

**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA MAXSUS
TA'LIM VAZIRLIGI**

BUXORO DAVLAT UNIVERSITETI

Fizika –matematika fakulteti

“Axborot texnologiyalari ” kafedrası

“KT: Informatika va axborot texnologiyalari” ta’lim yo’nalishi

“Kompyuter grafikasi va web dizayn” fanidan

KURS ISHI

Mavzu: HTML da forma va uning elementlarining hodisalarida JavaScript dasturlaridan foydalanish.

Bajardi: 1-1PMIK-13 guruh talabasi Rahmonqulov Ma’mur Qobul o’g’li.

Kurs ishi himoya qilingan sana

“ ____ ” _____ 2016 y.

Ball _____

Komissiya a’zolari

Buxoro- 2016

Mundarija:

Kirish.....	3
I. HTML haqida umumiy tushunchalar.....	5
1.1. HTML haqida umumiy ma'lumot.....	5
1.2. HTML da ob'yektlar ierarxiyasi.....	9
1.3. HTML da formalar.....	12
II. JavaScript dasturlash tili.....	14
2.1. Javascript haqida umumiy tushunchalar.....	14
2.2. JavaScriptda funksiyalar yozish.....	17
2.3. Forma hodisalariga funksiyalarni joriy qilish.....	20
Xulosa.....	24
Foydalanilgan adabiyotlar ro'yxati.....	25

KIRISH

O'zbekiston Respublikasi mustaqillikka erishgan yillaridan e'tiboran mamlakatimiz hayotida haqiqatan ham ulkan tarixiy ahamiyatga ega bo'lgan eng muhim ijtimoiy dasturlar – kadrlar tayyorlash milliy dasturi va maktab ta'limini rivojlantirish davlat umummilliy dasturining ijrosi amalda nihoyasiga yetkazildi.

O'zbekistonda o'qitish texnologiyalari zamonaviylashtirish, jadallashtirish rivojlangan iqtisodiyotli mamlakatlarga qaraganda yanada dolzarb ahamiyatga ega. Chunki hozirgi kunda o'zbek milliy ta'lim tizimining salohiyatli tizimli rivojlanishi yanada yuqori pog'onasiga ko'tarildi.

Prezidentimiz I.A.Karimov 2001-yil Oliy Majlisining 5-sessiyasida so'zlagan nutqida axborot texnologiyalarni va kompyuterlarni jamiyat hayotiga, kishilarning turmush tarziga, o'rta maktab, o'rta maxsus va oliy ta'lim muassasalariga jadallik bilan kirish g'oyasi ilgari surildi. Xususan, biz kabi yoshlarni zamon talablariga javob beruvchi mutaxassislar qilib tayorlashda alohida e'tibor qaratilmoqda.

Respublikamizda olib borilayotgan islohotlarning tarkibida yuqori malakali mutaxassislarning roli benihoyat kattadir. Prezidentimiz ta'kidlaganlaridek, "Ertangi kun yangicha fikrlay oladigan zamonaviy bilimga ega bo'lgan yuksak malakali mutaxassislarni talab etadi".

Respublikamiz mutaxassislar tayyorlashni davr talablari darajasida tashkil etishda yetarlicha ishlar olib borilayotganligiga qaramasdan, hozirgacha bir qator muammolar o'z yechimini topgani yo'q. Xususan, jahon andozalariga mos va raqobatbardosh, bilimli, malakali, kuchli mutaxassislarni tayyorlashga imkon beruvchi pedagogik shart – sharoitlarni yaratish, mazkur jarayonni metodik jihatdan to'g'ri tashkil etish, mutaxassislik fanlaridan kasbiy ta'lim va malakalarni shakllantiruvchi ilmiy – uslubiy qo'llanmalar ishlab chiqish kabi masalalarni shular qatoriga kiritish mumkin.

Hozirgi zamon mutaxisslardan faoliyat doirasi qanday bo'lishidan qat'iy nazar, informatika bo'yicha keng ko'lamdagi bilimlarga zamonaviy hisoblash

texnikasi va information aloqa va kommunikatsiya tizimlari, orgtexnika vositalari va ulardan foydalanish borasida yetarli malakaga ega bo'lishi, hamda yangi information texnika va texnologiya asoslarini uning ertangi kuni, rivoji to'g'risidagi bilimlarni o'zida mujassamlashtirgan bo'lishi shart. Zamonaviy hisoblash texnikalari va information texnologiyalarining kun sayin rivojlanib, jamiyatning esa, tobora informatsiyalashib borishi sababli, uzluksiz ta'lim tizimining o'rta va yuqori bosqichlariga informatika, ishlab chiqarish va boshqarish jarayonining kompyuterlashtirish bo'yicha bir qator o'quv fanlari kiritilgan.

XX asr o'rtalariga kelib, tezkor mashina mexanizmlaridan foydalanila boshlandi, murakkab texnika va texnologiyalardan ko'p funksiyalilari o'ylab topildi. Ko'pgina masalalarni hal qilish jarayonida axborot hajmi behisob bir majmuaga aylandi, hamda bu axborotlarni yig'ish va uzatish vositalarini yaratish, ularni vaqtida qayta ishlab, boshqarish uchun zarur bo'lgan choralarni belgilab chiqish kerak bo'ladi.

Mavzuning dolzarbligi. Axborot kommunikatsion texnologiyalar(AKT) talabalarning kelajakdagi faoliyatiga tayyorlashga ko'mak beradi. Hozirgi zamonda mehnat faoliyati o'quvchilar qoniqish bilan ishlayotgan kompyuterlar, texnologiyalar, dasturlar va qurilmalar yordamida boshqariladi. AKT o'rganish va o'qitishning yangi imkoniyatlarini ochib beradi. Inson multimediya ko'rgazma vositalari yordamida, o'quv ma'lumotlarini qabul qilganda sezgi a'zolari ishtirok etadi va u oson tushunadi. Bugun kunda web texnologiyalari juda tez sur'atlarda rivojlanmoqda. Zamontalabidan kelib chiqqan holda web texnologiyalarini, jumladan JavaScript dasturlash tilini chuqurroq o'rganishimiz zarur hisoblanadi.

Kurs ishining tarkibi. Kurs ishi kirish qismi, 2 bob, xulosa va foydalanilgan adabiyotlar ro'yxatidan tashkil topgan bo'lib jami 25 betdan iborat.

I. HTML haqida umumiy tushunchalar.

1.1. HTML haqida umumiy ma'lumot.

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

«ko'shtirnoq» ichida beriladi. - Parametr qiymatida ba'zan yozuv registri muxim. SHuni esda tutish lozimki, hamma teglar o'zining individual parametriga ega bo'lishiga karamay, shunday bir qator parametrlar mavjudki, ularni <BODY> bo'limining barcha teglarida ishlatish mumkin. Bu parametrlar CLASS, ID, LANG, LANGUAGE, STYLE va TITLElardir. HTML-hujjatini yozishni boshlashda ishlatiladigan birinchi teg bu <HTML> tegidir. U har doim hujjat yozuvining boshida bo'lishi lozim. YAkunlovchi teg esa </HTML> shakliga ega bo'lishi kerak. Bu teglar, ular orasida joylashgan yozuvning hammasi butun bir HTML-hujjatini anglatishi bildiradi. Aslida esa hujjat oddiy matnli ASCII-faylidir. Bu teglarsiz brauzer hujjati formatini aniqlab, tarjima qila olmaydi. Ko'pincha bu teg parametriga ega emas. HTML 4.0 versiyasiga qadar VERSION parametri mavjud edi. HTML 4.0da esa VERSION o'rniga <!DOCTYPE> parametri paydo bo'ldi.

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

Hujjatning HEAD bo'limi

HEAD bo'limi sarlavha hisoblanadi va u majburiy teg emas, biroq mukammal tuzilgan sarlavha juda ham foydali bo'lishi mumkin. Sarlavha qismining maqsadi hujjatni tarjima kilayotgan dastur uchun mos axborotni etkazib berishdan iborat. Hujjat nomini ko'rsatuvchi <TITLE> tegidan tashqari bu bo'limning qolgan barcha teglari ekranda aks ettirilmaydi. Odatda <HEAD> tegi darhol <HTML> tegidan keyin keladi. <TITLE> tegi sarlavhaning tegidir, va hujjatga nom berish uchun hizmat kiladi. Hujjat nomi <TITLE> va </TITLE> teglar orasidagi matn qatoridan iborat. Bu nom barauzer oynasining sarlavhasida paydo bo'ladi (bunda sarlavha nomi 60 belgidan ko'p bo'lmasligi lozim). O'zgartirilmagan holda bu matn hujjatga «zakladka» (bookmark) berilganda ishlatiladi. Hujjat nomi uning tarkibini qisqacha ta'riflashi lozim. Bunda umumiy ma'noga ega bo'lgan nomlar (masalan,

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

Hujjatning BODY bo'limi

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

<**BODY**> tegi parametrlari:

ALINK - faol murojaat (ssылka)ning rangini belgilaydi.

BACKGROUND - fondagi tasvir sifatida foydalaniluvchi tasvirning

URL-manzilini belgilaydi.

BOTTOMMARGIN - hujjatning quyi chegaralarini piksellarda belgilaydi.

BGCOLOR - hujjat fonining ranglarini belgilaydi.

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

LEFTMARGIN - chap chegaralarni piksellarda belgilaydi.

LINK - xali ko'rib chiqilmagan ssылkaning rangin belgilaydi.

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

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

TEXT - matn rangini aniqlaydi.

TOPMARGIN - yuqori chegarasini piksellarda o'rnatadi.

VLINK - ishlatilgan murojaat rangini belgilaydi.

BOTTOMMARGIN, LEFTMARGIN, RIGHTMARGIN va TOPMARGIN pametrlari matn chegarasi va darcha chetlari orasidagi masofani piksellarda belgilaydi. (Faqat HTML 4.0 versiyasidan boshlab IE brauzerlari bu parametrlarni

taniy oladi)BGPROPERTIES parametri faqatgina bitta FIXED qiymatiga ega. HTML dagi ranglar o'n oltilik sanoq tizimida (RGB), yoki ranglar nomi yordamida berilishi mumkin. Ranglar bazasi 3 ta rangga - qizil (R) , yashil (G) va ko'k (B) ranglarga asoslangan bo'lib, u RGB deb belgilanadi. Har-bir rang uchun 00 dan FF gacha bo'lgan o'n oltilik sanoq tizimidagi qiymat beriladi, bu esa 0 dan 255 gacha bo'lgan diapazonga to'g'ri keladi. So'ngra bu qiymatlar bir songa birlashtiriladi va ularning oldiga “#” belgisi kuyiladi. Masalan, #800080 siyohrangni bildiradi.

Misollar:

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

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

<BODY LINK = “#ff0000”> yoki <BODY LINK = RED>

<BODY LINK = “#00FFFF” ALINK = “#800080”> yoki <BODY VLINK = Aqua ALINK = PURPLE>

Hamma barauzlar o'n oltilik sanoq tizimidagi standart ranlarni taniydi. Bular quyidagilardir:

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

Misol:

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

Agar BGCOLOR parametri rangni nomi yoki uning tarkibiy kislarnnn i n oltilik sanoq tizimidagi kodda keltirish vazifasi yordamida fon ranggini chiqarsh

uchun ishlatilsa, BACKGROUND tasvir yordamida sahifaga fon berishda foydalaniladi. Tasvir sifatida GIF yoki JPG formatidagi grafik fayllar ishlatiladi. HTML-hujjat fonidagi tasvir doimo butun sahifani to'ldirib turadi. Agar tasvir o'lchami darcha o'lchamidan kichik bo'lsa, u mozayka tamoyiliga asosan kupaytiriladi. Odatda fon tasviri sifatida tarmoq orqali yuklash uchun uncha ko'p vaqt ketmaydigan kichik tasvir tanlab olinadi, yoki fon sifatida shaffof relef logotipi tasviridan foydalaniladi.

Misol:

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

Sahifa yaratilishida doimo fon rangini berish tavsiya qilinadi. Agar fon tasviri ham

berilayotgan bo'lsa, fon va tasvir ranglari bir-biriga yaqin bo'lgani ma'qul.

Misol:

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

Misol:

<HTML>

<HEAD>

<TITLE> - sahifa fonini berish misoli

</TITLE>

</HEAD>

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

1.2. HTML da ob'yektlar ierarxiyasi

Web saxifaning barcha elementlari JavaScript tilidagi ierarxik tuzilishi buyicha kuruladi. Xar bir element ob'ekt kurinishida nomayon buladi. Va bundan xar bir ob'ekt anik tarkibga va usulga ega buladi. Uz navbatida JavaScript tili Web saxifa ob'ektlarini oson boshkarish imkoniyatini beradi. Va bunda siz ob'ektlar ierarxiasini tushunishingiz juda muxim. Bularning kandy ruy berishini navbatdagi misol orkali tez tushunishingiz mumkin.

Oddiy HTML saxifani kuramiz:

Demak biz 2 ta tasvir, 1 ta murojat va kandydir matn chikarish uchun 2 maydondan iborat forma va 1 ta tugmachaga egamiz. JavaScript nuktaiy nazaridan browser darchasi – allakanday Window ob'ekti bulib xisobllanadi. Bu ob'ekt uz navbatida bir necha rasmiylashtirish elementlariga ega. Darcha ichkarisiga biz html dokumentini urnatishimiz urnatishimiz mumkin (yoki boshka tildagi fayl, xozircha html fayli bilan chegaralanib turamiz). Bunday saxifa document ob'ektidan sira xam fark kilmaydi. Bu shuni anglatadiki document ob'ekti JavaScript tilida xozirgi paytda yuklangan html dokumentni namoyon etadi. Document ob'ekti JavaScript tilida juda muxim ob'ektlardan biri xisoblanadi va siz undan kup marta foydalanasiz. Masalan Document ob'ektining tarkibiga , Web saxifaning foning rangi kiradi. SHunday kilib biz uchun shunisi muximki html ob'ektlvridan barchasi document ob'ektining tarkibiga kiradi. Html ob'ektlariga misol, murojaat yoki tuldirilgan forma bula oladi.

Albatta biz bu ierarxiya yordamida turli ob'ektlar xakida ma'lumotga ega bulishimiz va uni boshkarish imkoniga ega bulishimiz kerak. Buning uchun biz JavaScript tilida turli ob'ektlarga ruxsat kandy tashkil etilganligini bilishimiz shart. Kurinib turibdiki ierarxik tuzilishga kura xar bir ob'ekt uz nomiga ega. Mos ravishda agar siz html saxifadagi 1-rasmga murojat kilmokchi bulsangiz ob'ektlar ierarxiasidagi uz yunalishingizni tanlab olishingiz kerak va eng yukori chukkidan boshlash kerak. Bu tuzilishdagi 1-ob'ekt document deb ataladi. Saxifadagi 1-rasm images [0] ob'ekti kabi tasvirlangan. Bu shuni anglatadiki JavaScript da document.images[0] deb yozib bu ob'ektga ruxsat olgan bulamiz. Masalan agarda siz formaning 1-elementiga kandy matn kiritilganini bilmokchi bulsangiz datslab

kanday kilib ob'ektga ruxsat olishni aniklab olishingiz kerak va yana ob'ektlar ierarxiasining eng yukorisidan boshlaymiz. Sungra `elements[0]` ob'ektga yul kursatamiz va ketma-ketlik bilan barcha ob'ektlarni nomini yozamiz va natijada shu narsa aniklanadiki 1-maydonga ruxsatni shularni yozish orkali olish mumkin:

`Document.forms[0].Elements[0]`. Endi foydalanuvchi tomonidan kiritilgan matnni kanday kilib bilish mumkin. Ob'ektning kaysi turi yoki xossalaridan biri bu informatsiyaga ruxsat berishi mumkinligini aniklash uchun JavaScript xakidagi ma'lumotnomalarga murjat kilish kerak. U erda siz kiritilgan matn maydoniga mos keluvchi element `value` xossasi borligi va u kiritilgan matnga mos keladi. Demak endi biz kiritilgan kiymatni ukish uchun barcha kerakli narsalarga egamiz. Buning uchun JavaScript satrida kuyidagini yozish mumkin.

Name = `document.froms[0].elements[0].value`

Olingan satr **name** uzgaruvchisiga yuklanadi. Endi biz bu uzgaruvchidan xoxlagan kerakli paytda foydalanishimiz mumkin. Masalan, biz `alert ("Hi"+name)` buyrug'i yozilgan darcha yaratishimiz mumkin. Natijada, agar foydalanuvchi bu maydonga 'Stefan' cuzini kiritgan bulsa `alert ("Hi"+name)` buyrug'iga kura yangi darchada 'Hi Stefan' salomi paydo buladi. Agar siz katta saxifalar bilan ish kursangiz turli ob'ektlarga nomlar buyicha adreslash protsedurasida juda chalkashlik bulishi mumkin. Masalan `document.forms[3].elements[17]` `document.forms[2].elements[18]` ob'ektlariga murojaat kilish kerak bulsin? Bunday muommadan kutilish uchun turli ob'ektlarga uzingiz nom berishingiz mumkin. Bularni kanday kilishni misol orkali kurishingiz mumkin:

`<form name = "myForm"`

`Name`

`< input type = "text" name = "kirit " value = " "> <br>`

bu yozuv shuni bildiradiki, `forms[0]` ob'ekti endi 2-chi ism `myForm` ga ega buldi. Xuddi shunday `elements[0]` urniga kirit ni yozishingiz mumkin (`<input>` tegida oxirgi atributi `name` da kursatilgan). SHu tarika

`name = document.forms[0].elements[0].value` urniga

`name = document.myForm.kirit.value` ni yozishingiz mumkin. Bu sezilarli darajada JavaScript da dasturlashni soddalashtiradi, ayniksa katta Web-saxifalar kup mikdorda ob'ektlarga ega bulsa. (Nomlarni yozishda registrnlarni, kata-kichik xarflar xolatini e'tiborga olish kerak, ya'ni myForm urniga myform yozish mumkin emas) JavaScript da ob'ektlarning kupchilik xususiyatlarida ruxsat fakat ukish uchun emas. YAna siz ularga yangi kiymatlar yozish imkoniyatiga xam egasiz.

1.3. HTML da formalar.

Formalar internetda keng ishlatiladi. Formaga kiritilgpn ma'lumotlar tez – tez kayta serverga yuboriladi yoki biror bir adresga eleketron pochta orkali junatiladi. Muammo shunday, foydalanuvchi tomonidan kiritilgan ma'lumot tugriligiga ishonish. Uni internetga yuborishdan oldin Java S tili yordamida oson tekshirish mumkin. Oldin bu formani tekshirish kanday bulishini kursatmokchi edik, keyin esa internetga ma'lumot junatishning kanday imkoniyatlari borligini kursatamiz. Dastlab biz oddiy skript yarataylik. Aytaylik html-saxifa 2 ta matn kirituvchi elementga ega bulsin. Ulardan biriga foydalanuvchi uz ismini, 2 – siga elektron pochta manzilini kiritsin. Siz u erga biror bir ma'lumotni kiritib tugmani bosib kuring.

Agar siz u erga xech narsa kiritmagan bulsangiz xatolik yuz berish xakida ma'lumot olasiz. 1 – elementdagi istalgan ma'lumot tugrilik jixatdan tekshiriladi. Albatta agar foydalanuvchi boshka nomni kiritgan bulsa u bularni kafolatlamaydi. Brauzer xatolarni sonlarga xam karshilik kursatmaydi. Masalan, siz '17' deb kiritsangiz u xolda siz 'Salom 17' degan salomni olasiz. Demak, bu tekshirish ideal bula olmaydi.

Formaning 2 chi elementi bir muncha murakkabirokdir. Unga oddiy satr kirgizib, masalan ismingizni. Siz unga @ ni yozmaguncha bu element ish bajarmaydi. @ belgini mavjudligi bilan foydalanuvchi elektron pochta manzilini tugri kiritgan bula oladi. Garchi ma'nosiz bulsada birgina @ belginig uzi xam bu

shartga tugri keladi.internetdagi xar bir elektron pochta manzilida @ belgi katnashadi. SHunday ekan bu belgiga tekshirish bu erda juda urinli. Skript bu 2 ta element bilan kanday ishlandi va unda tekshirish kay kurinishda buladi. Bu kuyidagicha olib boriladi:

Dastlab </body> bulimidagi kodni kurib chikamiz. Biz bu erda fakat 2 ta tugmachani matn kiritish uchun yaratamiz. Bu tugmachalar test1(...) va test2 (...) funktsiyalarini kay biri bosilganini karab chikarishda.

Funktsiyaga argument sifatida this.form kombinatsiyasini beramiz, chunki u kiyinchilik bizga usha funktsiyaga va xuddi usha bizga kerak elementga murojat kilishga ruxsat beradi. Test1 (form) funktsiya berilgan satr bushligini tekshiradi. If (form.text1.value= = "")... orkali bajariladi. Bu erda uzgaruvchi 'form' ga funktsiya chakirilganda 'this form' dan olingan kiymat beriladi. Biz elementga kiritilayotgan satrni form text iga value suzini kushib aniklab olishimiz mumkin. Satr bush emasligiga ishonish uchun uni " " bilan takkoslab kuramiz. Agar kiritilgan satr " " bilan mos kelsa u xolda satrga xech nima kiritilmagan buladi va bulardan foydalanuvchi xato xakida xabar kabul kiladi. Agar biror bir narsa kiritilgan bulsa foydalanuvchi "ok" tasdigini oladi. Keyingi muammo shundan iboratki foydalanuvchi forma maydoniga probel kiritish mumkin va bu tugri ma'lumot sifatida kabul kilinadi! Agar sida bu probellarni olib tashlash istagi bulsa u xolda albatta unga bu imkoniyatni tekshirishni kushishingiz mumkin. Endi test2(form) funktsiyasini kuramiz. Bu erda yana kiritilgan satr bushlik bilan tekshiriladi. (ukuvchi tomonidan kiritilgan ma'lumot xakikatdan bor yukligiga ishonish uchun). Lekin biz if buyrugiga yana nimadir kushdik. // belgilar yigindisi "ok" buyrugi deb ataladi. If buyrkgi birinchi va ikkinchi takkoslash nima bilan tugallanishini tekshiradi. Agar bulardan bittasi bajarilsa if buyrugi true kiymatiga erishadi va keyingi skript bajariladi. Suz bilan tushuntirganda keltirilgan satr yoki bush yoki unda @ belgisi yukligi xakida xatolik degan xabar olasiz. (if buyrugidagi 2 chi operator kiritilgan satr @ belgisiga ega ekanligini tekshiradi).

II. JavaScript dasturlash tili.

2.1. Javascript haqida umumiy tushunchalar.

JavaScript- bu 1995 yilda Netscape firmasi tomonidan ishlab chiqilgan va skriptlar tuzish uchun mo'ljallangan yangi dasturlash tilidir. Uning yordamida interaktiv Web- sahifalar yaratish mumkin. Quyida siz JavaScript yordamida qanday ishlar qilish mumkinligini ko'rib chiqamiz.

JavaScript- bu dasturlash texnologiyasi bo'lib, *HTML xujjatlarni yaratishda ishlatiladi. Unda makrobuyruq texnologiyasi, ya'ni bir necha buyruqni birmakrobuyruq shaklida tasvirlash keng qo'llanilgan. Bu makrobuyruq matnlari maxsus qoidalar asosida yoziladi. Va u HTML xujjatga kiritiladi.*

JavaScriptning boshqa versiyalari ham mavjud bo'lib, ular Icript, VB Script (Visual Basic Script) va boshqalar. *HTMLning* JavaScript kiritilgan joyda quyidagicha ko'rsatiladi:

<Script Language= «Java Script»>

<Script Language= «J Script»>

<Script Language= « VB Script»>

JavaScript - bu Java emas!

Ko'pgina foydalanuvchilar JavaScript dasturlash tilini bu Java tilining o'zginasi deb xisoblashadi (bu ikki tilning nomida o'xshashlik borligi uchun). Lekin bu unday emas. Biz bu darsda JavaScript va Java ning barcha farqlarini ko'rib o'tmaymiz, faqat shuni yodda tutingki bu ikki til umuman boshqa-boshqadir. Agar bu ikki tilning farqi va o'xshash tomonlarini bilmoqchi bulsangiz Netscape firmasining saytiga yoki ushbu mavzuga tegishli saytlarga murojaat qilishingiz mumkin.

JavaScript ham dasturlash tiliga o'xshab ketadi. U o'z tashkil etuvchilarigaega. Uning tashkil etuvchilari o'zgaruvchilar, massivlar, operatorlar, ob'ektlar, funktsiyalar, hodisalar, komentariyalar va h.k. JavaScript da kichik va katta harflar farqlanadi. Bundan tashqari interpretator dasturi bo'shliqlarni inobatga

olmaydi, har bir operator « ; » bilan tugashi talab qilinadi. Agarda yangi satrdan yozilgan bo'lmasa.

O'zgaruvchilar. O'zgaruvchilar nomlari albatta harflardan yoki chizish belgisi bilan boshlanishi lozim. O'zgaruvchilar tasvirlanishi tavsiya etiladi. Bunda o'zgaruvchilar butun,

haqiqiy, satriy, mantiqiy qiymatlar ko'rinishida bo'lishi mumkin. O'zgaruvchilar son, so'z, belgilar ketma-ketligi yoki ularning kombinatsiyalari. O'zgaruvchilarni belgilash uchun maxsus kalitli so'zlar mavjud. Masalan:

var:

```
var x=3.14159;var y=2; var z=3;
```

Bu yerda x o'zgaruvchisining qiymati 3.14159, u= 2 va z = 3.

Masalan, x=y+z; //bu yerda x=5

JavaScript da satri o'zgaruvchilar. Ular belgilar yig'indisi, belgilar va raqamlardan iborat. Satri o'zgaruvchilar " qo'shtirnoq ichida berildai. Masalan:

```
var str1="Magistr 1-kurs ";
```

var str2="Magistr 2-kurs "; Qo'shtirnoq ichida satri o'zgaruvchilarda "\" belgisini ishlatish xam mumkin. Masalan: var str3="Magistr\s Ok!"; Ushbu belgi yordamida maxsus belgilarni o'rnatish mumkin. Masalan.

\r – orqaga qaytish

\n – yangi satrga o'tish

\t - tabulyatsiya

Nomlash tartibi. O'zgaruvchilar nomi va JavaScriptfunktsiyalari "A"- "Z", "a"- "z")

belgilaridan boshlanadi va (" _ ") chiziq yordamida.

O'zgaruvchilar turlari. O'zgaruvchilarni skriptlarda qo'llashning 2 xil ko'rinishi mavjud.

Global o'zgaruvchi: Ixtiyoriy joyda qo'llash mumkin. Lokal: Mavjud funktsiyada qo'llash mumkin. Lokal o'zgaruvchini funktsiyani ichida e'lon qilishda va qo'llashda xizmatchi so'z **var** qo'llaniladi. Masalan. **Lokal** o'zgaruvchilarni e'lon qilish **var total=0.Global** o'zgaruvchilarni e'lon qilish esa.

total = 0

Butun son (Integer). Butun sonlar 1, 164, 102390 ko'rinishda bo'ladi. Ular berilishiga ko'ra 8, 10, 16 lik sanoq sistemasida beriladi. 0-9 sonlar va a-f va A-F harflar bilan belgilanadi. *JavaScript* da yozilishi 0x yoki 0X ko'rinishda bo'ladi. Masalan: O'n oltilik sanoq sistemada 0x17 sakizlik sanoq sistemasida 027 ko'rinishda yoziladi.

Suzuvchi vergulli o'zgaruvchi

- 3.1415927
- 31415927e-7
- .31415927E1

Mantiqiy o'zgaruvchilar. Mantiqiy ifodalar faqat ikkita qiymatni qabul qiladi. Rost (**true**) yoki yolg'on (**false**). *JavaScript* ba'zi bir xollarda 0 (false) va 1

(true) larni qo'llab bo'lmaydi.

Satrlı o'zgaruvchilar. Satrlı o'zgaruvchilar bitta va bir necha simvollardan tashkil topishi mumkin. Ular (") yoki bittalik (') qo'shtirnoq ichida yozilishi mumkin. Masalan:

- "Blah"
- 'Blah'
- "1234"
- "one line \n another line"

Massivlar. Massivlar nom va indeksga ega bo'ladi. Massivni tasvirlash quyidagicha bo'ladi. arr Massiv nomi: indekslar ro'yxati. Masalan: arr Magistr[i] [j]

Operatorlar. *JavaScript* da arifmetik, mantiqiy, bit, satr ko'rinishdagi operatorlar mavjud. Bundan tashqari operatorlar bloki, shartli va tsikl operatorlari ishlatiladi.

Funktsiyalar. *JavaScript* da funktsiyalar quyidagi ko'rinishda bo'ladi:

Function funksiya nomi ((argument 1), (argument 2)..(argument n))
{operatorlar}.

Kerak bo'lsa funksiya nomiga biror qiymat berilishi mumkin va buning

uchun:

Return (operatorlar).

Ob'ekt. Ob'ektning xossalari deb ataluvchi ba'zi bir berilganlarning (xujjat, oy-yillar, ko'rinishlar va b.) majmuidir. Masalan. Xujjatning ob'ekt quyidagi xossalarga ega: matn rangi, fon rangi, oxirgi o'zgartirish sanasi, xujjat o'tgan URL va shunga o'xshash masalalar.

Xodisalar. Hodisalar JavaScript elementi bo'lib, ular kerakli amallarni ma'lum vaqtlarda ishlatish imkoniyatini beradi. Masalan. Ob'ekt ustida bajariladigan hodisalar majmui.

JavaScript ni ishga tushirish. JavaScript tilida yozilgan skriptlarni ishga tushirish uchun nima qilish kerak? Buning uchun sizga JavaScript bilan ishlay oladigan brauzer kerak bo'ladi. Masalan Netscape Navigator (2.0 versiyasi va undan yuqori) yoki Microsoft Internet Explorer (MSIE - 3.0 versiyasidan boshlab). Ushbu ikki brauzerning keng tarqalishi foydalanuvchilarga JavaScript tilida yozilgan skriptlar bilan ishlash imkoniyati paydo bo'ldi.

2.2. JavaScriptda funksiyalar yozish.

Mijoz tomonidagi sahifani yaratishni boshqarishda hujjatning obyektli mexanizmidan foydalanilgan. Bunda har bir HTML-konteyner-bu obyekt hisoblanadi va quyidagi uchlikni tashkil etadi:

- xususiyatlar
- usullar
- xolatlar

Obyektli model sahifalar va brauzerlar o'rtasidagi bog'lanishsifatida ko'rinishi mumkin. Obyektli model - bu HTML kod orqali berilgan elementlarni obyekt, usul, xususiyat va xolatlar ko'rinishida tanish va ular bilan ishlash demakdir. U yordamida biz brauzerga va foydalanuvchiga murojaat qilishimiz, xabarlar yuborishimiz mumkin. Brauzer bizning buyruqralimizni bajaradi va ekranda sahifaning kerakli qismlarini o'zgartiradi.

Obyektlar bir xil tipli xususiyatlar, usullar va xolatlar to'plamini bir xil tipli obyektlar sinflarida birlashtiradi. Obyektlarning o'zlari faqat hujjatni brauzer yordamida yuklashda yoki dasturning natijasi sifatida namoyon bo'ladi. Ushbu holatni doimo yodda tutish kerak.

Xususiyat

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

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

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

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

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

Umumiylik uchun rasm xususiyatlari JavaScript da HTML-razmetkada mavjud bo'lmagan obyektlarga bo'linadi. Masalan, vosita sifatida Navigator deb nomlanuvchi obyekt, yoki JavaScript dagi eng asosiy obyektlardan - brauzer oynasi obyektini olishimiz mumkin.

Usullar

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

```
function hello()  
{ id=window.open("", "example", "width=400, height=150");  
id.focus(); id.document.open();  
id.document.write("<H1>Salom!</H1>");
```

```
id.document.write("<HR><FORM>");
id.document.write("<INPUT TYPE=button VALUE='Oynani yopish '");
id.document.write("onClick='window.opener.focus();window.close();'>");
id.document.close();
}
```

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

Holat

Usullar va xususiyatlardan tashqari obyektlarni holatlar bilan ham xarakterlash mumkin. SHaxsan, JavaScript da dasturlashda ushbu holatlarni qayta ishlovchi vositalar mavjud. Naprimen, button tipidagi obyekt bilan (INPUT konteyneri bilan button - "Tugma") click holati amalga oshishi mumkin, ya'ni foydalanuvchi tugmani bosishi mumkin. Buning uchun INPUT konteyneri atributi click holatni - onClick holatiga kengaytirgan. Ushbu atribut qiymati sifatida HTML hujjat muallifi tomonidan JavaScript da tuzilgan holatni qayta ishlovchi dastur ko'rsatiladi:

```
<INPUT TYPE=button VALUE="Najat" onClick="window.alert('Marhamat,
yana bir bor bosing');">
```

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

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

2.3. Forma hodisalariga funksiyalarni joriy qilish.

Xodisalar. Hodisalar JavaScript elementi bo'lib, ular kerakli amallarni ma'lum vaqtlarda ishlatish imkoniyatini beradi. Masalan. Ob'ekt ustida bajariladigan hodisalar majmui.

Xodisalar va ularni ishlab chikaruvchilar JavaScript tilida dasturlashtirishning eng zarur kismi xisoblanadi. Xodisalar foydalanuvchining u yoki bu xarakatlaridan tashkil topadi. Agar u kaysidir tugmani bossa, murojaatini kesib utsa, "Mouse Over" xodisasi yuz beradi.

Xodisalarning bir necha turlari mavjud. Biz JavaScript dasturimizda ulardan bir nechtasini bajarishimiz mumkin va buni xodisalarni ishlovchi maxsus dastur yordamida amalga oshirishimiz mumkin. Demak, tugmachani bosish orkali tushib ketuvchi oyna xodisasini yaratish mumkin. Bu shuni anglatadiki, Click – bosish xodisasi bilan oyna yaratish reaksiyasi bulishi kerak.

Joriy vaktida biz ishlatishimiz kerak bulgan dastur – xodisalarni ishlab chikaruvchi OnClick deb ataladi. Va u kompyuterga xodisa ruy berganda nima kilish kerakligini xabar beradi.

Berilagan misol bir nechta yangi xususiyatlarga ega, ularni tartib buyicha kurib chikamiz. Birinchi yangi xususiyat bu OnClick="alert('Y0')" bulib u <input> tegiga tegishli. Aytib utganimizdek, bu tugmani bosganingizda nima sodir bulishini aniklang.

```
<script language="JavaScript">  
<!-- hide  
Function calculation()  
Var x=12;Var y=5;Var result=x+y;  
Alert (result);  
// -  
</script>  
</head>
```

```
<body>  
<form>  
<input type = "button" value = "calculate" onclick = "calculation()">  
</form>  
</body>
```



Bu yerda tugmachani bosish bilan calculation() funksiyasini chikarish jarayoni ruy beradi. Sezgan bulsangiz, bu funksiya x,y va result uzgaruvchilaridan foydalanib, kushish amalini bajaradi. Uzgaruvchilarni var kalit suzi bilan e'lon qilish mumkin. Uzgaruvchilar turli kattaliklarni son, satr va xakozalarni saklash uchun ishlatiladi.

Skriptning var result = x+y satri brauzerga result uzgaruvchisini xosil kushish va unga x+u arifmetik amalni joylashtirishni (5+12) xabar beradi. Bundan sung result uzgaruvchisi 17 kiymatni oladi. Bu xolda alert (result) va alert(17) buyrugi bir xil vazifani bajaradi. Boshkacha kilib aytganda, bu 17 son yozilgan darchaga ega bulamiz.

Formaga kiritilgan ma'lumotlarni berishning kandy imkoniyatlari mavjud? Forma ma'lumotlarini berishning eng oddiy usuli bu elektron pochta orkali uzatishdir. (bu usulni kengrok urganamiz). Agar siz formaga kiritilayotgan ma'lumotlarga server kuz-kulok bulib turishini xoxlasangiz u xolda siz CGI interfeysidan foydalanishingiz kerak. U sizga ma'lumotlarni avtomatik ravishda kayta nomlashga imkon beradi. Masalan, server mijozlardan ayrimlari kira olish imkoniyati bulgan ma'lumotlar bazasini yaratish mumkin. Boshka misol- yahoo singari kidirish saxifalari. Odatda unda shaxsiy ma'lumotlar bazasidan yaratilgan surovni kidirish uchun forma tashkil kilingan. Natijada foydalanuvchi mos tugmachani bosishdan oldinrok javobni oladi. Unga toki joriy serverni kullab-kuvvatlovchi odamlar ularga kursatilgan ma'lumotlarni ukish va kidirishni kutib utirishga xojat kolmaydi. Bularning barchasini serverning uzi bajaradi. Java S bunday imkoniyatlarni bajarishga yul kuymaydi.

Java S ma'lumotlarni serverdagi iror bir faylga yozib olish imkoniyati bulmagani sababli u yordamida ukuv takrizlari kitobini yarataolmaysiz. Buni fakat CGI interfeysi orkali bajarish mumkin.

Albatta, siz foydalanuvchilar elektron pochta orkali yuborgan ma'lumotlar uchun takriz kitobi yaratishingiz mumkin. Lekin bu xolda siz elektron pochta orkali olingan takrizlarni kulda bitta faylga kiritishingiz kerak. Agar siz xar kuni 1000 takriz olishni nazarda tutmasangiz shunday kilishingiz mumkin. SHunga mos skript oddiy HTML matndan iborat buladi va xech kanday Java S dasturlashtirish kerak emas. Albatta bitta xolni xisobga olmaganda, ya'ni agar sizga ma'lumotlarni yuborishdan oldin tekshirish kerak bulsa bu erda xakikatan xam Java S kerak buladi. Fakat biz xamma erda xam ishlaymaydigan mailto buyrugini kiritamiz, u fakat Microsoft Explorer 3.0 da kullab-kuvvatlanmaydi.

Enctype="text/plain" parametri matnni xuddi uzini xech kodlangan kismalarsiz yuborish uchun ishlatiladi. Bu sezilarli darajada bunday pochtani ukishni soddalashtiradi.

Agar siz formani setga uzatilmasdan tekshirishni istasangiz, u xolda onSubmit xodisalar ishlovchi dasturdan foydalanishingiz mumkin. Siz bu dastur uchun < form > tegida chakirik joylashtirishingiz mumkin. Masalan:

Agar shu tarika tuzilgan formada notugri ma'lumotlar kiritilgan bulsa bu forma internetga yuborilmaydi.

Focus() usuli yordamida formani yanada samimiy kilish mumkin. Demak, siz kaysi element birinchi bulib belgilanishini tanlashingiz mumkin. Demak, siz kaysi element birinchi bulib belgilanishini tanlashingiz mumkin. YOki brauzerga notugri ma'lumotlar kiritilgan foramatni belgilashni buyurishingiz mumkin. YA'ni brazerning uzi kursorni siz kursatgan format elementiga urnatadi va foydalanuvchiga biror narsa kursatishda formatni chikarishga tugri kelmaydi. Buni kuyidagi skript formati orkali kilish mumkin:

```
Function setfocus( ) {  
    Docuent.first.text1.focus( );    }
```

bu yozuv men kursatib utgan kritiluvchi matnning birinchi elementini begilaydi. Siz barcha formalaning jismlarni kursatishingiz kerak, joriy xolda u first deb ataladi va formaning elementi nomi text1. Agar siz saxifa yuklanganda joriy element belgilanishini istasangiz buning uchun siz <body> tegini onload atributi bilan tuldirishingiz mumkin. Bu shunday kurinishda buladi:

```
<body onload = "setfocus()">
```

YAna bir misol bilan tuldirishingizga tugri keladi:

```
Function setfocus( ) {  
    Document.first.text1.focus( );  
    Document.first.text1.select( );    }
```

YAna bir kdni kuring:

Bunda fakat element belgilanmaydi, balki undagi matn xam belgilanadi.

Web, gipermatnli tizim sifatidi ikki xil nuqtai nazar bilan qarash mumkin. Birinchidan, gipermatnli o'tishlarni (ssilkalar - ANCHOR konteyneri), matnlarni ko'rsatuvchi hisoblanadi. Ikkinchidan, sahifada mavjud bo'lgan bir qator obyektlar bilan ishlovchi tizim sifatida.

Xulosa

Bugungi kunda Informatika va axborot kommunikatsiya texnologiyalari sohasi jadal rivojlanmoqda, shu bilan birgalikda kompyuter tarmoqlari ham. Biz yoshlar o'zimiz va o'zimizdan keyingi avlodlarga Informatika va axborot kommunikatsiya texnologiyalari sirlarini chuqurroq o'rganishlari uchun yordam berishimiz kerak.

Web sahifani generatsiya qilish jarayonida "mijoz-server " arxitekturasi bilan bog'liq ravishda dilemmalar hosil bo'ladi. Sahifalar mijoz tomonida ham server tomonidagi kabi generatsiya qilinadi. 1995 yilda Netscape kompaniyasi mutaxassisleri mijoz tomonidagi sahifalarni generatsiya qilish uchun maxsus dasturlash tili yaratishdi va uni JavaScript deb nomdashdi.

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

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

Amaliyotda buni biz, sahifa fonining rangini yoki hujjatdagi rasm xususiyatlarini o'zgartirishda, yangi oyna ochish yoki ogohlantirish berish jarayonlarida yaqqol kuzatishimiz mumkin.

Foydalanilgan adabiyotlar ro'yxati:

1. Karimov I.A. Yuksak malakali mutaxassislar –taraqqiyot omili. T., O'zbekiston, 1995 y, 30 – 31 b.
2. Sh.F. Madraximov va boshqalar, Informatika va programmalash, Toshkent-2005.
3. “Интернет технологияси” фани бўйича амалий машғулотларни ўтказиш учун ўқув-услубий қўлланма. I-семестр.
4. С.Л.Эшқобилов. Интернет – ахборот қидирув. Т–“Фан” нашриёти, 2006й.
5. S.K.Ro'zimov. Kompyuter savodxonligi. T – “Fan” nashriyoti, 2006y.
6. M.M.Aripov, J.U.Muxammadiyev. Informatika. Informatsion texnologiyalar. Darslik. T, 2006y.
7. M.M.Aripov va boshqalar. Информатика. Ахборот технологиялари. Тошкент, ТДТУ, 2002й.
8. О.П.Ильина, Информационные технологии. С-Петербург, 2002.
9. Компьютерные технологии обработки информации. Под ред.С.В.Назарова. М., Финансы и статистика,.2001й.
10. Интернет с нуля. Под редакцией Н.Домина. М. Учебное пособие - 2006й.
11. Castro Elizabeth. HTML 4 for the World Wide Web. Berkeley: Press, 1998
12. В. Дронов JAVA Script в Web – дизайне. С.Петербург. 2001.
13. Будилов В.А. Практические занятия по HTML (учебное пособие) 2004г, 903с.