yurt taraqqièti va xalq farovonligining garovi. 2017.
3. Mirziyoyev Sh.M. Erkin va farovon, demokratik O‘zbekiston davlatini birgalikda barpo etamiz. 2017.
4. Usmonov J.T., Xujaqulov T.A. Ma’lumotlar bazasini boshqarish tizimi// o‘quv qo‘llanma. - T. : Aloqachi, 2018. – 96 b.
5. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. – СПб.: Питер, 2011. –496 с.
6. Макфарланд Д. Новая большая книга CSS – СПб.: Питер, 2016. –720 с.
7. Гольцман В. MySQL 5.0. Библиотека программиста. Питер; Санкт-Петербург; 2010.

Internet saytlari

1. <http://www.tutorialspoint.com/>
4. <http://www.phpframework.com>
7. www.w3schools.com/
8. www.tutorialspoint.com/php/

MUNDARIJA

KIRISH.....	3
I-BOB. WEB-TEXNOLOGIYA HAQIDA TUSHUNCHA. BRAUZER DASTURLAR. WEB-SAHIFANING TUZILISHI VA YARATISH IMKONIYATLARI, ASOSIY TUSHUNCHALARI. HTML TILIGA KIRISH, UNING ASOSIY ELEMENTLARI. HTML DA MATNLAR BILAN ISHLOVCHI TEGLAR.....	5
1.1. Web-sahifa, Web-sayt, Web-server tushunchalari.....	5
1.2. Belgilash tillarining rivojlanish bosqichlari.....	8
1.3. “Klient-server” texnologiyasi.....	12
1.4. HTML tiliga kirish, uning asosiy elementlari.....	13
1.5. Web sahifaning tuzilishi va yaratish imkoniyatlari.....	14
1.6. Matnlar bilan ishlash.....	20
NAZORAT SAVOLLARI.....	25
II-BOB. HTML DA RO‘YXATLAR, JADVALLAR VA XAVOLALAR BILAN ISHLASH. HTMLDA MULTIMEDIA OBEKTLARI.....	26
2.1. Htmlda ro‘yxatlar bilan ishlash.....	13
2.2. Htmlda jadvallar bilan ishlash	32
2.3. Htmlda multimedia obyektlari va gipermatnli hujjat	37
NAZORAT SAVOLLARI.....	43
III-BOB. HTML TILIDA GIPERMATN VA FORMALAR JOYLASHTIRISH TEGLARI. FREYMLAR HOSIL QILISH VA ULAR BILAN ISHLASH.....	44
3.1. Htmlda gipermatn bilan ishlash.....	44
3.2. Shakllar va ularning vazifalari.....	46
3.3. Freymlar hosil qilish va ular bilan ishlash.....	67
NAZORAT SAVOLLARI.....	71

IV-BOB. KASKADLI STILLAR JADVALI(CSS), CSS STILLARNI QO'LLASH, HTMLGA ULANISH USULLARI. SELEKTOR TURLARI.....	72
4.1. Css kirish. Kaskadli uslublar jadvallarini HTML hujjatlariga ulash.....	37
4.2. CSS3 da mavjud sintaksis qoidalari va selectorlar.....	76
4.3. Css da rang berish usullari.....	81
NAZORAT SAVOLLARI.....	84
V-BOB. CSS DA MATNLAR, RASM VA XAVOLALARGA ISHLOV BERISH.....	85
5.1. Cssda matnlarga ishlov berish.....	85
5.2. Rasmlarga ishlov berish.....	92
5.3. Xavolalarga ishlov berish.....	95
NAZORAT SAVOLLARI.....	97
VI-BOB. CSSDA RO'YXATLAR, JADVALLAR VA SHAKLLARGA ISHLOV BERISH.....	98
6.1. Cssda ro'yxatlarga ishlov berish.....	98
6.2. Jadvallarga chegaralar va fon qo'shish	100
6.3. Forma elementlarining ko'rinishini o'zgartirish	103
6.4. Bloklar bilan ishlash.....	107
6.5. Elementlarning joylashishini nazorat qilish.....	110
NAZORAT SAVOLLARI.....	125
VII-BOB. JAVASCRIPTGA KIRISH. MALUMOT TURLARI. ARIFMETIK OPERATORLAR.....	126
7.1. Script yaratish.....	126
7.2. Javascriptda izohlar.....	129
7.3. JavaScriptda o'zgaruvchi.....	130
7.4. Arifmetik amallar.....	133

NAZORAT SAVOLLARI.....	134
VIII-BOB. JAVASCRIPTDA SHART OPERATORLARI.....	135
8.1. JavaScriptda IF shart operatori.....	135
8.2. JavaScriptda if else operatori.....	135
8.3. Switch operatori	137
8.4. JavaScriptda shartni tekshirishda ishlataladigan operatorlar.....	138
NAZORAT SAVOLLARI.....	139
IX-BOB. JAVASCRIPTDA SIKL OPERATORLARI BILAN ISHLASH.....	140
9.1. JavaScriptda for takrorlash operatori.....	140
9.2. WHILE takrorlash operatori.....	143
9.3. DO WHILE takrorlash operatori.....	143
NAZORAT SAVOLLARI.....	143
X-BOB. JAVASCRIPTDA FUNKSIYALAR VA MASSIVLAR BILAN ISHLASH.....	144
10.1. JavaScriptda funksiyalar bilan ishslash JavaScriptda funksiyalar bilan ishslash.....	144
10.2. JavaScriptda massivlar bilan ishslash.....	149
NAZORAT SAVOLLARI.....	152
XI-BOB. JAVASCRIPTDA SATRLAR BILAN ISHLOVCHI FUNKSIYALAR BILAN ISHLASH.....	153
11.1. JavaScriptda obektlar.....	153
11.2. JavaScript obekt xususiyatlari.....	158
11.3. Ko‘p obektlarni yaratish. Konstruktor belgisi.....	162
11.4. O‘rnatilgan obektlar.....	165
NAZORAT SAVOLLARI.....	169
XII-BOB. DOM ELEMENTLARI. QUERY SELECTORLAR.....	170
12.1. Hujjat obekti modeli.....	170
12.2. HTML elementlarini o‘zgartirish va ularga namunalar.....	181

NAZORAT SAVOLLARI.....	185
XIII-BOB. WEB SAYTLARNI YARATISHDA CMS	
TEXNOLOGIYASIDAN FOYDALANISH.....	186
13.1 CMS. Asosiy tushunchalar. CMS ning maqsad va vazifalar....	186
13.2. CMS turlari.....	189
13.3. CMS texnologiyasi - Joomlada sayt yaratish.....	191
NAZORAT SAVOLLARI.....	202
TEST.....	203
QISQARTMALAR VA ATAMALAR.....	206
“WEB TEXNOLOGIYALAR” FANI BO‘YICHA GLOSSARIY.....	207
FOYDALANILGAN ADABIYOTLAR.....	209
MUNDARIJA.....	210