

Javlon ABDULLO

MUKAMMAL DASTURLASH

2-kitob:

JavaScript

3-nashr

Toshkent
«Akademnashr»
2022

UO'K: 004.42
KBK: 32.973-018
A 15

A 15 Abdullo, Javlon

JavaScript [Matn]: ilmiy-ommabop / Abdullo Javlon. –
Toshkent: Akademnashr, 2022. – 300 b.

ISBN 978-9943-6501-6-9

UO'K: 004.42
KBK: 32.973

Ushbu kitob zamonaviy dasturlashni o'rganmoqchi bo'lganlar, dasturlash bilan shug'ullanadiganlar va hatto dasturiy ta'minot yaratish bo'yicha yetuk mutaxassislariga ham mo'ljallangan bo'lib, ilm olish va uni mustahkamlash uchun eng kerakli bo'ladigan zaruriy bilimlarni o'zida jamlagan hamda sodda tushunarli tilda bayon etilgan. Ortiqcha ma'lumotlar o'rniga o'n yillab yig'ilgan tajribalardan foydalanib aniq ko'rsatmalar keltirilgan. Kitobdagi ma'lumotlarni o'zlashtirgan o'quvchi malakali veb-dizayner sifatida keng faoliyat yuritshdan tashqari, nafaqat «front-end» (timusti) dasturlovchi bo'lishi mumkin, balki to'laqonli veb-dasturiy ta'minot yaratish borasida yetuk bilimlarga ham ega bo'ladi.

*Kitobda yoritilgan mavzular bo'yicha fikr-mulohazalar, takliflar, savollar hamda o'z amaliy dasturlash faoliyatingizda muammolar yuzaga kelsa, **ilm.yurt.uz** sayti orqali murojaat qilishingiz mumkin.*

Ushbu kitobni yoki uning sahifalarining nusxasini muallif ruxsatisiz har qanday ko'rinishda tarqatish qonunga muvofiq ta'qib etiladi. Undagi ma'lumotlardan foydalanilganda kitob nomi qayd etilishi shart.

ISBN 978-9943-6501-6-9

© Javlon ABDULLO
«JavaScript»
© «Akademnashr», 2022

1-BO'LIM

KIRISH

1.1. Veb-dasturlash tili: yaratilish tarixiga bir nazar

JavaScript – dunyoda eng ommabop va juda keng tatbiq etiladigan veb-dasturlash tilidir. Hozirda barcha veb-brauzerlar ushbu tilni qo'llagani bois uning joriy etilish qamrovi nihoyatda keng, u har bir sohaga faol kirib borgan.

Ushbu til haqida mukammal tasavvurga ega bo'lish uchun, avvalo, veb-texnologiyaning ishlash uslubi bilan tanishib chiqish lozim bo'ladi. Sodda qilib aytganda, brauzerda ishorat yozib, unga murojaat qilinganda, ko'rsatilgan manzildagi serverga so'rov yuboradi. U yerdagi tegishli dasturiy ta'minot mos ravishda javob qaytaradi. Brauzer ham, o'z navbatida, kelgan javob bo'yi-cha natijani aks ettiradi. Umuman olganda, brauzer bajaradigan barcha xatti-harakatlar foydalanuvchining o'z kompyuterida (yoki maxsus qurilmasida) amalga oshiriladi, buni foydalanuvchi tomoni sifatida qaraladi.

Juda keng ommalashayotgan HTMLga qo'shimcha imkoniyatlar berish uchun u bilan ishlaydigan dastur-

lash tili yaratishga ehtiyoj sezilgan edi. Bunday dasturlash tili shunday qulay bo'lishi kerak ediki, toki dasturlash bo'yicha chuqur bilimga ega bo'lmaganlar ham, faqat joylashtiruv tilini o'zlashtirganlar ham uni qiyinchiliksiz qo'llay olishsin. JavaScriptgacha mavjud dasturlash tillari operatsion tizimda ishlashga mo'ljallangan bo'lib, hech biri ochiq o'giriluvchan emasdi.

1995-yil bunday talablarga javob beradigan sodda, ochiq kodli dasturiy ta'minotni qo'llaydigan, o'girib-ishlatuvchi vositani mavjud «Netscape»ga yangi 2.0-talqini (versiyasi) sifatida biriktirib, tez vaqt ichida muomalaga chiqarish zarurati tug'ildi. Agar biroz kechikilsa, «Microsoft» kompaniyasi o'z mahsuloti bilan axborot texnologiyalari bozorini zabt etishi turgan gap edi. JavaScript yaratuvchisi Brendan Eyx xotiralarida yozishicha, u o'n kun ichida tinimsiz izlanib yangi tilning yadrosini yaratadi.

«Netscape Communications» kompaniyasi ta'sischilaridan Mark Andressen va «Sun Microsystems» ta'sischilaridan Bill Djoy, Brendan Eyx boshchiligida Java dasturlash tilining sodda qismini o'z ichiga olgan va sintaksisi jihatidan «C» dasturlash tiliga o'xshatilgan ochiq kodli til yaratishdi. Mazkur til tarkibi jihatidan «Self», «Smalltalk» va «Lisp» dasturlash tillariga yaqin ko'rinishga keltirildi. Albatta, ushbu o'xshashliklar va sodda sintaksisli ekani tufayli ushbu til ilk namoyishdayoq axborot texnologiyalari sohasidagi yetuk 28 ta kompaniya tomonidan tan olindi. Ular o'z mahsulotlarida shu tildan foydalanishni, uni amaliyotga tatbiq

etishni lozim topishdi. Lekin shoshilinch tayyorlangan til o'ziga yarasha nuqsonlarga ham ega edi. Ko'p o'tmay «Microsoft»ning undan ko'ra mukammal hisoblangan va uning o'rnini bosa oladigan «Visual Basic» asosidagi «JScript» hamda «VBScript» tillari yaratildi.

Ammo «Netscape»da mavjud kamchiliklar vaqtida, tez bartaraf etilganligi yuqoridagilarning ommalashishiga yo'l bermadi. Shu bilan birga, «Netscape» o'z tilini «ECMAScript» (European Computer Manufacturer's Association) nomi ostida standartlashtirishga erishdi.

2000-yillargacha deyarli barcha hisob-kitoblar, bajariladigan ishlar server tarafida amalga oshirilgani tufayli JavaScript mukammal til sifatida dasturlovchilar tomonidan u qadar keng qo'llanilmadi. Veb-texnologiya rivojlanib borgani sari ming-million foydalanuvchilarga ma'lumot uzatadigan serverlarda so'rovlarni kamaytirishga zarurat tug'ildi. Bundan tashqari, foydalanuvchi (mijoz) tomonida dasturning bajarilishi, faqatgina yagona foydalanuvchi uchun mo'ljallangani bois tez amalga oshishi bilan birga transportga ketadigan vaqtdan ham yutishga imkon beradi. Chunki server nafaqat butunlay boshqa joyda, balki ummon ortida, bir necha ming kilometr olisda joylashishi mumkin.

2005-yilga kelib «Google» kompaniyasi JavaScript yordamida veb-xaritani taqdim etdi va uning juda katta imkoniyatlarga egaligini butun jahonga namoyon qildi. Faqat JavaScript bilan ishlovchi «Google»ning katta jamoasi o'z sohasini kengaytirish maqsadida serverda ham aynan shu til orqali ishlovchi «**node.js**»ni taqdim

etishdi. Hozirgi kunda JavaScript nafaqat foydalanuvchi tomonida, balki serverda ham ishlaydigan mukammal dasturlash tili sifatida keng taqdim etilmoqda.

«Google» va axborot texnologiyalari sohasida ishlovchi boshqa yetakchi kompaniyalar hamda keng dasturlovchilarning talab va takliflarini inobatga olgan holda 2010-yilga kelib JavaScriptning 1.8.5 talqini yaratildi. Albatta, unda 15 yil oldingisiga nisbatan ulkan imkoniyatlar qoʻshilgan (masalan, JQuery imkoniyatlarini deyarli qamrab olgan), katta oʻzgarishlarga ega va xatarli nuqsonlardan xoli, shu bilan birga, ommabop brauzerlarda bir xilda natija beradigan yagona tarkibga keltirilgan.

1.2. Qoʻllanish sohasi

Dastavval JavaScript faqat veb-brauzerlar uchun foydalanuvchining qurilmasida ishlaydigan dasturiy taʼminot sifatida yaratildi. Soʻngra esa, yuqorida qayd etilgani kabi, serverda faoliyat yuritadigan talqinlari ham ixtiro etildi. Ushbu kitobda asosan foydalanuvchi (mijoz) tomonida — veb-brauzerda ishlaydigan asl JavaScript imkoniyatlari haqida atroflicha soʻz yuritiladi. Serverning oʻzida JavaScript asosida ishlaydigan dasturiy taʼminotlar (**node.js**, **nitro**, **rhino**, **sling** kabi) bayon etilmaydi. Umuman olganda, serverda ishlaydigan boshqa dasturiy taʼminotlar koʻp. Shuni nazarda

tutsak, asosini JavaScript tashkil etadigan ta'minotlar ommaviy qo'llaniladi deb bo'lmaydi. Ammo dasturlovchilarning ko'pi ommaviy tarzda faqat JavaScriptdan foydalanishadi. Barcha brauzerlar uni qo'llaydi. Xuddi HTML singari oddiy matn tahrirlovchi muhit orqali yozilgan kodni ixtiyoriy veb-brauzer orqali aks etishini ko'rish mumkin. Amalda esa faqat kompyuterdagi veb-brauzerlardan tashqari smartfonlarda, planshetlarda, veb-orazli (interfeysli) barcha oynali qurilmalarda va hattoki televizor va soatlarda ham ma'lum doiradagi dasturlar tuzish va natijasidan bahramand bo'lish mumkin.

Texnik jihatdan HTML namoyon bo'lgan joyda JavaScript ham ishlaydi, faqat maxsus tarzda brauzerning sozlamasida o'chirib qo'yilmagan bo'lishi kerak. Ushbu kitobda hozirda eng so'nggi hisoblangan (aslida barcha imkoniyatlari haligacha to'liq tatbiq etib bo'linmagan) **ECMAScriptning 10**-talqinidagi (ECMAScript 2019) to'ldirishlar bilan JavaScript imkoniyatlari xususidagi ma'lumotlar boyitilib, asoslantirilib, misollar bilan batafsil yoritilgan. Keltirilgan ilmlar barcha (desktop va mobile) brauzerlarda bir xil ishlaydi, ba'zi maxsus istisnolar o'z o'rnida bayon etilgan.

Biz misol tariqasida ilk mukammal brauzer bo'lgan «Netscape» o'rniga JavaScript ixtirochilari tomonidan taqdim etilgan **Firefox** va undan qanday foydalanilishi xususida batafsil so'z yuritimiz. Negaki, u ochiq kodli va keng ommaviy tarzda takomillashtirilib boriladigan, dasturlovchi uchun qulay imkoniyatlarga ega brauzer

hisoblanadi. Albatta, hozirda **Google Chrome** ham undagi deyarli barcha imkoniyatlarni amalga oshira oladi, shuning uchun ham ko'pchilik eng ommaviy brauzer sifatida **Chromeni** tanlashi mumkin. Tanlovni o'rganuvchining o'ziga qoldiramiz va har ikkisida bir xil bo'lgan dasturlovchilar uchun mo'ljallangan muhit haqida qisqacha to'xtalib o'tamiz.

Birinci qadamda «Mozilla Firefox» yoki «Google Chrome» brauzerlaridan birini operatsion tizimga o'rnatamiz. So'ngra «F12» tugmasini (Ctrl+Alt+I yoki operatsion tizim va brauzer talqiniga qarab boshqa tugmalar birikmasi bo'lishi mumkin) bosish orqali maxsus muhitni chaqiramiz. Unda dasturlashni o'rganmoqchi bo'lganlar uchun murakkablik kasb etmaydigan, sodda interfeys (dasturiy ta'minot orazi) mavjud. Bizga birinchi o'rinda kerakli bo'ladigani — «konsol» bo'limidir.

Konsol — alohida (vab-sahifa namoyon bo'ladigan maydondan tashqarida, odatda, yashirin tarzda bo'ladigan) bajarilayotgan amallardagi xatoliklarni kuzatib borish yoki qo'shimcha JavaScript kodini ishga tushirish uchun muloqot darchasi. Buni to'liq tasavvur qilish maqsadida quyida dasturlovchi uchun mo'ljallangan muhitning ko'rinishini keltiramiz:



Ushbu muhit orqali o'rganish jarayonida yozilgan kodlarning natijasini tez ko'rishdan tashqari, ularning

bajarilishidagi o'zgarishlarni qadam-baqadam kuzatib borish imkoniyati ham mavjud. Konsol imkoniyatlari va undan unumli foydalanish bo'yicha keyingi bo'limlarda [3.4.3.] amaliy mashg'ulotlar yordamida batafsil tanishtirib o'tamiz.

2-BO'LIM

DASTURLASH ASOSLARI

2.1. Xabar ko'rsatish

Dasturlashni o'rganishga kirishishdan avval faqat JavaScriptgagina xos bo'lgan xabar chiqarish uslublari bilan tanishib o'tsak, ular birmuncha sodda va kod yozish jarayonida foydalanuvchi bilan muloqot qilish uchun keng qo'llaniladi.

2.1.1. Ogohlantirish (alert)

Ogohlantirish (alert) — ekranda maxsus oyna ichida xabar chiqarib, to u yopilguncha navbatdagi kodlarning bajarilishini to'xtatib turadi. Sintaksisi:

```
alert(xabar)
```

Ushbu buyruq berilganda xabar alohida oynada, sahifaning eng ustki qatlamida paydo bo'ladi va qolgan obyektlar nafaol tusga o'tadi. Toki u yopilmaguncha («ESC» yoki «OK» tugmasi bosilguncha) brauzer barcha yuklanish va kodlarning bajarilishini to'xtatadi.

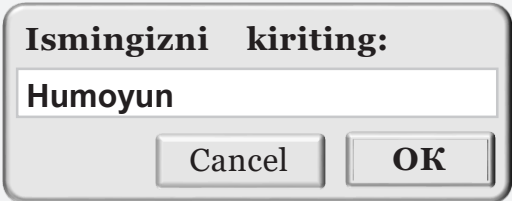
Misol:	<code>alert("Muhim xabar!")</code>
Natija:	

2.1.2. So‘rov (prompt)

So‘rov buyrug‘i ham xuddi ogohlantirish singari ishlaydi, faqat unda keltirilgan qiymatni foydalanuvchi ham o‘zgartira olish imkoniyati mavjud. Ya‘ni biror narsaning odatiy qiymatini berib, uni o‘z holicha qoldirish yoki muqobiliga almashtirish tanlovi veb-sahifa foydalanuvchisining hukmiga havola etiladi. Sintaksisi:

`prompt(xabar, qiymat)`

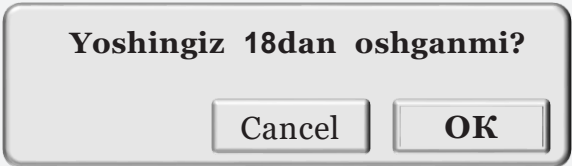
Bunda «OK» (tasdiqlash) tugmasi (yoki «ENTER») bosilgandagina darchadagi qiymat o‘zlashtiriladi. Agar «Cancel» (bekor qilish) tugmasi (yoki «ESC») bosilsa, buyruq e‘tiborsiz qoldiriladi.

Misol:	<code>prompt("Ismingizni kiriting", "Humoyun")</code>
Natija:	

Kiritilgan qiymatlarni qanday o'zlashtirilib, ularning ustida amallar bajarish keyingi mavzularda batafsil, misollar bilan keng yoritilgan.

2.1.3. Tasdiqlash (confirm)

Tasdiqlash buyrug'i aynan ogohlantirish singari ishlaydi. Bunda foydalanuvchiga nafaqat xabar beriladi, balki undan oynada namoyon etilgan jumлага munosabati so'raladi. Agar fikrga qo'shilsa tasdiqlash, aks holda bekor etish tugmasi tanlanadi. Dastur kodi bosilgan tugmaga qarab «rost» yoki «yolg'on» ma'nolaridan birini bildiruvchi mantiqiy qiymatni qabul qiladi.

Misol:	confirm("Yoshingiz 18dan oshganmi?")
Natija:	

2.1.4. Konsol yozuvi

Yuqorida keltirilgan uchta usuldan tashqari foydalanuvchiga ko'rinishi uchun konsol [1.2.] orqali ham ma'lumot chiqarish mumkin. Ammo oddiy foydalanuvchilar, odatda, bunday yozuvlarga ahamiyat bermaydi. Aslida konsoldagi chiqarilgan yozuv orqali muloqot

ham oʻrnatilmaydi. Bu asosan, dasturlovchilar uchundir, chunki yozilgan kodlarning toʻgʻri ishlashini tekshirish jarayonida bunday yozuvlar keng qoʻllaniladi. Unda qiymatlarni batafsil yoyib koʻrish imkoniyati ham mavjud.

Konsolga qiymatlarni chiqarish «**console.log**» va «**console.dir**» buyruqlari yordamida amalga oshiriladi. Ular bilan «console» obyektiga bagʻishlangan mavzuda [3.4.3.] batafsil tanishib oʻtamiz.

2.2. Javascriptni HTMLga bogʻlash

Ushbu kitob JavaScriptga bagʻishlangani bois, avvalo uning kodlari qanday tatbiq etilishi bilan tanishib oʻtish lozim. Keyingi mavzularda dasturlash boʻyicha umumiy tushunchalar JavaScript misolida yoritib boriladi. Keltirilayotgan nazariya va amaliy misollarning toʻlaqonli namoyon boʻlishi uchun yuqorida koʻrib oʻtilgan — dasturlovchiga moʻljallangan muhit yetarli emas [1.2.].

HTML — joylashtirish tili, JavaScript — esa dasturlash tili. Bir-biriga chambarchas bogʻliq ushbu tillarning qoʻllanilish chegaralarini aniqlab olish zarur.

JavaScript ham HTMLga xuddi CSS singari bogʻlanadi, yaʼni maxsus teg ichida dasturlash kodini yozish orqali yoki alohida fayl sifatida eʼlon qilingan kodni biriktirish ham mumkin.

Teg ichida kod yozish uchun maxsus **SCRIPT** tegi ishlatiladi. Uning quyidagi «alamatlar»i mavjud:

▪ **src** — dasturlash kodlari yozilgan fayl manzili ko'rsatiladi. Aslida alohida faylda emas, ushbu tegning ichida kod yozish ham mumkin, lekin HTML va JavaScript kodlar chalkashib ketmasligi uchun bunday yozish tavsiya etilmaydi. Doim maxsus faylni shu alomat orqali ko'rsatish ma'qul hisoblanadi.

▪ **async** — birgalikda bajarishni ta'minlaydi. Odatda, HTML sahifa yuklanib bo'lgandan keyingina unda yozilgan JavaScript kodlar (mavjud bo'lsa) ishga tushadi. Zarurmas qiymatga ega ushbu alomat esa, to'liq yuklanishni kutmay, o'qilgan kodlarni darhol amalga oshirishi lozimligini ko'rsatadi. Bunda dasturlash kodi bajarilishi bilan birga sahifaning qurilishi ham davom etadi. Odatda, ushbu alomat o'chirilgan bo'ladi.

▪ **defer** — HTML sahifa to'liq yuklanib bo'lgandan keyingina «scr»da ko'rsatilgan fayldagi kodlar ishlashni boshlaydi. Bu juda nozik alomat hisoblanadi. Yetarli tajribaga ega bo'lmagan dasturlovchilar ushbu alomat kodni sahifa yuklanib bo'lgandan keyin bajarilishini ta'minlaydi deb o'ylab xatoga yo'l qo'yishadi. Agar kod alohida faylda yozilib, «scr»da ko'rsatilishi o'rniga, tegning ichida keltirilgan bo'lsa, albatta kutilgan natijani bermaydi. Shuni inobatga olish lozimki, HTML sahifa hali to'la yuklanib bo'lmasdan «scr»ni ko'rishi bilan undagi faylni ham biryo'la ko'chirib olishni boshlaydi. Agar ko'rsatilgan JavaScript fayl oldin yuklanib olsa, undagi kodlar darhol ishga tushadi va HTML sahifada keltirilgan elementlar ustida amal bajarish lozim bo'lsa,

hali yuklanib bo'lmagani uchun «topilmadi» degan xato ma'lumot chiqaradi. Dastur tuzish mobaynidagi sinovda JavaScript fayl doim (ehtimol hajmi kattaligi uchun) keyin yuklanadigan bo'lsa, katta xatolar o'tkazib yuborilishi mumkin. Haqiqiy foydalanuvchi ishlayotgan paytda JavaScript fayl «kesh»ga saqlanganligi uchun, tez yuklanib, keyinchalik xato oshkor bo'lishi tayin. Odatda, ushbu alomat o'chirilgan bo'ladi, biz uni doim qo'llashni tavsiya etamiz. Chunki dastavval HTML sahifa yuklanib o'z elementlarini namoyish etadi, so'ngra ular ustida dasturiy amallar bajarilishi maqsadga muvofiqdir.

▪ **language** — dasturlash kodlari qaysi tilda yozilganligini ko'rsatadi, u quyidagi qiymatlardan birini qabul qiladi:

◇ **JavaScript**: Odatiy qiymat, brauzer uchun aniqlik kiritish maqsadida talqin raqamini ham qo'shib qo'yish mumkin. Aslida brauzer (o'zining talqiniga mos) JavaScriptning eng oxirgi talqini bo'yicha kodlarning bajarilishini ta'minlaydi.

◇ **JScript**: «Microsoft» kompaniyasi tomonidan taqdim etilgan boshqacharoq yondashuvdagi dasturlash tili.

◇ **VBS** yoki **VBScript**: Microsoft kompaniyasining «Visual Basic» asosidagi Internet Explorer uchun chiqargan dasturlash tili.

▪ **type** — bajariladigan faylning tavsif-turi. HTML5 tilida uni keltirish shart emas, avvalgi talqinlarida esa «text/javascript», «text/vbscript» singari ko'rsatilar edi.

Quyida JavaScript kodlari alohida faylda yozilganda uni HTMLda qanday yuklanishiga misol keltiramiz:

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title> «JavaScript»ni o'rganamiz.</title>
    <script defer src="value.js"></script>
  </head>
  <body>
    . . . . .
  </body>
</html>
```

Ushbu mavzuni yoritish davomida JavaScript va HTML kodlari qanday tarzda bir-biridan ajratilishini aniq belgilab oldik. Albatta, ularni imkon qadar boshqa-boshqa faylda alohida yozish mukammal dasturlash talablariga mos keladi. Shu bilan birga, bu tarzda yozish amaliyotda ko'pchilik birgalikda bitta loyiha ustida ishlanayotgan vaziyatlarda vazifalarni aniq bo'lish olish jarayonida ham juda qo'l keladi. Hozircha HTML teglarining ichida JavaScript kodi qanday o'rin olishi mumkinligiga sodda misol keltiramiz:







```
<script> alert("Sahifa yuklanmoqda"); </script>
```

Agarda HTML va JavaScript kodlari alohida-alohida faylda yozish talabiga bo'ysunmaslikka to'g'ri kelsa, bit-ta faylda umumlashtirilganda, «**script**» tegini «HTML» kodlarining ixtiyoriy qismiga joylashtirish mumkin, ammo uni faqat «head» tegi ichida keltirish tavsiya etiladi.

yat sifatida funksiyani o'zlashtirish yo'li orqali (Har ikkisi ham quyidagi misolda keltirilgan).

Tovushli faylni ijro etish uchun faqat muhim vositalarni qo'llagan holda muxtasar misol keltiriladi. Unda ba'zi muhim jihatlarga e'tibor qaratish lozim (Lavha ham kerakli o'zgartirishlar bilan shu asnoda qo'llanilishi mumkin).

Sahifa ilk yuklanganda tovush fayliga hali murojaat bo'lmaganligi sababli, uning hatto mavjudligi tekshirilmaydi. Shuning uchun JavaScript yordamida, avvalo, faylning ko'rsatilgan manzilda borligini va uning ijroga yaroqliligini aniqlash lozim bo'ladi.

Ikkinchi navbatda, bosiladigan tugma ijro holatiga mosligi ta'minlanishi lozim. Hali fayl yuklanib tekshiruvdan o'tmagan holati () , yuklanayotgan vaziyati () , ijroga tayyorligi () , to'xtatib turilganligi () , ijro davom etayotganligi () hamda yaroqsizligi () bir-biridan yaqqol ajragan holda ayni bitta tugma nishonchalari sifatida ko'rsatiladi.

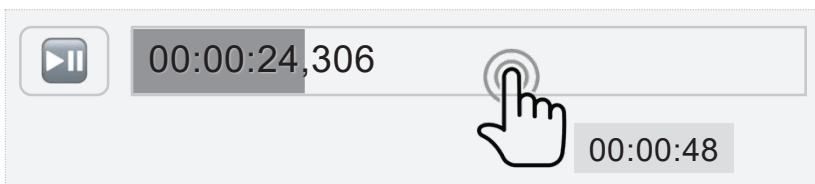
Tugma yonidagi (div) maydonchada ayni paytda qaysi joyi ijro etilayotganini ko'rsatish bilan kerakli o'ringa o'tkazish imkoniyati ham yaratiladi.

Yana bir muhim jihat, sichqoncha bir marta bosilganda boshqa amal, ikki marta bosilganda boshqa amal bajarilishi uchun uddaburonlik bilan yo'l tutishga to'g'ri keladi. Aslida juft bosilganda u ikki marta alohida-alohida bosilgandek hisoblanadi. Faqat oraliq vaqt o'ta qisqa bo'lsa, juft bosilgan deb inobatga olish lozim. Tugma ustida sichqoncha juft bosilsa, butkul to'xtatish

vazifasini bajarishi uchun ijrochining vaziyatini o'zgartirish bilan birga, bir zum «yakun» (□) nishonchasini ko'rsatib, so'ng boshidan boshlashni bildiruvchiga (▶) almashtiramiz.

Albatta, ijrochi ayni paytdagi joyni soniyalarda qaytaradi. Lekin foydalanuvchiga tushunarli bo'lishi uchun uni vaqt sifatida ko'rsatish kerak. Buni amalga oshiradigan qo'shimcha funksiya yozamiz. U kerak bo'lsa, millisoniyani ham qaytaradi.

Yuqoridagi talablar bo'yicha hosil qilingan shamoilda 24-soniyada to'xtatib qo'yilgan tovush ijrochisini 48-soniyaga o'tkazish lahzasi quyidagicha namoyon bo'ladi:



Avval HTML kodini keltiramiz, u tugma va maydon uchun CSSni ham o'z ichiga oladi:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title> Mukammal dasturlash. 5-mashq. </title>
    <style type="text/css">
      #tugma { // maxsus nishonchali tugma uchun
        width: 32px;
        float: left;
        margin-right: 6px;
      }
    </style>
  </head>
  <body>
    <div id="tugma">
      <img alt="Play/Pause button" data-bbox="168 438 218 472" />
    </div>
  </body>
</html>
```

```

#maydon { // ijro foizini aniqlovchi to'rtburchak
  width: 1px; // dastlabki foiz sifatida
  height: 17px;
  float: left;
  background: gray;
  font-size: 12px;
  line-height: 18px; // ichki yozuv balandligi
}
#maydon:before { // maydon chegarasi
  content: '';
  position: absolute;
  border: 1px solid lightgray;
  height: 16px;
  width: 200px; // maydon eni o'lchami
  cursor: pointer;
}
</style>
<script src="audio.js"></script>
</head>
<body>
  <div style="width: 240px;">
    <button id="tugma">🔊</button> // ilk holati
    <div id="maydon"></div>
  </div>
  <audio id="ijrochi">
    <source src='https://yurt.uz/vatan.mp3' />
  </audio>
</body>
<html>

```

Ushbu HTML sahifada biriktirilgan «audio.js» nomli JavaScript faylning tarkibi quyidagicha:

```

var ijroEt = { // oddiy obyekt sifatida yaratildi
  div: 'maydon', // HTMLdagi «maydon» (div)ning idsi
  eni: 200, // «maydon»ning eni (ijro foizi uchun)
  yuklash: function() {
    ijroEt.tugma.innerHTML = '🔄';
    ijroEt.holati = setInterval(function() {
      if (ijroEt.ijrochi.readyState != 4) {
        if (ijroEt.ijrochi.readyState == 0) {
          ijroEt.tugma.innerHTML = '🔄';
          alert('Fayl yuklanishda xatolik!');
          clearInterval(ijroEt.holati);
          return false;
        }
      } else { // fayl ijro uchun tayyor
        ijroEt.tugma.innerHTML = '▶';
        clearInterval(ijroEt.holati);
      }
    },
    40);
  },
  vaqti: function(timeInSeconds, millisoniya) {
    let nolla = function(num, size) {
      return ('000' + num).slice(size * -1);
    }, // oldidan nollar bilan to'ldiruvchi funksiya
    time = parseFloat(timeInSeconds).toFixed(3),
    hours = Math.floor(time / 60 / 60),
    minutes = Math.floor(time / 60) % 60,
    seconds = Math.floor(time - minutes * 60),
    mSeconds = time.slice(-3);
    time = nolla(hours, 2) + ':' + nolla(minutes, 2);
    time += ':' + nolla(seconds, 2);
    if (millisoniya) time += ',' + nolla(mSeconds, 3);
    return time; // soniyani vaqt ko'rishida qaytaradi
  },

```

```

foizi: function() {
  ijroEt.holati = setInterval(function() {
    let m = document.getElementById(ijroEt.div),
        hozir = ijroEt.ijrochi.currentTime,
        hamma = ijroEt.ijrochi.duration,
        f = hozir / hamma * ijroEt.eni;
    if (hozir >= hamma) { //yakunlangan bo'lsa
      f = ijroEt.eni + 1;
      ijroEt.tugma.innerHTML = '▶';
    }
    m.style.width = parseInt(f) + 'px';
    if (!ijroEt.ijrochi.paused) // vaqtni ko'rsatadi
      m.innerHTML = ijroEt.vaqti(hozir, true);
  }, 500); // har yarim soniyada foizi yangilanib turadi
},

bosdi: function(ijrochi, maydon) { //bitta bosganda
  if (ijroEt.ijrochi.paused) { //ijro etmoqda
    ijroEt.ijrochi.play();
    ijroEt.tugma.innerHTML = '⏮';
    ijroEt.foizi();
  } else { //ijro to'xtatib turildi
    ijroEt.ijrochi.pause();
    ijroEt.tugma.innerHTML = '▶⏮';
    clearInterval(ijroEt.holati);
  }
}
};

window.onload = function() { //sahifa yuklanganda
  ijroEt.ijrochi = document.getElementById("ijrochi");
  ijroEt.tugma = document.getElementById("tugma");
  ijroEt.yuklash(); // sahifa yuklanishi bilan tekshiradi

```

```

ijroEt.tugma.addEventListener("click", function() {
  // aynan 1 marta bosgan bo'lsa, 0.3 soniyadan so'ng ishlaydi
  clearTimeout(ijroEt.bos1);
  ijroEt.bos1 = setTimeout(ijroEt.bosdi, 300);
});

ijroEt.tugma.addEventListener("dblclick",function(){
  clearTimeout(ijroEt.bos1);
  // juft chertilsa 0.7 soniya  ko'rinib, so'ng  ga o'zgaradi
  ijroEt.tugma.innerHTML = '';
  setTimeout("ijroEt.tugma.innerHTML = ''", 700);
  ijroEt.ijrochi.pause();
  ijroEt.ijrochi.currentTime = 0;
});
ijroEt.ijrochi.onended = function() {
  let m = document.getElementById(ijroEt.div);
  m.innerHTML = ' Ijro yakunlandi.';
};
ijroEt.maydon = document.getElementById(ijroEt.div);
ijroEt.maydon.addEventListener("click",function(e){
  let p = e.x - this.offsetLeft;
  if (p < 0) p = 0;
  if (p > ijroEt.eni) p = ijroEt.eni;
  p = ijroEt.ijrochi.duration * p / ijroEt.eni;
  // bosilgan joyga mutanosib ravishda ijroni o'tkazish
  ijroEt.ijrochi.currentTime = p;
});
// sichqoncha ko'rsatkichi yoniga mos vaqtni ko'rsatish
ijroEt.maydon.onmouseover = function(e) {
  let p = e.x - this.offsetLeft,
      v = ijroEt.ijrochi.duration * p / ijroEt.eni;
  this.setAttribute("title", ijroEt.vaqti(v));
};
};

```

7.4. Matnni o‘qitish

Hozirda barcha ommabop brauzerlarda matnni o‘qib berish imkoniyati mavjud. Afsuski, hozircha o‘zbek tilini o‘qitish imkoniyati kiritilmagan. Lekin jahon tillaridan bizga ko‘proq tanish bo‘lgan rus va ingliz tillarida yozilgan matnlarni o‘qitish vositalaridan bimalol foydalanishimiz mumkin.

Amaliy mashq yordamida o‘qitish qanday amalga oshirilishi bilan tanishib boramiz. Avvalo, ijrochi faoliyatini boshqaruvchi tugma hosil qilamiz. Unda ijroni boshlash, to‘xtatib turish va yakunlashni bildiruvchi belgilar mos ravishda almashib turadi. Keyin o‘qitish uchun tizimda mavjud ovozlarni tanlash imkoniyatini beruvchi element yasaymiz. So‘ngra o‘qiladigan matn kiritish uchun darcha qo‘yamiz. Ushbu imkoniyatlarni quyidagi HTML kod orqali interfeysini yaratamiz:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Mukammal dasturlash. 6-mashq.</title>
    <script src="md.js"></script>
  </head>
  <body>
    <input id="ijro" type="button">
    <select id="ovozlar" style="display: none;">
      <!-- Ro'yxat yuklanguncha ko'rinmay turadi -->
    </select>
    <br>
```

```

<textarea>
    Это не обычная книга, это бестселлер.
</textarea>
</body>
<html>

```

Shuni inobatga olish kerakki, «speechSynthesis» sahifa yuklangandan keyin biroz o‘tib ovozlarni ro‘yxatida kerakli qiymatlarni beradi. Shuning uchun ilk «ijro» tugmasini bosganimizdan so‘ng ro‘yxat mavjud bo‘lmasa, hosil qilish funksiyasi alohida yozilgan. Quyida «md.js» faylidagi JavaScript kod qanday bayon etilganligi kerakli izohlar bilan keltiriladi:

```

class Mutolaa {
  constructor(tugmaID, matnID, tanlaID) {
    this.tugma = document.getElementById(tugmaID);
    this.matni = document.getElementById(matnID);
    this.tanla = document.getElementById(tanlaID);
    this.bosdi0 = '▶';
    this.bosdi1 = '⏮';
    this.bosdi2 = '⏭';
    this.tovush = window.speechSynthesis;
    this.tillar = [];
    this.tugma.value = this.bosdi0;
    let f = "Mutolaa.tinglash()";
    this.tugma.setAttribute('onclick', f);
    self = this;
  }
  // HTMLdagi tayinlov darchasida mavjud ovozlardan ...
  // ... faqat rus tiliga oidlarini (tizimda odatiy bo‘lgan ...
  // ... ovoz tanlangan holda) yuklab beruvchi funksiya:
  ovozTanlash() {
    var n, r, sel = self.tanla,
        ovoz = window.speechSynthesis.getVoices();

```



```

self.tillar = ovoz;
if (sel.style.display == 'none') {
    sel.style.display = 'inline';
    for (n = 0; n < ovoz.length; n++)
        if (ovoz[n].lang.indexOf('ru') >= 0) {
            var l = document.createElement('option');
            l.textContent = ovoz[n].name +
                ' (' + ovoz[n].lang + ')';
            if (ovoz[n].default) {
                l.setAttribute('selected', 'selected');
                r = ovoz[n].name;
            }
            l.setAttribute('data-lang', ovoz[n].lang);
            l.setAttribute('data-name', ovoz[n].name);
            sel.appendChild(l);
        }
    }
return r;
}
static tinglash() {
    var i, o, t, matn, text, sel = self.tanla;
    if (sel && sel.selectedOptions[0]) {
        sel = sel.selectedOptions[0];
        sel = sel.getAttribute('data-name');
    } else sel = self.ovozTanlash();
    if (self.tugma.value == self.bosdi0) { // ▶
        self.tovush.cancel();
        text = self.matni.value;
        matn = new SpeechSynthesisUtterance(text);
        matn.onend = function() {
            // O'qishdan to'xtaganda tugma ▶ ga o'zgaradi
            self.tugma.value = self.bosdi0;
        };
        // Tanlangan ovozda ijroni ta'minlash
        for (i = 0; i < self.tillar.length ; i++)

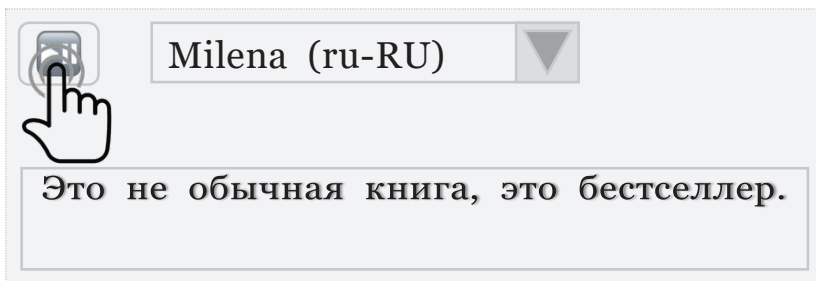
```

```

        if (self.tillar[i].name === sel)
            matn.voice = self.tillar[i];
        self.tovush.speak(matn);
        self.tugma.value = self.bosdi1;
    } else if (self.tugma.value == self.bosdi1) { // ⏸
        self.tovush.pause(matn);
        self.tugma.value = self.bosdi2;
    } else { // ▶
        self.tovush.resume(matn);
        self.tugma.value = self.bosdi1;
    }
}
}
}
// Sahifa yuklanishi bilan «Mutolaa» sinfi ishga tushadi
document.addEventListener('DOMContentLoaded',
    function() { new Mutolaa("ijro", "matn", "ovozlar") }
);
// Sahifa yopilganda yoki qayta yuklanganda ...
// ... o'qish davom etavermasligi uchun uni bekor qiladi.
window.onbeforeunload = function (e) {
    var s = window.speechSynthesis;
    if (s.paused || s.spoken) s.cancel();
};

```

Brauzerda quyidagicha interfeys namoyon bo'ladi:



7.5. Qat'iy tartibdan foydalanish

Taraqqiyot o'z-o'zidan ba'zi mavjud unsurlar iste'moldan chiqishiga yoki o'zgartirilishiga sabab bo'ladi. Internet ibtidosida brauzer faoliyatini takomillashtirish uchun muhim vosita sifatida yaratilgan, umumjahon tarmog'ining taraqqiyotida asosiy sababchilardan biri bo'lgan, to hozirgi kungacha ommaviy qo'llanilib, mukammallashtirilib kelinayotgan ilk dasturlash tili – JavaScriptning asoslari ham bundan mustasno emas.

7.5.1. Qat'iy tartib qo'llanilishi

Qat'iy tartib «ECMAScript5» talqinidan boshlab joriy etilgan. Takomillashtirish mobaynida ba'zi funksiyalar o'rniga muqobili qo'llanish tavsiya etilsa yoki unga murojaat shakli qisman o'zgartirilgan bo'lsa, avval tuzilgan dasturlar o'z vazifasini bajarishdan to'xtab qolmasligi uchun, eski kodlarni ham yangi talqinda qo'llanilaverilishi ta'kidlab o'tilgan edi. Ammo yangi texnologiya bejiz kiritilmaydi, kelgusida faqat undagi qonuniyatlardan foydalanish maqsadga muvofiq hisoblanadi. Har bir kodni yangicha qonuniyatlarga mos kelishini tekshirib borish uchun maydonning boshida **«"use strict";»** deb yozish kerak. Biriktiriladigan JavaScript kodli fayl to'liq yangi qonuniyatlar asosida yozilgani aniq bo'lsa, faylning boshida yoziladi. Faqatgina bitta funksiyaning o'zi yangicha tartibda yozilishi lozim bo'lsa, uning ilk satriga ushbu **«"use strict";»**

jumlasi qo‘shiladi. Bu qat’iy tartib yoqilganligini anglatadi va unda eskicha uslubda ishlab turgan ayrim kodlarda xatoliklar ko‘rsatilishi mumkin. Masalan: kodlarga to‘g‘ri qo‘llanilmaganda, xavfsiz hisoblanmagan (global obyektga murojaat vaqtida) amal bajarilganda yoki buzg‘unchi (bemisl cheksiz takrorlayveradigan) funksiyaga murojaatlar bo‘lganda.

Eski yozilgan kodlardan odatdagidek foydalanib, yangi yozilayotgan fayllarda qat’iy tartib o‘rnatilishi tavsiya etiladi. Zero, mavjud qonuniyatlarga aniq rioya etish — dasturiy mahsulotni takomillashtirishda yuzaga kelishi mumkin bo‘lgan muammolarning oldini oladi. Boshqa dasturlash tillarini o‘zlashtirishga ehtiyoj bo‘lsa, unga qiyoslab o‘rganishni soddalashtiradi, chunki JavaScriptdan boshqalari bu qadar «kechirimli» emas.

Qat’iy tartibning odatdagidan farqini yaqqol ko‘rish uchun, misol tariqasida aynan bir xil yozilgan ikki funksiya quyida keltiriladi:

```
function odatiy(omil, omil) {  
    gm = 9.8;  
    return gm * omil; // ikkinchi omil inobatga olinadi  
}  
  
function qattiy (omil, omil) { // yozilishdagi xato  
    "use strict";  
    mg = 9.8; // bajarishdagi xato  
    return mg * omil;  
}
```

Ikkinchi funksiya e‘lon qilingan zahotiy oq ikkita bir xil nomli omil berish mumkin emasligi haqida ogohlantiruvchi xatolik sodir bo‘ladi. Bu esa «yozilishdagi»

(sintaksis) xato hisoblanadi. Ikkinchi o'zgaruvchi nomi ni almashtirish orqali ushbu kamchilik bartaraf etilgan da funksiya muvaffaqiyatli yaratiladi, lekin ishga tushirilgan paytda «bajarishdagi» xato sodir bo'ladi. Sababi, qiymat berishdan avval o'zgaruvchi e'lon qilinmagan.

Yuqoridagi qat'iy tartibli funksiya to'g'ri bo'lishi uchun quyidagi ikki o'zgartirishni kiritish lozim:

function qattiy (omil, omil2) { "use strict"; let mg = 9.8; return mg * omil; }	<i>Omillar nomi har xil</i> <i>O'zgaruvchi e'lon qilindi</i>
odatiy (2, 3)	29.400000000000002
qattiy (2, 3)	19.6

7.5.2. Qat'iy tartib talablari

Qat'iy tartib qo'llash tavsiya etilar ekan, uning odatiy tartibdan farqlarini aniq sanab ko'rsatib o'tish maqsadga muvofiqdir. Quyida e'tibor qilish lozim tafovutlar bayon etiladi:

- **O'zgaruvchini e'lon qilish shart.** Har bir o'zgaruvchini, shu jumladan obyektini, qo'llashdan avval «var», «let» yoki «const» yordamida e'lon qilish talab etiladi. JavaScript aslida o'zgaruvchiga qiymat berib, uni qo'llab, so'ngra keyingi qatorlarda e'lon qilishga ham imkon beradi. Bunday qo'llanishni «yuksalish» (**hoisting**) deb yuritiladi. Qat'iy tartibda bunga yo'l qo'yilmaydi.

▪ **O‘zgaruvchini butkul o‘chirish taqiqlanadi.**

O‘zgaruvchining o‘zini, xususan, obyekt va funksiyani nomi orqali butunlay yo‘qotish mumkin emas. Obyektning ichki elementlari uchun, agar ular maxsus xususiyat sifatida aniqlanmagan bo‘lsa, bu usulni qo‘llashga ruxsat etiladi:

<pre>"use strict"; var j = [1, 12, 88]; delete j; delete j[1]; console.log(j); delete j.length;</pre>	<p><i>Xato!</i> <i>ruxsat etiladi</i> [1, empty, 88] <i>Xato!</i></p>
-------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------

▪ **Funksiya omillarining nomlari takrorlanmasligi kerak.** Bir xil nomda bir necha o‘zgaruvchi berish mantiqsizlik sanaladi. Yuqorida misol keltirilgan.

▪ **Sakkizlik sanoq tizimi bevosita qo‘llanmaydi.** Sakkizlik sanoq tizimida yozilgan sondan yoki maxsus belgidan to‘g‘ridan-to‘g‘ri foydalanish taqiqlanadi:

x = "\072";	:
y = 011;	9

▪ **Obyektning faqat o‘qishga mo‘ljallangan xususiyati o‘zgartirilmaydi.** Obyektning elementiga yozish imkoniyati cheklangan bo‘lsa yoki «get» funksiya sifatida aniqlanganda, unga qiymat berilganda «xato» qaytaradi. Odatiy holatda qiymatni o‘zlashtirmaydi va xatolik yuz berdi deb kod bajarishdan to‘xtamaydi:

var o = { a: 88 }; Object.defineProperty(o, "a", { writable: false });	
o.a = 1110;	<i>xato</i>
var f = { get x() { return 10; } }; f.x = 20;	<i>xato</i>

▪ **Maxsus nomlanishlarni o'zgaruvchi sifatida ishlatib bo'lmaydi.** Hozircha bu «let», «arguments», «eval», «implements», «interface», «static», «package», «public», «private», «protected», «yield», «NaN» kabi keng qo'llaniluvchi ommabop so'zlar uchungina tatbiq qilingan. Kelajakda dasturlovchi bexosdan tizimda mavjud nomlanishlarning o'ziga xos vazifasini o'zgartirib yubormasligi uchun boshqa maxsus kalit so'zlar ham qo'llanilishiga umid qilamiz. Misol:

var eval = 'Nimadir';	<i>xato</i>
var arguments = 'qiymat';	<i>xato</i>
var parseFloat = 99;	99
var parseInt = 10.9;	10.9

▪ **Biriktirilgan holda «with»dan foydalanish taqiqlanadi.** Ichki xossa va tamoyillarini qo'llashda tashqi obyektning nomini qayta-qayta yozavermaslik uchun «with»dan foydalanish imkoni cheklangan:

with (Math) { x = cos(3.1415); y = sin(1); };	<i>xato</i>
-----------------------------------------------	-------------

▪ **Funksiyaning ichki qismida «this» qoʻllanil-
sa, «undefined» qaytaradi.** Odatiy holda, aniq ob-
yekt berilmaganda, funksiya ichidagi «this» kalit soʻzi
global «window»ga koʻrsatgich hisoblanadi. Misol:

```
"use strict";  
function f() { console.log(this); }  
f();
```

undefined

Maxsus zaxiralangan kalit soʻzlar

?	107, 110, 117
=>	170
array	42
assert	147
async	130
auto	186
await	130
base64	181
behavior	186
boolean	35
break	118, 123
bubbling	191
case	118
catch	127, 132
closure	147
const	29
constructor	162, 171
continue	123
Date	84
debugger	136
decodeURI	139
default	118
delete	58
document	158, 212
DOM	212
DOMRec	218
DOMString	217
Element	212, 213
encodeURIComponent	139

eval	68, 139
extends	173
false	35
fetch	242
finally.....	127
float	32
for	120
function	37, 40
get	172
hoisting.....	289
if	83, 117
in.....	55, 121
Infinity	32
instanceof.....	46, 55
instant	186
int	32
integer	32
label.....	124
left	186
let	28, 39
location.....	197
NaN.....	32
new.....	45
node	228
number	31
object	41
Promise	130
prototype	93, 164
return	37
set.....	172
setTimeout	131
smooth	186

static	171
string	30
super	173
switch	118
then	132
this.....	155, 161
throw.....	128
top	186
true.....	35
try	127
typeof.....	43, 58
use strict.....	287
var.....	28, 39
void.....	58
while	122
window.....	59, 175
yield.....	50, 51

Foydalanilgan adabiyotlar

1. «**Mukammal dasturlash. 1-kitob: HTML va CSS**», Javlon Abdullo. «*Akademanshr*» nashriyoti, 2021.
2. «**Axborot-kommunikatsiya texnologiyalari izohli lug‘ati**» Amirov D.M. Vb 2-nashr. *Toshkent*, 2010.
3. «**Learn JavaScript and Ajax with W3Schools**», Hege Refsnes. Vb, «*Wiley*», 2010.
4. «**Professional JavaScript for Web Developers**», Nicholas Zakas. «*Wrox*», 2005.
5. «**Creating Applications with Mozilla**», David Boswell. Vb, «*O'Reilly Media*», 2002.
6. «**You Don't Know JS: Scope & Closures**», Kyle Simpson. «*O'Reilly Media*», 2014.
7. «**Eloquent JavaScript, A Modern Introduction to Programming**», Marijn Haverbeke. 3-nashr, «*No Starch Press*», 2018.
8. «**Speaking JavaScript: An In-Depth Guide for Programmers**», Axel Rauschmayer. «*O'Reilly Media*», 2014.
9. «**JavaScript and JQuery: Interactive Front-End Web Development**», Jon Duckett. «*Wiley*», 2014.
10. «**The Principles of Object-Oriented JavaScript**», Nicholas Zakas. «*No Starch Press*», 2014.
11. «**JavaScript for impatient programmers**», Axel Rauschmayer. Fran Caye, 2019.

12. **«Exploring ES6: Upgrade to the Next Version of JavaScript»**, Axel Rauschmayer. *«Leanpub»*, 2015.
13. **«Programming JavaScript Applications: Robust Web Architecture with Node, HTML5 & Modern JS Libraries»**, Eric Elliott. *«O'Reilly Media»*, 2014.
14. **«Pro JavaScript Techniques»**, John Resig v. b. 2-nashr, *«APress»*, 2016.
15. **«JavaScript для FrontEnd-разработчиков. Написание. Тестирование. Развертывание»**, Кириченко А.В., *«Наука и техника»*, 2020.

MUNDARIJA

1-BO‘LIM. KIRISH

- 1.1. Veb-dasturlash tili: yaratilish tarixiga bir nazar 3
- 1.2. Qo‘llanish sohasi..... 6

2-BO‘LIM. DASTURLASH ASOSLARI

- 2.1. Xabar ko‘rsatish..... 10
 - 2.1.1. Ogohlantirish (alert) 10
 - 2.1.2. So‘rov (prompt)..... 11
 - 2.1.3. Tasdiqlash (confirm)..... 12
 - 2.1.4. Konsol yozuvi 12
- 2.2. Javascriptni HTMLga bog‘lash..... 13
- 2.3. Dasturlash tili tushunchalari..... 18
 - 2.3.1. Obyekt 18
 - 2.3.2. Kod yozish qoidalari..... 20
 - 2.3.3. Had va ajratuvchilar 23
 - 2.3.4. O‘zgaruvchi va o‘zgarmaslar..... 25
- 2.4. Turlar 30
 - 2.4.1. Satr 30
 - 2.4.2. Son..... 31
 - 2.4.3. Maxsus qiymatlar 34
 - 2.4.4. Mantiqiy tur 35
 - 2.4.5. Funksiya 36
 - 2.4.6. Obyekt 41
 - 2.4.7. Turni aniqlash 43
- 2.5. Amallar 48

2.6. Biriktirilgan obyektlar	59
2.6.1. Son (Number)	60
2.6.2. Satr (String)	66
2.6.3. Jadval (Array)	74
2.6.4. To'plam (Set)	81
2.6.5. Mantiqiy tur (Boolean)	83
2.6.6. Sana (Date)	84
2.6.7. Matematika (Math).....	87
2.6.8. Obyekt (Object).....	92
2.6.9. Xarita (Map)	99
2.6.10. JSON	102
2.7. Muntazam ifodalar	107

3-BO'LIM. OPERATORLAR BILAN ISHLASH

3.1. Tanlash operatorlari	116
3.1.1. if	117
3.1.2. ? ... :	117
3.1.3. switch	118
3.2. Takrorlash operatorlari	120
3.2.1. for	120
3.2.2. for ... in	121
3.2.3. for ... of	121
3.2.4. while	122
3.2.5. do ... while	122
3.2.6. Takrorlanishni tark etish.....	123
3.3. Talab operatorlari.....	126
3.3.1. try	126
3.3.2. throw	128
3.3.3. «async/await»	129

3.4. Kod tahlili (debug).....	134
3.4.1. Qadam-baqadam tekshirish.....	134
3.4.2. Error obyektı	138
3.4.3. console obyektı.....	140
3.4.4. BDD usulida avtomatik sinash.....	146

4-BO‘LIM. OBYEKTGA YO‘NALTIRILGAN DASTURLASH

4.1. OYD haqida tushuncha.....	152
4.2. Obyekt yaratish usullari.....	153
4.3. Tarkiblash (abstraction).....	156
4.4. Qobiqlash (encapsulation)	160
4.5. Meroslash (inheritance)	162
4.6. Turlichalash (polymorphism).....	165
4.7. OYDning «class ... extends» shakli.....	168

5-BO‘LIM. VEB-SAHIFA OBYEKT LARI

5.1. window obyektı.....	175
5.1.1. window xususiyatlari.....	175
5.1.2. window uslublari.....	181
5.1.3. Hodisalar.....	186
5.1.4. «location» obyektı.....	197
5.1.5. «history» obyektı.....	199
5.1.6. «navigator» obyektı	203
5.1.7. «performance» obyektı.....	208
5.2. «document» obyektı.....	212
5.2.1. «Element» obyektı	213

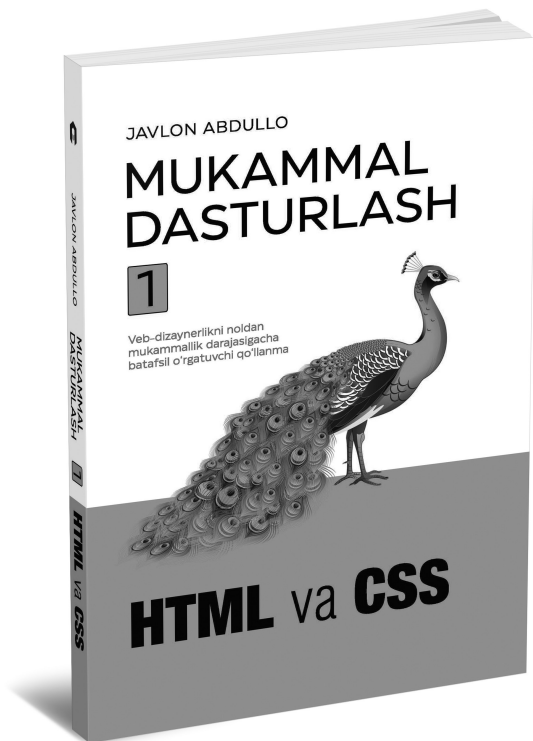
5.2.2. «document»ning o'ziga xos xususiyatlari	223
5.2.3. «document»ning o'ziga xos uslublari.....	226
5.2.4. document.cookie	233
5.3. Darchalar qiymatlari	237
5.3.1. Qiymatni olish va o'zgartirish.....	237
5.3.2. AJAX orqali ma'lumot almashish.....	240

6-BO'LIM. SHAKLLARNI HARAKATLANTIRISH

6.1. «canvas» obykti	249
6.2. Suriluvchan oynalar.....	259
6.3. Elementlarni harakatlantirish.....	264

7-BO'LIM. AMALIY MASHQLAR

7.1. Joylashuvni aniqlash	268
7.2. Sahifalararo ma'lumot almashish.....	271
7.3. Ijrochilar bilan ishlash.....	273
7.4. Matnni o'qitish	283
7.5. Qat'iy tartibdan foydalanish	287
7.5.1. Qat'iy tartib qo'llanilishi.....	287
7.5.2. Qat'iy tartib talablari	289
Maxsus zaxiralangan kalit so'zlar.....	293
Foydalanilgan adabiyotlar	296



Ushbu kitob
«Mukammal dasturlash. 1-kitob: HTML va CSS»ning
mantiqiy davomi hisoblanadi

Ilmiy-ommabop nashr

Javlon ABDULLO

MUKAMMAL DASTURLASH

2-kitob

JavaScript

Muharrir: Abdulla SHAROPOV

Badiiy muharrir: Bahridin BOZOROV

Texnik muharrir: Dilshod NAZAROV

Sahifalovchi: Inomjon O'SAROV

Musahhih: Mahfuza IMOMOVA

Nashriyot litsenziyasi: AI №134, 27.04.2009
Terishga berildi: 08.10.2020-y.
Bosishga ruxsat etildi: 09.02.2022-y.
Ofset qog'ozi. Qog'oz bichimi: 60x84¹/₁₆,
Cambria garniturası. Ofset bosma.
Hisob-nashriyot t.: 15,1. Shartli b.t.: 17,6.
Adadi: 3000 nusxa.
Buyurtma № 6

«Akademnashr» nashriyotida tayyorlandi va chop etildi.
100156, Toshkent shahri Chilonzor tumani 20^A-mavze 42-uy.

Tel.: (+99871) 216-87-81
e-mail: info@akademnashr.uz
web: www.akademnashr.uz