

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

BUXORO DAVLAT UNIVERSITETI

Fizika –matematika fakulteti

“Informatika va axborot texnologiyalari” kafedrası

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

“Kompyuter grafikasi va web dizayn” fanidan

KURS ISHI

**Mavzu: JavaScript orqali Web sahifa elementlari bilan ishlash
yo’llari.**

Bajardi: 1-1PMIK-13 guruh talabasi Olimova Odina Salimovna.

Kurs ishi himoya qilingan sana

Komissiya a’zolari

“ ____ ” _____ 2016 y.

Ball _____

Buxoro-2016y

Mundarija:

Kirish.....	3
.....	
I. JavaScript dasturlash tili	5
1.1. JavaScript ga kirish.....	5
1.2. JavaScript ning obyektli modeli tushunchasi.....	6
1.3. JavaScript ning URL-sxemasi.....	8
II. JavaScriptda formalar	1
.....	4
2.1. JavaScriptda formalar	1
.....	4
2.2. Web sahifada tasvir	1
.....	9
2.3. JavaScript va qavatlar.....	2
	1
2.4. JavaScript 1.2 da sichkoncha bilan ishlashdagi xodisalar....	
Xulosa.....	3
.....	2
Foydalanilgan adabiyotlar	3
ro'yxati.....	3

Kirish.

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

1995 yilda Netscape kompaniyasi mutaxassislari 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.

"JavaScript" nomi Netscape ning o'zigagina tegishlidir. SHunga o'xshash Microsoft tomonidan ishlab chiqilgan tilning rasmiy nomi Jscript deb ataladi. JavaScript ning bir qator versiyalari JScript ning versiyalari bilan birgalikda ishlaydi (qisman ammo to'liq emas), ya'ni JavaScript tili JScript tilining qism to'plami hisoblanadi.

JavaScript tili ECMA (European Computer Manufacturers Association - Evropa kompyuter ishlab chiqarish assotsiatsiyasi) tomonidan standartlashtirilgan. Ushbu standartlar ECMA-262 va ISO-16262 nomlarini keltirib chiqardi. Bu standartlar JavaScript 1.1 ga mosh tushuvchi ECMAScript tilini taqdim etadi.

Ta'kidlash kerakki, bugungi kunda JavaScript ning barcha turlari ham ECMA standartiga mos tushavermaydi.

JavaScript ning obyektli modeli tushunchasi.

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'lanish sifatida ko'rinishi mumkin. Obyektli model - bu HTML kod orqali berilgan elementlarni obyekt, usul, xususiyat va holatlar ko'rinishida tanish va ular bilan ishlash demakdir. U yordamida biz brauzerga va foydalanuvchiga murojaat qilishimiz, xabarlar yuborishimiz mumkin. Brauzer bizning buyruqlarimizni bajaradi va ekranda sahifaning kerakli qismlarini o'zgartiradi.

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 mavjud. Ushbu atribut uni gipermurojaatga 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. Naprimmer, 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.*

Kodni HTML-sahifaga joylashtirish

Dasturlashni endi boshlovchilar uchun doimo bir savol mavjud bo'ladi: "Dasturni qanday jihozlash va uni qanday bajarish?". Ushbu savolga javob berib ko'ramiz, ammo bunda JavaScript-kodning barcha xususiyat va kattaliklari haqida unutmazlik kerak.

Birinchidan, JavaScript-kodni brauzerda bajarish. Brauzerda JavaScript interpretator o'rnatilgan. Umuman olganda JavaScript ni qo'llashda to'rtta funktsional usuldan foydalanish mumkin:

- gipermatnli murojaat (URL sxema);
- holatni qayta ishlovchi (handler);
- podstanovka (entity) (Microsoft Internet Explorer ning 5.X va yuqori versiyalarida mavjud);

- vstavka (SCRIPT konteyneri).

JavaScript bo'yicha qo'llanmalarda JavaScript ni qo'llanilishi SCRIPT konteyneri orqali beriladi. Ammo dasturlash nuqtainazari bo'yicha bu to'la to'g'ri emas, ushbu tartib quyidagi savolga javob bermaydi: JavaScript-kod boshqaruvni qanday qo'lga kiritadi? Ya'ni JavaScriptda yozilgan va HTML hujjatga joylashtirilgan dastur qanday bajariladi.

HTML hujjat muallifining malakasi va bilim darajasiga qarab JavaScript ning bir nechta usullaridan foydalanishi mumkin.

JavaScript ning URL-sxemasi

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

Ushbu holda URL http - bilan boshlanadi va ruxsat sxemasini (http sxema).

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

```
<A      HREF="JavaScript:JavaScript
kod">...</A>
<IMG
SRC="JavaScript:JavaScript kod">
,
```

Ushbu holda "JavaScript _ kod" matni JavaScript dagi gipermurojaat bosilganda qayta ishlash dasturi hisoblanadi va keyingi holda rasimni yuklash chog'ida qo'llanilishi keltirilgan.

Masalan, Vnimanie!!! nomli gipermatnli murojaat bosilganda ogohlantirish oynasining ochilishi quyidagicha amalga oshiriladi: (ochish) `<a HREF="JavaScript:alert(' Vnimanie!!!');"> Vnimanie!!!</a>`



submit tipidagi tugmani bosish orqali formadagi matn obyekti to'ldirilishi quyidagicha amalga oshiriladi:

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

URL da murakkab dasturlarni joylashtirish va funksiyalarni chaqirish mumkin. SHuni

unutmaslik kerakki JavaScript ning bu sxemasi barcha brauzerlarda ham ishlayvermaydi,

Netscape Navigator tipidagi va Internet Explorer ning to'rtinchi versiyasidan boshlab ishlaydi.

o

Holatlarni qayta ishlovchilar

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

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

Podstanovkalar

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

```
<SCRIPT>
function l()
{
str = window.location.href;
return(str.length);
}
</SCRIPT>
<FORM><INPUT VALUE="&{window.location.href};" SIZE="&{l()};">
</FORM>
<SCRIPT>
<!--Bu izoh ...JavaScript-kod...// -->
</SCRIPT>
<BODY>
Hujjat tanasi
</BODY>
</HTML>
```

Bilamizki, hujjatning sarlavha qismidagi matnlar brauzer oynasida

ko'rinmaydi. SHuning uchun bu qismga hujjat tanasida chaqiriluvchi va ishlatiluvchi o'zgaruvchilar va funksiyalar joylashtiriladi. Bu sohada Netscape Navigator brauzeri Internet Explorerga qaraganda biroz qat'iyroq. Agar hujjat tanasidagi funksiya sarlavha qismida e'lon qilinmagan bo'lsa, ushbu funksiya aniqlanmaganligi haqida xabar beradi.

Funksiyalarni joylashtirish va foydalanishga **misol** ko'ramiz:

```
<HTML> <HEAD> <SCRIPT> function time scroll() {  
d = new Date();  
window.status =  
d.getHours()+":"+d.getMinutes()+":"+d.getSeconds();  
setTimeout('time scroll();',500); }  
</SCRIPT>  
</HEAD>  
<BODY onLoad=time scroll()>  
<CENTER>  
<H1>Holat satridagi soat </H1>
```

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

O'rnatish (SCRIPT konteyneri-interpretorni majburiy chaqirish)

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

SCRIPT konteyneri ikkita asosiy funksiyani bajaradi:

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

Birinchi funksiyasi o'zgaruvchilar va funksiyalarni joylashtirish uchun

qo'llaniladi. Ikkinchisi - bu hujjatni yuklash yoki qayta yuklash vaqtida JavaScript kod natijasini joylashtirishdir.

HTML-hujjatga kodni joylashtirish

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

Sarlavha qismida kod SCRIPT konteyneri orasiga joylashtiriladi:

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

Ushbu misolda biz hujjat sarlavhasida time scroll() funksiyasini yaratdik va unga BODY (onLoad=time_scroll()) konteynerining load holatida murojaat

qildik. Quyidagi funksiyani yaratish va chaqirish orqali alohida oyna yaratish mumkin:

```
function sel()
{
id = window.open("", "example", "width=500,height=200,status,menu");
id.focus();
id.document.open();
id.document.write("<HTML><HEAD>");
id.document.write("<BODY>");
id.document.write("<CENTER>");
id.document.write("<H1>Change text into child window.</H1>");
id.document.write("<FORM NAME=f>");
id.document.write("<INPUT TYPE=text NAME=test SIZE=20 MAXLENGTH=20 VALUE='This is the test'>");
id.document.write("<INPUT TYPE=button VALUE='Close the window' onClick=window.close()></FORM>");
id.document.write("</CENTER>");
id.document.write("</BODY></HTML>");
id.document.close();
}
<INPUT TYPE=button VALUE="Holat satrini o'zgartirish"
onClick="id.defaultStatus='Salom'; id.focus();">
```

Sinflar ierarxiyasi

Obyektga-mo'ljallangan dasturlash tili obyektlar daraxtidan tashkil topadi. JavaScript da bu ierarxik daraxt Window obyektidan boshlanadi, ya'ni har bir obyekt u yoki bu oynada yoziladi. Ixtiyoriy obyektga yoki obyekt xususiyatiga

murojaat qilish uchun undan yuqorida turgang obyekt orqali murojaat qilish kerak bo'ladi:

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

Ba'zan JavaScript ning turli obyektlarida bir xil nomli xususiyatlar bo'ladi. Bu holda dasturchi qaysi obyekt xususiyatiga murojaat qilayotganini aniq ko'rsatishi kerak. Masalan, Window va Document larda location xususiyati mavjud. Faqat, Window uchun bu Location sinfi obyekt, Document - URL da ko'rsatilib yuklanayotgan hujjatni adresini ifodalaydi.

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

JAVASCRIPTDA FORMALAR.

Formalar internetda keng ishlatiladi. Formaga kiritilgan ma'lumotlar tez – tez qayta serverga yuboriladi yoki biror bir adresga elektron pochta orqali jo'natiladi. Muammo shunday, foydalanuvchi tomonidan kiritilgan ma'lumot to'g'riligiga ishonish. Uni internetga yuborishdan oldin Java S tili yordamida oson tekshirish mumkin. Oldin bu formani tekshirish qanday bo'lishini kursatmoqchi edik, keyin esa internetga ma'lumot jo'natishning qanday imkoniyatlari borligini ko'rsatamiz. Dastlab biz oddiy skript yarataylik. Aytaylik html-saxifa 2 ta matn kirituvchi elementga ega bulsin. Ulardan biriga foydalanuvchi o'z ismini, 2 – siga elektron pochta manzilini kiritsin. Siz u erga biror bir ma'lumotni kiritib tugmani bosib ko'ring.

Agar siz u erga hech narsa kiritmagan bo'lsangiz xatolik yuz berish haqida ma'lumot olasiz. 1 – elementdagi istalgan ma'lumot to'g'rilik jihatdan tekshiriladi.

Albatta agar foydalanuvchi boshqa nomni kiritgan bo'lsa u ularni kafolatlamaydi. Brauzer xatolarni sonlarga ham qarshilik ko'rsatmaydi. Masalan, siz '17' deb kiritsangiz u holda siz 'Salom 17' degan salomni olasiz. Demak, bu tekshirish ideal bo'la olmaydi.

Formaning 2 chi elementi bir muncha murakkabroqdir. Unga oddiy satr kirgizib, masalan ismingizni. Siz unga @ ni yozmaguncha bu element ish bajarmaydi. @ belgini mavjudligi bilan foydalanuvchi elektron pochta manzilini to'g'ri kiritgan bo'la oladi. Garchi ma'nosiz bo'lsada birgina @ belgining o'zi ham bu shartga to'g'ri keladi. Internetdagi har bir elektron pochta manzilida @ belgi qatnashadi. Shunday ekan bu belgiga tekshirish bu erda juda o'rinli. Skript bu 2 ta element bilan qanday ishlandi va unda tekshirish qay ko'rinishda bo'ladi. Bu quyidagicha olib boriladi:

Dastlab </body> bo'limidagi kodni ko'rib chiqamiz. Biz bu yerda faqat 2 ta tugmachani matn kiritish uchun yaratamiz. Bu tugmachalar test1(...) va test2 (...) funkstiyalarini qay biri bosilganini qarab chiqarishda.

Funksiyaga argument sifatida this.form kombinastiyasini beramiz, chunki u qiyinchilik bizga o'sha funksiyaga va xuddi o'sha bizga kerak elementga murojaat qilishga ruxsat beradi. Test1 (form) funksiya berilgan satr bo'shligini tekshiradi. If (form.text1.value == "")... orkali bajariladi. Bu erda uzgaruvchi 'form' ga funkstiya chaqirilganda 'this form' dan olingan qiymat beriladi. Biz elementga kiritilayotgan satrni form text iga value so'zini qo'shib aniqlab olishimiz mumkin. Satr bo'sh emasligiga ishonish uchun uni " " bilan taqqoslab ko'ramiz. Agar kiritilgan satr " " bilan mos kelsa u xolda satrga xech nima kiritilmagan bo'ladi va bulardan foydalanuvchi xato xaqida xabar qabul qiladi. Agar biror bir narsa kiritilgan bo'lsa foydalanuvchi "ok" tasdig'ini oladi. Keyingi muammo shundan iboratki foydalanuvchi forma maydoniga probel kiritish mumkin va bu to'g'ri ma'lumot sifatida kabul qilinadi! Agar sizda bu probellarni olib tashlash istagi bo'lsa u holda albatta unga bu imkoniyatni tekshirishni ko'shishingiz mumkin. Endi test2(form) funkstiyasini ko'ramiz. Bu yerda yana kiritilgan satr bo'shlik

bilan tekshiriladi. (o'quvchi tomonidan kiritilgan ma'lumot haqiqatdan bor yo'qligiga ishonish uchun). Lekin biz if buyrug'iga yana nimadir qo'shdik. // belgilar yig'indisi "ok" buyrug'i deb ataladi. If buyrug'i birinchi va ikkinchi taqqoslash nima bilan tugallanishini tekshiradi. Agar bulardan bittasi bajarilsa if buyrug'i true qiymatiga erishadi va keyingi skript bajariladi. So'z bilan tushuntirganda keltirilgan satr yoki bo'sh yoki unda @ belgisi yo'qligi haqida xatolik degan xabar olasiz. (if buyrug'idagi 2 chi operator kiritilgan satr @ belgisiga ega ekanligini tekshiradi).

Ma'lum bir belgilar borligini tekshirish.

Ayrim xollarda formaga kiritilayotganlar faqat ayrim belgilar yoki sonlardan iborat bo'lishi ya'ni ma'lumotni chegaralash kerak bo'lib qoladi. Keltirilgan ma'lumot faqat sonlardan iborat bo'lgan telefon nomerlari haqida eslashimiz kifoya. Bunda sizga keltirilgan ma'lumotlar sonmi yo'qmi, shuni tekshirish kerak bo'ladi. Bu holatda murakkablik shundan iboratki ko'pchilik odamlar telefon nomerlariga yana har xil belgilar, masalan, 01234-5678, 01234/56789 yoki 1234_56789 qo'shishlari mumkin. Foydalanuvchidan telefon nomerlarida bunday belgilarni foydalanishini man qilib bo'lmaydi. Shu bois biz skriptimizni sonlar va bir necha belgilarni tekshiruvchi prostedurani qo'shishimiz kerak. Skriptning dastlabki kodi quyidagicha:

Formaga kiritilgan ma'lumotni taqdim etish.

Formaga kiritilgan ma'lumotlarni berishning qanday imkoniyatlari mavjud? Forma ma'lumotlarini berishning eng oddiy usuli bu elektron pochta orqali uzatishdir. (bu usulni kengroq o'rganamiz). Agar siz formaga kiritilayotgan ma'lumotlarga server ko'z-quloq bo'lib turishini xoxlasangiz u holda siz CGI interfeysidan foydalanishingiz kerak. U sizga ma'lumotlarni avtomatik ravishda qayta nomlashga imkon beradi. Masalan, server mijozlardan ayrimlari kira olish imkoniyati bo'lgan ma'lumotlar bazasini yaratish mumkin. Boshqa misol- yahoo

singari qidirish saxifalari. Odatda unda shaxsiy ma'lumotlar bazasidan yaratilgan so'rovni qidirish uchun forma tashkil qilingan. Natijada foydalanuvchi mos tugmachani bosishdan oldinroq javobni oladi. Unga toki joriy serverni qo'llab-quvvatlovchi odamlar ularga ko'rsatilgan ma'lumotlarni o'qish va qidirishni kutib o'tirishga hojat qolmaydi. Bularning barchasini serverning o'zi bajaradi. Java S bunday imkoniyatlarni bajarishga yo'l qo'ymaydi.

Java S ma'lumotlarni serverdagi biror bir faylga yozib olish imkoniyati bo'lmagani sababli u yordamida o'quv taqrizlari kitobini yarata olmaysiz. Buni faqat CGI interfeysi orqali bajarish mumkin.

Albatta, siz foydalanuvchilar elektron pochta orqali yuborgan ma'lumotlar uchun taqriz kitobi yaratishingiz mumkin. Lekin bu holda siz elektron pochta orqali olingan taqrizlarni qo'lda bitta faylga kiritishingiz kerak. Agar siz xar kuni 1000 taqriz olishni nazarda tutmasangiz shunday qilishingiz mumkin. Shunga mos skript oddiy HTML matndan iborat bo'ladi va xech qanday Java S dasturlashtirish kerak emas. Albatta bitta holni hisobga olmaganda, ya'ni agar sizga ma'lumotlarni yuborishdan oldin tekshirish kerak bo'lsa bu yerda haqiqatan ham Java S kerak bo'ladi. Faqat biz hamma yerda ham ishlamaydigan mailto buyrug'ini kiritamiz, u faqat Microsoft Explorer 3.0 da qullab-quvvatlanmaydi.

Enctype="text/plain" parametri matnni xuddi o'zini hech kodlangan qismlarsiz yuborish uchun ishlatiladi. Bu sezilarli darajada bunday pochtani o'qishni soddalashtiradi.

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

Agar shu tariqa tuzilgan formada noto'g'ri ma'lumotlar kiritilgan bo'lsa bu forma internetga yuborilmaydi.

§ Formaning anik elementini belgilash.

Focus() usuli yordamida formani yanada samimiy qilish mumkin. Demak, siz qaysi element birinchi bo'lib belgilanishini tanlashingiz mumkin. Demak, siz qaysi element birinchi bo'lib belgilanishini tanlashingiz mumkin. Yoki brauzerga noto'g'ri ma'lumotlar kiritilgan formatni belgilashni buyurishingiz mumkin. Ya'ni brazerning uzi kursorni siz kursatgan format elementiga urnatadi va foydalanuvchiga biror narsa kursatishda formatni chikarishga tugri kelmaydi. Buni quyidagi skript formati orqali qilish mumkin:

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

bu yozuv men ko'rsatib o'tgan kiritiluvchi matnning birinchi elementini begilaydi. Siz barcha formalaning jismlarni ko'rsatishingiz kerak, joriy holda 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 to'ldirishingiz mumkin. Bu shunday ko'rinishda bo'ladi:

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

Yana bir misol bilan to'ldirishingizga to'g'ri keladi:

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

Yana bir kodni ko'ring:

Bunda faqat element belgilanmaydi, balki undagi matn ham belgilanadi

Web saxifada tasvir.

Endi Java S ning 1.1 versiyasidan boshlab ishlash mumkin bo'lgan Image ob'ektini ko'rib chiqamiz. (ya'ni netscape navigator 3.0 bilan). Image ob'ekti yordamida siz Web saxifangizga grafik obrazlar bilan o'zgartirishlar kiritishingiz mumkin. Bu bizga qisman multiplikatsiya yaratish imkonini beradi. Shuni eslatamizki, foydalanuvchilar brauzerlarning eski versiyalari orkali ushbu qismda keltirilgan skriptlarda foydalana olmaydi. (Netscape Navigator yoki Internet Explorer 3.0 kabi). Yoki yaxshi holatda ham ulardan to'la effekt olib bo'lmaydi.

Keling dastlab Java S da Web saxifaga tasvirni qanday qilib joylashtirishni ko'rib chiqamiz. Biz ko'rayotgan tilda tasvir massiv ko'rinishda beriladi. Massiv bu image deb nomlanadi va document ob'ektining asoslaridan biri xisoblanadi. Web saxifadagi xar bir tasvir tartib raqamiga ega: 1-tasvir 0 raqamini, 2-esa 1 va x k kabi raqamlarni oladi. Shu tariqa 1-tasvirga biz document.image[0] deb murojaat qilishimiz mumkin. Html dokumentdagi har bir tasvir Image ob'ekti o'zining aniq bir xususiyatiga egaki, qaysiki Java S tilidan unga murojaat qilish mumkin. Masalan, siz uning xususiyatlaridan width va height lar orqali tasvirning o'lchamini aniqlashingiz mumkin. Ya'ni document.image[0].width yozuv orqali Web saxifadagi tasvirning kengligini aniqlashingiz mumkin(nuktalarda). Baxtga qarshi, barcha tasvirning indekslarini o'zgartirib turish qiyin kechadi, ayniqsa agar bir saxifangizda ular yetarli ko'p bo'lsa. Ushbu muammo tasvirlarga o'zining shaxsiy nomini belgilash orqali ham bo'ladi. Demak, agar siz tasvirni ushbu teg

`<img src = "imj.gif" name = "myimage" width = ""100" height="100">`

orqali kiritsangiz u holda siz unga document.myimage yoki document.images["myimage"] deb yozish bilan murojaat qilishingiz mumkin.

Yangi tasvirlarni yuklash.

Albatta bu biz xoxlagan narsa bo'lmasa ham Web saxifada tasvir o'lchamini qanday olishni bilib qo'ygan mumkin. Bu Web saxifada tasvirlar almashinishi joriy qilishni istaymiz va buning uchun bizga src atributi kerak bo'ladi. tegidagi

holatdagi kabi src atributi keltirilgan tasvirning manzilini saqlaydi. Endi siz Java S ning 1.1 versiyasida allaqachon Web saxifaga yuklangan tasvirga yangi adres berish imkoniyatiga egasiz. Va natijada tasvir Web saxifaga eski adresni yangi adres bilan almashtirgan holda yuklanadi. Misol uchun ushbu yozuvni qo'ramiz:

```
<img src = "img1.gif" name = "Myimg" width = 100 height = 100>
```

Bu yerda img1.gif tasvir yuklanadi va Myimg nomini oladi. Navbatdagi satrda oldingi img1.gif tasvir allaqachon yangi img2.gif bilan almashadi:

```
Documunt.MyImages.src = "img2.gif";
```

Bunda yangi tasvir har doim eski tasvir o'lchamini oladi. Va siz endi uning o'lchamini o'zlashtira olmaysiz.

Tasvirlarni ogoxlantiruvchi yuklanishi.

Bunday xarakteristikamizning kamchiliklaridan biri bo'lib, src ga yangi adres berilgandan so'ng unga mos tasvir yuklanish jarayoni boshlanishi hisoblanadi. Va bu oldindan qilinmaganligi sababli yangi tasvir internet orqali uzatilguncha va uz joyiga borguncha bir necha vaqt o'tadi. Ayrim hollarda bu mumkin, lekin bunday tez – tez to'xtalishlarga ko'nish mumkin emas. Bu holat bilan biz nima qilishimiz kerak? Albatta, bu masalaning yechimi bo'lib, tasvirlarni ogoxlantiruvchi yuklanishdan foydalanish hisoblanadi.

Buning uchun bu yangi Image ob'ektini yaratishimiz kerak. Navbatdagi satrlarni ko'rib chiqamiz:

```
HiddenImg = new Images();
```

```
HiddenImg.sec = "img3.gif";
```

1 – satrda yangi Image ob'ekti yaratiladi. 2 – satrda keyinchalik hiddenimg ob'ekti yordamida nomayon bo'ladigan tasvir adresi ko'rsatiladi. Ko'rib o'tganimizdek,

src atributidagi yangi adres brauzerini ushbu berilgan adresli tasvirni yuklashga majbur qiladi. Shuning uchun, bizning misolimizdagi 2-satr bajarilganda img2.gif tasvir yuklana boshlaydi. Lekin HiddenImgning o'z nomidan kelib chiqqan holda ("yashirin tasvir") brauzer yuklashni tamom qilgandan so'ng ekranda tasvir paydo bo'lmaydi. U faqat kompyuterning xotirasida kelgusi foydalanish uchun saqlangan bo'ladi. Ekranga tasvirni chiqarish uchun biz

`document.myImage.src = hiddenImg.src;`

satridan foydalanamiz:

lekin endi tasvir xotiradan darrov olinadi va ekranda namoyon bo'ladi. Shu tariqa biz tasvirni ogohlantiruvchi yuklanishni boshqardik.

Siz anik bir hodisaga reakstiya sifatida tasvirlarning almashinishidan foydalangan holda ajoyib natija olishingiz mumkin. Masalan, sichkoncha ko'rsatkichi saxifaning aniq bir joyiga borganda tasvirni almashtirishingiz mumkin. Navbatdagi misolda oddiy qilib sichkoncha ko'rsatkichi tasvirga joylashtirilganida tekshirib ko'ring (umuman bunda siz agar brauzeringiz faqatgina Java S 1.0 ni k-kuv.digan bo'lsa, xatolar haqida xabar olasiz, bundan qutilishni so'ngroq ko'rib chikamiz).

- O'quvchi Java S 1.1 k-kuv.maydigan brauzerdan foydalanyapti
- 2 chi tasvir yuklatilmagan
- Buning uchun biz Web saxifadagi har bir tasvir uchun yangi komandalar yozishimiz kerak
- Biz shunday skriptga ega bo'lishni xoxlar edikki, qaysiki uni Web saxifalarning ko'pchiligida yana va yana ishlatish imkoni bo'lsa.

Endi biz ushbu muommalarni yechuvchi skriptni to'la variantini ko'rib chiqamiz. Bu skript uzun bo'lganiga qaramasdan siz uni bir yozib olishingiz bilan ushbu

muommalar haqida qayg'urmaysiz. Ushbu skript o'zini ixchamligini saqlab qolishi uchun 2 ta shartni ko'rib chiqish kerak:

Berilgan skript barcha tasvirlarni pics massiviga joylashtiradi. Ushbu massiv eng dastlab chaqirilgan preload () funksiyasi yordamida yaratiladi. Preload () funkstiyasini chaqiruvi oddiygina quyidagi ko'rinishda bo'ladi.

Bu shuni anglatadiki, bu skript serverdan 2 ta img1.gif va img2.gif tasvirlarini yuklashi kerak. Ulardan birinchisini sichqoncha ko'rsatkichi tasvir maydoniga tushmaganda hosil bo'ladigan tasvirdir. Foydalanuvchi sichqoncha ko'rsatkichini tasvirga joylashtirsa, u holda 2 chi tasvir paydo bo'ladi. Preload () funkstiyasini chaqirish maboynida " link 1 " suzini 1 chi argument sifatida ko'rsatamiz va Web saxifadagi Image ob'ektiga oldindan yuklatilgan 2 ta tasvirni berib qo'yamiz. Agar siz bizning misolimizda < body > bo'limini ko'rib chiqsangiz o'sha link 1 nom bilan atalgan tasvirni topasiz. Biz tasvirni Web-saxifada hosil qilish uchun skriptni yozib yurmasdan uning tartib raqamidan emas balki uning nomidan foydalanamiz. On () va off () funksiyalarni ikkalasi ham onMouseOver va onMouseOut hodisalarni qayta ishlash programmasi vositasida chaqiriladi. Image elementining o'zi MouseOver va MouseOut hodisalarini bajarib bilmagani uchun biz bu tasvirlarni murojaat qilishimiz kerak. On () funkstiyasi ko'rsatilgan tasvirdan boshqasini dastlabki holatiga keltirishni ko'rishimiz mumkin. Aks holda darrov bir necha tasvir belgilangan bo'lib qoladi, shuning uchun buni qilish juda zarur. (Gap shundaki agarda foydalanuvchi darxol darcha chegarasidan ko'rsatkichni tasvirga joylashtirsa MouseOut hodisasi registrastiya qilinmaydi).

QISM.QATLAMLAR.

Katlam – bu Netscape Navigator 4.0 brauzerining ajoyib yangi imkoniyatlaridan biridir. U Web – saxifalarni tasvir kabi ob'ektlarini anik joylashtirishga imkon yaratadi. Bundan tashkari siz HTML saxifangizga turli ob'ektlarni qo'yishingiz mumkin. Hattoki ob'ektlarni ko'rinmas qilishingiz mumkin. Java S tilida _____ ni oson boshkarishingiz mumkin. Joriy vakt

_____ dan siz faqat Netscape Navigator 4.0 foydalanishingiz mumkin. Aslini olganda qatlam deganda nimani tushunamiz? Buni tushuntirish oson va buni oddiy misolda, bir necha qog'oz varag'ini olib tushuntiramiz. Bir varaqqa matn yozing. Bashqasiga tasvir chizing va xakoza. Endi bu varaqlarni stol ustiga qo'ying. Faraz qilamiz har bir varaq biror bir qatlam . U har xil ob'ektlarni bizning misolimizda matn va tasvirni ichiga olishi mumkin. Endi tasvirli varaqni olib stol bo'ylab harakatlantiramiz. Diqqat bilan tasvirni varaq izidan qarakatlanishini kuzating. Agar biz varakni o'ngga bursak tasvir ham o'ngga buriladi! Qatlamlar o'zida turli ob'ektlarni saqlashi mumkin, masalan tasvirlar, formalar, matnlar HTML saxifangizga qo'yilishi mumkin. Agar siz biror bir qatlamni harakatlantirsangiz u holda bu qavatdagi qolgan barcha ob'ektlar xam bu xarakatlarni takrorlaydi. Kavatlar bir-biri bilan stol ustidagi oddiy qog'oz varaqlari singari joylashishi mumkin. Har bir qavatda yaltiroq maydon bo'lishi mumkin. Varaqda teshik ya'ni 1 chi varaqdagi “ yaltiroq maydon ” dan pastda joylashgan varaqlar ko'rinadi.

Qavatlar hosil qilish .

Qavat hosil qilish uchun biz < layer > va < ilayer > tegidan foydalanishimiz kerak. Quyidagi parametrlardan foydalanishingiz mumkin:

Name = “ layerName”	Varaq nomi
Left = x position	Chapdan yuqori burchaqni absissasi
Top = y position	Chapdan yuqori burchakni ordinatasi
z-index = layer index	Qavat uchun indeks nomeri
width = layer width	Qavat kengligi (nuqtada)
clip = “ x1-offset, y1-offset,x2-	Qavatning ko'rinuvchi qismini belgilaydi

offset,y2-offset ”	Bizning qavat tagida qaysi qavat borligini aniqlaydi
above = “ layerName”	Qaysi qavat bizning qavat ustidaligini aniqlaydi
below = “ layerName ”	Bu qavatning ko’rinishi
visibility show/hide/inherit	Fon ranggi standart rang nomi yoki rgb yozuv
bgcolor = “ rgbcolor ”	Fon tasvir.
background = “ imageurl ”	

<layer> tegi anik jolashtirishingiz mumkin bo’lgan qavatlar uchun ishlatiladi. Agar siz qavat holatini ko’rsatmasangiz (left top parametrlari yordamida) u holda jimlik holatiga ko’ra u darchaning yuqori chap burchakni egallaydi.

<ilayer> teg yordamida

Keling endi oddiy misoldan boshlaymiz. Biz 2 ta qavat hosil qilmoqchimiz. Ulardan birinchisiga tasvir, ikkinchisiga esa joylaymiz. Bu qilmokchi bo’lgan narsa ushbu matnni joriy tasvir ustida ko’rsatish.

Ko’rib turibmizki <layer> tegi yordamida biz 2 ta qavat shaklini aniqlaymiz. 2 ta qavat ham 200/100 (left va top parametrlari orqali) singari joylashtiriladi. <layer> va </layer> yoki <ilayer> va </ilayer> teglari orasidagi barcha narsalar tavsiflanayotgan qavatga tegishlidir. Bundan tashkari biz ko’rsatilgan qavatlarining paydo bo’lish tartibini z-index parametrlari orqali belgiladik ya’ni bizning holatda siz brauzerga matn tasvir yuqorisida yozilishni xabar beramiz. Umuman olganda, z-indexning eng yuqoridagi qavati barcha qavatlar ustida ko’rsatiladi. Siz z-index

ni tanlashda faqatgina 0 va 1 qiymatlar bilan chegaralangan emassiz-umuman istagan musbat sonlarni tanlashingiz mumkin.

JavaScript va qavatlar.

Endi biz kavatlariga Java S orkali kandy ruxsat olishni kurib chikamiz. Foydalanuvchi tugmachani bosish orkali biror bir qavatni ko'rsatishi yoki berkitish imkoniyatiga ega bo'luvchi misolni ko'rishdan boshlaymiz. Dastlab bu qavatlar Java S da qay tarzda hosil bo'lishini ko'rishimiz kerak. Odatdagidek buning bir necha usullari bor. Eng yaxshisi har bir qavatga nom berib chiqish. Demak, agar biz `</layer>` deb qavat hosil qilsak u xolda biz unga keyinchalik `document.layers["mylayer"]` konstrukstiyasi orqali ruxsat olishimiz mumkin. Bu qavatlariga ruxsatni biz butun sonli indeks orqali olishimiz mumkin. Demak, eng pastki qavatga ruxsat `document.layers[0]` deb yozish bilan olinadi. Shunga e'tibor qaratingki indeks bu z-index parametr bilan bir xil narsa emas. Agar misol uchun siz `layer1` va `layer2` deb nomlanuvchi z-index nomerlari 17 va 100 bo'lgan qatorlarga ega bo'lamiz. U holda siz ularga `document.layers[17]` va `document.layers[100]` deb emas balki `document.layers[0]` va `document.layers[1]` orqali ruxsat olishingiz mumkin. Qavatlar Java S skriptlari yordamida o'zgartirib bo'luvchi bir necha xususiyatlarga ega. Navbatdagi misolda 1 – qavatni bekituvchi yoki aksincha hosil qiluvchi tugmacha tashkil qilingan. (Netscape Novigator +4 yoki undan yuqori versiyalari talab qilinadi). Berilgan tugmacha `ShowHide ( )` funkstiyasini chiqaradi. Bu funktsiya `layer (mylayer)` ob'ektining ko'rinuvchanlik xususiyatiga ruxsat olish masalasi echilgan. `Document.layers["mylayer"].visibility` parametriga "show" yoki "hide" qavatini berib siz ushbu qavatni ko'rsatish yoki berkitish imkoniyatiga ega bo'lasiz. Show va hide bu qirkib olingan kalit so'zlar emas balki satr hisoblanib, siz `Document.layers["mylayer"].visibility = show` deb yoza olmaysiz.

§ Qavatlarni joylashtirish.

Left va top xususiyatlar berilgan qavatning holatini belgilaydi. Siz ularga yangi qavatlar berib uni o'zgartirishingiz mumkin. Masalan navbatdagi satrda qavatning gorizantal holati 200 nuqtada berilyapti:

```
Document.layers["mylayer2"].left = 200;
```

Endi biz qavatlarni aralashtirish dasturiga o'tamiz, u brauzer darchasida qandaydir aylanishlar yaratadi. Skript quyidagi ko'rinishda:

Biz mylayer2 nomli qavat yaratdik. <body> tegida biz onload prostedurasini ishlatganimizni ko'rish mumkin. Bizga saxifa yuklatilgandan darrov qavat aylanishini boshlash zarur. Onload hodisalarini qayta ishlash prostedurasida biz setInterval () funksiyasidan foydalanamiz. Bu Java S versiyasining yangi usullaridan biridir. Undan biror bir funksiyani yana va qayta yana aniq bir intervalida chaqirishda foydalaniladi. Oldin biz buning uchun setTimeout vaqt funksiyasidan foydalanganmiz. SetInterval funksiyasi ham xuddi shunday ishlaydi, biroq siz uni bir marta chaqirsangiz yetarli bo'ladi.

SetInterval funksiyasida biz move () funksiyasini har bir 20 millisekunddan chaqirish imkoniga egamiz. Move () funksiyasi esa o'z navbatida har 20 millisekunddan qavatni yangi sohaga almashtiradi. Bu funksiyani yana va yana chaqirishimiz natijasida matnimizni tez harakatlanishini ko'ramiz. Move() funksiyasida qiladigan narsamiz shuki qavat uchun yangi koordinata hisoblash va uni yozish: `document.layers["mylayer2"].left = pos.`

§ Drag va Drop nima.

Java S 1.2 ni xodisalarining yangi modeli va kavatlar mexanizmi yordamida Web saxifamizda drag va drop sxemasini ishlatish imkoniyatiga ega bo'lamiz.

Drag va drop nima? Masalan, bir qancha OS lar (windows 95/NT yoki Mac OS) bizga fayllarni axlat qutisiga olib borgan holda tozalashga ruxsat beradi. Boshqacha qilib aytganda siz fayl tasviriga sichqoncha tugmasini turtasiz va uni

axlat qutisiga olib borasiz – drag va so'ng uni u yerga tashlaysiz – drop. Biz bu erda amalga oshirmoqchi bo'lgan drag va drop mexanizmi Web saxifada chegaralanadi. Shuning uchun siz HTML caxifadagi ob'ektlarni kompyuteringizning kattik diski yoki boshqa xususiy harakatlarga keltirish uchun bu yerda keltirilgan haddan foydalana olmaysiz. JavaScript +tili to'g'ridan to'g'ri Drag va Drop mexanizmini qo'llab quvvatlamaydi. Bu bizda image ob'ekti uchun drageable yoki biror bir xususiyatni belgilash imkoni yo'q degan ma'noni beradi. Shuning uchun biz buning uchun zarur kodni o'zimiz yozishimiz kerak. Lekin bu unchalik qiyin emas. Demak bizga nima kerak? Bizga ikki narsa kerak. Birinchidan biz sichqoncha bilan bog'liq aniq xodisani aniqlashimiz kerak, ya'ni qanday tushunishimiz kerak. So'ngra biz ekran bo'ylab ob'ektlarni qanday qilib o'tkazishni o'ylashimiz kerak. Albatta biz ob'ektlarni yaratish va ularni ekran bo'ylab joylashtirishni qavat singari yangi imkoniyatdan foydalangan holda bajaramiz. Har bir ob'ekt o'zining qavati bilan namoyon bo'ladi.

JavaScript 1.2 da sichkoncha bilan ishlashdagi xodisalar.

Biz sichqoncha bilan ishlashda yuz beradigan xodisalardan qaysi birini ishlatishimiz kerak? Bizda MouseDrag kabi hodisa yo'q lekin bu uni Mouse Dam ,Mouse Move va Mouse up hodisalarni qayta ishlab berishimiz mumkin. Javas ni 1.2 versiyasida hodisalarninig yangi modulidan foydalaniladi . Biz bularsiz ushbu masalani echib bilmaymiz . Keling bu yangi modelning bir qancha muhim qismiga yana bir ko'rib chiqaylik. Foydalanuvchi brauzer darchasining biror bir joyida sichkoncha tugmasini bosadi. Bizning skript bu hodisa qaysi ob'ekt (ya'ni kavat) bilan bog'liqligini va bu hodisani bajarish kerak. Bizga ushbu hodisa koordinata nuqtalarini bilishimiz kerak. Java S 1.2 da ushbu nuqtalarni koordinatalarini ro'y berishining saqlaydigan yangi Event ob'ekti tashkil qilingan. Hodisani qamrab olinishi boshqa bir muhim momentni uzx ichiga oladi. Masalan agar foydalanuvchi sichkoncha tugmachasini turtsa, u holda unga mos xodisa to'g'ridan-to'g'ri Button ob'ektiga yuboriladi. Lekin bizning misolimizda bu xodisa Window ob'ektida

ishlatilishi kerak. Shuning uchun biz darcha ob'ektiga sichqoncha bilan bog'liq hodisa haqidagi signalni qamrab olishga ruxsat beramiz. Bu navbatdagi misolda namoyon bo'ladi. (Click xodisasida) siz brauzer darchasining ixtiyoriy joyiga sichqoncha bilan turtishingiz mumkin. Bunda darchada ushbu hodisa ro'y bergan joyning koordinatasi haqidagi xabar darchasi namoyon bo'ladi.

Window.onClick = displayCoords; satri shuni anglatadiki, click hodisasi sodir bo'lganida nima sodir bo'ladi. Aniqroq qilib aytganda brauzerning click hodisasiga reaksiyasi sifatida displayCoords () funksiyasi chaqiriladi (displayCoords dan so'ng qavs qo'yish kerak emasligini e'tiborga oling). Uz navbatida displayCoords funksiyasi quyidagi ko'rinishda aniqlanadi:

```
Function displayCoords (l) {
```

```
Alert ("x:"+l.page.x+ "y:"+l.page.y);
```

Ko'rib turganingizday bu funksiya argumentga ega (uni l deb atadiq). Xaqiqatdan esa bu Event ob'ekti bo'lib bu displayCoords () funksiyasiga ishlatilishiga beriladi. Event ob'ekti pege x va page y kabi xususiyatlarga ega, ulardan hodisa sodir bo'lgan joy nuqtasini koordinatalarini olish mumkin. Xabar darchasi esa faqatgina shu qiymatni ko'rsatadi.

MouseDown, MouseMove va MouseUp.

Aytib o'tganimizdek Java S da MouseDrag hodisasi yo'q. Shuning uchun biz Drag va Drop mexanizmini hosil qilishda MouseDown, MouseMove va MouseUp hodisalaridan foydalanishimiz kerak. Navbatdagi misolimizda MouseMove yordamida sichqonchaning joriy koordinatalari holatlar satrida namoyon bo'ladi.

Event.MouseMove ni yozishda siz MouseMove so'zini bosh harflar bilan yozishingiz kerakligiga e'tiborni qarating. MouseMove hodisasi sodir bo'lganda qaysi funksiya chaqirilishini ko'rsatishga kelganda esa siz uni kichkina harflar bilan yozishingiz kerak bo'ladi:

`Window.on mousemove = ...`

Endi biz oxirgi 2 ta misolni birlashtirishimiz mumkin. Biz foydalanuvchi sichqonchani tugmachasini bosganda uning koordinatasi paydo bo'lishini xoxlaymiz. Misolning kodi quyidagi ko'rinishda bo'ladi:

1 chidan biz window ob'ektini MouseDown va MouseUp hodisasi haqida sichqonchani qabul qilishga majbur qilamiz.

`Window.CaptureEvents ( Event.MouseDown/Event.MouseUp ) ;`

Ko'rib turganingizdek biz / (yoki) belgisidan window ob'ekti ko'rsatilgan hodisalardan bir nechtasini qabul qilish kerak degan maqsadda foydalanamiz. Ko'rsatilgan hodisalar joyga ega bo'lsa unda nima sodir bo'lishini quyidagi 2 ta satr orqali tavsiflanadi.

`Window.On MouseDown = StartDrag;`

`Window.On MouseUp = EndDrag;`

Quyidagi satrda MouseMove hodisasi ro'y berib window ob'ekti signal qabul qilganda nima sodir bo'lishi aniq bo'lgan.

`Window.On MouseMove = MoveIt;`

Lekin to'xtang axir biz Event.MouseMove ni window.captureEvents () da aniqlamadikku! Bu shuni anglatadiki joriy hodisa window ob'ekti tomonidan qamrab olinmaydi. Unda nimaga agar window ob'ekti bu hodisa haqida hech nimaga ega bo'lmasa, biz window ob'ektiga MoveIt () funkstiyasini ko'rsatamiz! Bu savolga javobni MouseDown hodisasi sodir bo'lgandan so'ng chaqiriladigan StartDrag () funkstiyasida topishingiz mumkin:

`function StartDrag (l) {`

`Window.CaptureEvents ( Event.MouseMove);`

Bu shuni bildiradiki sichqoncha tugmachasi bosilgandan darrov window ob'ekti MouseMove xodisasini qamrab oladi. Va agar MouseUp hodisasi sodir bo'lsa biz MouseMove hodisasini to'xtatishimiz kerak. Bu EndDrag () funksiyasida ReleaseEvents () usuli yordamida bajariladi.

MoveIt () funksiyasi sichqonchani koordinatalarini xolatlar satriga yozadi. Endi bizda Drag va Drop mexanizmini amalga oshiruvchi skriptning barcha elementlari bor. Va biz endi ob'ektlarimizni ekranda chizishni boshlasak ham bo'ladi.

§ Harakatlanayotgan ob'ektlarni ko'rsatish.

Oldingi darslarimizda biz qanday qilib qavatlar yordamida ob'ektlarni joylashtirishni yaratishni ko'rgan edik. Bu endi foydalanuvchi qaysi qavatdagi sichqoncha tugmasi bilan turtganini aniqlashimiz kerak. Misolni

Drag Obj massivi foydalanuvchi joylashtirmokchi bo'lgan qavatlarni o'zining ichiga oladi. Xar bir qavat Drag Obj da biror bir nomerga ega. Uni biz keyinroq ko'ramiz.

Sichqoncha bilan bog'langan hodisani qamrab olish kodini (biz oldin xam ishlatgan) yana ishlatganimizni ko'rish mumkin.

Window.CaptureEvents (Event.MouseDown/Event.MouseUp);

Window.On MouseDown = StartDrag;

Window.On MouseUp = EndDrag;

Window.On MouseMove = MoveIt;

Biz Stand Drag funksiyasiga quyidagini qo'shdik.

Current Obj = which Obj (l);

Which Obj (l) funksiyasi qaysi ob'ektga turtish bajarilganini aniqlaydi. U mos qavat nomerini qaytaradi. Agar hech qaysi qavat belgilanmagan bo'lsa u null

qavatni qaytaradi. Olingan qiymat current Obj o'zgaruvchisida saqlanadi. Bu shuni anglatadiki, joriy vaqtda biz joylashtirmoqchi bo'lgan qavatning nomerini current Obj dan olishimiz mumkin (agar hech qaysini o'tkazmoqchi bo'lmasak bu null qavatda bo'ladi).

Which Obj funksiyasida siz har bir qavatning left, top, width va height xususiyatini tekshirishimiz mumkin. Ushbu qiymatlar yordamida bir foydalanuvchining qaysi ob'ektni tugmacha bilan turtganini tekshirishimiz mumkin.

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 mutaxassislari 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. Informatika. Axborot texnologiyalari. Toshkent, TDTU, 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