

MIJOZ TOMONIDA DASTURLASH. JAVASCRIPT NI HTML- HUJJATLARGA JOYLASHTIRISH

O‘tanov Xondamir Jo‘ra o‘g‘li

Jontemirov Sarvar Azamat o‘g‘li

SHDPI Biologiya yunalishi talabalari

xondamirotanov@gmail.com +99888 411 1306

Jontemirov Sarvar Azamat o‘g‘li

SHDPI Biologiya yunalishi talabalari

sarvarjontemirov028@gmail.com +99888 589 5868

Annotasiya: ushbu maqolada mijoz tomonida dasturlash, Javascript ni HTML hujjatlarga joylashtirish jarayoni Xaqida ma'lumot keltirilgan bulib 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.ushbu jarayon Xaqida batafsil ma'lumot birilgan.

Kalit so‘zlar: JavaScript, Netscape, ECMA, mijoz-server, Web, gipermatnli,

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.

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.

“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’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 ...>... HREF atributi mavjed. Ushbu atribut uni giper murojaatga aylantiradi:

intuit

Agar <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 brauzer, 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 unutmash 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 JavaScript da 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.

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.

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

O'rnatish (SCRIPT konteyneri-interpretatori majburiy chaqirish)

SCRIPT konteyneri – bu podstanovka usulining rivojlangan varianti hisoblanadi. Jumladan, SCRIPT odatda Server Side Includes, ya'ni server tomonidagi hujjatlarni generatsiya qiluvchi ham deb ataladi. Interpretator SCRIPT teglari orasidagi barcha 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=t  SIZE=20  MAXLENGTH=20
VALUE='This
Is the test'>");
Id.document.write("<INPUT  TYPE=button      VALUE='Close      the
      window'
onClick=window.close()></FORM>");
id.document.write("</CENTER>");
id.document.write("</BODY></HTML>");

```

```
id.document.close();
}
```

```
<INPUT TYPE=button VALUE="Holat satrini o'zgartirish"
onClick="id.defaultStatus='Salom'; id.focus();">
```

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 turgan 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 hil nomli hususiyatlar bo'ladi. Bu holda dasturchi qaysi obyekt xususiyatiga murojaat qilayotganini aniq ko'rsatishi kerak. Masalan, Window va Document larda location hususiyati mavjud. Faqat, Window uchun bu Location sinfi obyekt, Document – URL da ko'rsatilib yuklanayotgan hujjatni adresini ifodalaydi.

Xulosa qilib aytganda: 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().

FOYDALANGAN ADABIYOTLAR:

1. Berdiyeva, Gulnoza, and Avazbek Narzulloyev. "VIRTUAL BORLIQ VA UNING TURLI SOHALARDA QO 'LLANILISHI." Solution of social problems in management and economy 3.4 (2024): 18-23
2. Berdieva, Gulnoza. "The Role, Importance And Relevance Of Information Technology In The Motivational Phase Of Teaching." The American Journal of Applied sciences 3.04 (2021): 334-338.
3. Berdiyeva, G. "INFORMATIKA VA AXBOROT TEXNOLOGIYALARI FANINI OQITISHDA MASOFAVIY TALIM TEXNOLOGIYALARINING AHAMIYATI." Экономика и социум 12-1 (91) (2021): 146-150.
4. Berdieva, Gulnoza, and Named after Nizami. "The importance of students' use of information technology in computer science." (2021).
5. Бердиева, Гулноза. "Электрон таълимни ташкил этиш ва ахборот тизимлари таҳлили" Муғаллим ҳам узликсиз билимлендириу Илмиу-методикалик журнал 3.3(2022):121-123

6. Berdiyeva, Gulnoza. "TA'LIMDA MULTIMEDIA VOSITALARIDAN FOYDALANISH." THEORY AND ANALYTICAL ASPECTS OF RECENT RESEARCH Turkey International scientific-online conference 4.1(2022): 150-152.

7. Jurayev, Suhrobjon. "GENERAL ISSUES OF IMAGE PROCESSING AND ANALYSIS AND CAPABILITIES OF WEB PROGRAMMING LANGUAGES." Академические исследования в современной науке 3.4 (2024): 97-108.

8. Jurayev, Suhrobjon. "TASVIRLARGA ISHLOV BERISH VA TAHLIL ETISHNING UMUMIY MASALALARI VA WEB DASTURLASH TILLARINING IMKONIYATLARI." Инновационные исследования в современном мире: теория и практика 3.1 (2024): 113-122.

9. Saloxiddin o'g'li, Jurayev Suxrob, and Botirov Odilbek Xayrullo o'g'li. "MODERN SERVER TECHNOLOGIES." НАУЧНЫЕ ДОСТИЖЕНИЯ И ОТКРЫТИЯ 2019: сборник статей X (2019): 53.

10. Shamsiddinov, G'iyosjon, Barchin Ro'ziqulova, and Laziza Inatillayeva. "BOSHLANG 'ICH TA'LIMDA AXBOROT TEXNOLOGIYALARIDAN FOYDALANISH USULLARI VA AFZALLIKLARI." Педагогика и психология в современном мире: теоретические и практические исследования 3.10 (2024): 39-41.

11. Shamsiddinov, G'iyosjon, Umida Nurmaxmatova, and Durdona Turayeva. "INFORMATIKA VA RAQAMLI TEXNOLOGIYALARNING TA'LIM JARAYONIDAGI O'RNI." Science and innovation in the education system 3.4 (2024): 102-105.