

JAVLON ABDULLO

MUKAMMAL DASTURLASH

1

Veb-dizaynerlikni noldan
mukammallik darajasigacha
batafsil o'rgatuvchi qo'llanma



HTML va CSS

JAVLON ABDULLO

MUKAMMAL DASTURLASH

1-KITOB:

HTML va **CSS**

Toshkent
«Akademnashr»
2021

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

A 15 Javlon Abdullo

Mukammal dasturlash. 1-kitob. HTML va CSS [Matn]; Ilmiy-ommabop / Javlon Abdullo. – Toshkent: «Akademnashr», 2021. – 288 b.

ISBN 978-9943-6964-4-0

UO'K: 004.42
KBK: 32.973-018

Ushbu kitob zamonaviy dasturlashni o'rganmoqchi bo'lganlar, dasturlash bilan shug'ullanadiganlar, hatto dasturiy ta'minot yaratish bo'yicha yetuk mutaxassislar uchun ham mo'ljallangan bo'lib, ilm olish va mustahkamlash uchun eng kerakli bilimlarni o'zida jamlagan hamda sodda, tushunarli tilda bayon etilgan, ortiqcha ma'lumotlar o'rniغا o'n yildan ziyod amaliy ishlash jarayonida orttirilgan hayotiy tajribalardan kelib chiqqan holda foydali ko'satmalar samarali tarzda va to'g'ri keltirilgan mukammal qo'llanmalarning ilk qismidir. Shuning o'zi ham unda yoritilgan bilimni o'zlashtirgan kishining yetuk veb dizayner sifatida keng faoliyat yuritishiga yetarli zamin yaratadi.

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

Ushbu kitobni yoki uning sahifalari nusxasini muallifning tegishli ruxsatsiz har qanday ko'rinishda tarqatish qonunga muvofiq taqiqilanadi. Undagi ma'lumotlardan foydalanylinda kitob nomi qayd etilishi shart.

ISBN 978-9943-6964-4-0

© Javlon ABDULLO
«HTML va CSS»
© «Akademnashr», 2021

1-BO'LIM

KIRISH

1.1. Kitob xususida

Hozirda shiddat bilan rivojlanib borayotgan hayotimizni internetsiz tasavvur qila olmaymiz. Ishlab chiqargan mahsulotimizni, ilmiy va adabiy merosimizni, yuksak madaniyatimizni keng ommaga, hatto butun jahonga yoyishimizda global tarmoq bemisl imkoniyatlarga ega.

Internetda ma'lumotlar qanday almashinishini, o'quvchiga qay tarzda namoyon bo'lishini dasturlovchi nazari bilan kuzatish uchun esa, avvalo, HTML va CSS tushunchalaridan xabardor bo'lish lozim. Aslida, ularga dasturlash tillari sifatida qaralmaydi va faqat veb-sahifalarni akslantirishda foydalaniladi, xolos. Saytlar va veb-sahifalarning bejirim, qulay interfeysda hamda foydalanuvchiga tushunarli bo'lishi, shu bilan birga, ommabop brauzerlarda kutilganidek namoyon bo'lishi uchun HTML va CSS imkoniyatlarini puxta o'rganib, ulardan o'z o'rniда to'g'ri foydalanadigan bo'lishimiz kerak. Bu yo'lida ushbu kitob doimiy, samarali hamroh bo'ladi.

Ushbu mukammal qo'llanma dasturlash tilini o'rganishga endi qadam qo'yamoqchi bo'lgan o'quvchiga ham, bir necha yillar mobaynida dasturlash bilan shu g'ullanadiganlar uchun ham, dasturlashni mukamma-

61649

o'zlashtirib olganlar uchun ham mo'ljallangan bo'lib, o'z bilimlarini mustahkamlashlariga ulkan hissa qo'shishiga ishonamiz. Zero, bunda yangi texnologik darajaga ko'tarilgan va oldingilaridan juda katta tafovutga ega HTMLning beshinchi va CSSning uchinchi talqini asos sifatida qaralib, HTML va CSS imkoniyatlari batafsil yoritilgan. Hozirgi kunda qo'l soatidan tortib mobil telefonlar-u planshetlar, kompyuterlar-u maxsus serverlar, hatto televizorlar-u «aqlii texnologiya» nomini olgan barcha uskunalar HTML va CSSni qo'llovchi o'z dasturiy ta'minotlariga ega. Mazkur imkoniyat ularda aks-lantirish mumkin bo'lgan barcha narsalarni namoyon qilish uchun yetarli, ammo texnologiya taraqqiyoti bir joyda to'xtab qolmaydi. Keyinchalik kiritiladigan muhim yangiliklar ushbu kitobda ko'rsatilgan saytda va qo'llanmaning keyingi nashrlarida qo'shib boriladi. Ifoda etilgan kodlar qanday tavsiflangan bo'lsa, barcha omma-bop brauzerlar, ya'ni Mozilla Firefox, Google Chrome, Microsoft Edge, Opera, Safari, Internet Explorer va boshqalarning yuqori talqinlarida hamda mobil brauzerlarda birdek namoyon bo'ladi. Maxsus istisnolar o'z o'rnida bayon etilgan. Katta hajmli va ko'p imkoniyatli saytlar yaratish uchun ham maxsus ko'rsatmalar berib borilgan.

1.2. Qisqacha tarix

1986-yilda standartlashtirish bo'yicha xalqaro tashkilot (ISO) Standard Generalized Markup Language (SGML) – «Standart umumlashtirilgan belgilanishlar tili» nomli standartni qabul qilgan. U ISO 8879 raqami bo'yicha ro'yxatdan o'tgan bo'lib, turli operatsion tizimlar va kiritish qurilmalarida ham qulay o'zlashtiriladigan matn shaklidagi ma'lumotlarni akslantirishga mo'ljallangan. Unda ma'lumotlar qanday ko'rinishda ifodalanishi keltirilgan, xolos, maxsus til sifatida barcha buyruqlar jamlanmagan va ularning aynan qay tarzda namoyon bo'lishi batafsил keltirilmagan. Shuning uchun ham ushbu standart ommalashmay qolgan.

1991-yilda Jenevadagi Yevropa yadro tadqiqotlar tashkiloti (CERN)ning britaniyalik olimi Tim Berners Li ma'lumotlarni to'g'ri sahifalashni bilmaydiganlar uchun ham elektron tarzda hujjatlarni almashishni qulaylashtirish maqsasida maxsus til yaratdi. Unga asos sifatida SGML olingan bo'lib, HTML – Hyper Text Markup Language (*Gipermatnli belgilanishlar tili*) deb nomlandi. Oradan ikki yil o'tib, 1993-yilda qirqta belgilanish (ya'ni «**teg**») asosida HTMLning 1.2-talqini yaratildi va o'sha paytdagi yagona grafik imkoniyatli brauzer – «Mosaic» dasturi yordamida ma'lumotlar akslantirildi. «Mosaic» NCSA – National Center for Supercomputing Applications – «Superkompyuter ilovalar uchun milliy markaz» mahsuloti edi. Aynan shu markazda WWW

(Wold Wide Web) – «Butunjahon o'rgimchak to'ri» ham ishlab chiqilgan.

1994-yilda «Butunjahon Internet tarmog'idagi konsortsium» W3C (World Wide Web Consortium) tashkil etilib, mavjud tilga ulkan o'zgartirishlar kiritilib, 1995-yili HTMLning 2.0 talqini havola etildi. Asosiy o'zgarish ma'lumotlarni formada jamlab serverga yuborish mexanizmi qo'shilganligida bo'ldi. Ammo bu umuman talab doirasida emasdi. Unga murakkab matematik formula-lar, turli shakldagi jadvallar, shu bilan birga, rasm bilan matnni bирgalikda akslantirish imkoniyatlari kiritilishi lozim edi. Biroq bunday imkoniyatlarga ega brauzerning o'zi u paytda mavjud bo'lмаган. Kengayib borayotgan texnologik imkoniyatlarni inobatga olgan holda HTML mualliflari 3-talqinni yaratish mobaynida yangicha yondashish bilan CSS vositasini ham amalga tatbiq etishdi. Ya'ni HTML orqali joylashtiriladigan obyektlarning ketma-ketligi aniqlanadi, xolos, ularning qay tarzda namoyon bo'lishi esa alohida sintaksisga ega bo'lgan CSS yordamida ko'rsatiladi.

1994-yili «Mosaic» brauzeri yaratuvchilari Jeyms Klark boshchiligidagi anchagina mukammal «Netscape» nomli yangi brauzer yaratishdi va ilk bor savdoga chiqarishdi. Unda amaliyatga tatbiq qilingan va bir qancha to'ldirishlar kiritilgan HTMLning 3.2-talqini va CSSning 1-talqini bekam-u ko'st namoyon bo'lardi. Hatto bir necha yillardan keyin ham Microsoft shirkati tomonidan yaratilgan «Internet Explorer» brauzerining 3-talqinida ham «Netscape Navigator»ning 1-talqinidagi bir qancha imkoniyatlar kiritilmagan edi. Insoniyatning

ma'lumot almashishga tashnaligi hamda «Netscape Navigator»ning izlanishlarga o'chligi va yangiliklarni tezkor tatbiq etishi uning 1995-yillarda dunyodagi eng tez rivojlanayotgan shirkat bo'lishiga sababchi bo'lgan. Keyinchalik Microsoft kompaniyasi «W3C»ning homiysiga aylanib, o'z operatsion tizimiga tekin ravishda «Internet Explorer»ni qo'shib berish bilan asta-sekin «Netscape Navigator»ni butkul ortda qoldirdi. Shu boisdan hozirda «Netscape Navigator» eng ommalashgan brauzerlar safida o'rinn olmay qo'yan, uning yaratuvchilari esa takomillashtirish ustidagi faoliyatlarini to'xtatishgan.

1997-yili HTMLning 4-talqini joriy etildi. Unda bir qancha belgilanishlar eskirgan deb topilib, qo'llamaslikka unday boshlandi. Ularning o'rniga CSSning imkoniyatlaridan foydalanish tavsiya etildi. 1999-yili HTML 4.01-talqini standart asosida foydalanishga chiqarildi. So'ngra o'n yildan ziyod vaqt ushbu tilni takomillashtirish butkul to'xtab qoldi. Sababi, yaratuvchilari HTMLning o'rniga XML (*Kengaytirilgan belgilanishlar tili*)dan foydalanishni ma'qul topishdi. Lekin brauzer yaratuvchilari ushbu qarashni ma'qullahsmadi va o'zları talab doirasida HTML va CSS imkoniyatlarını kengaytirib boraverishdi. Natijada bitta qoidani turli brauzerlar uchun qayta-qayta yozishga to'g'ri keladigan bo'lib qoldi. HTML voz kechib bo'lmaydigan til ekanligini anglaganlaridan so'ng ushbu tilga juda jiddiy qaralib, o'zaro kelishmay, yagona standartga qat'iy rioya qilmay tarqoq ishlayotgan eng mashhur shirkatlar (*Microsoft, Opera, Mozilla, IBM, Google, Apple* va

boshqalar) birlashib, yangi HTML5 ustida izlanish olib borishga qaror qilishdi. Hozirda 2013-yil fevralda taqdim etilgan HTMLning 5.0-talqini bo'yicha barcha brauzerlar faoliyat olib bormoqda, unda ovozli va harakatli tasvirlarni hamda o'zgaruvchan vektorli grafik (SVG)larni akslantirishga katta e'tibor qaratilgan. HTML5ni rivojlantirish 2022-yilgacha davom etishi mumkin, lekin oradan bir qancha yil o'tsa ham, uning qonuniyatlariga sezilarli o'zgartirish kiritilmadi. Ushbu kitobda yoritilgan bilimlar HTML5 va CSS3 imkoniyatlarini to'la qamrab oladi.

HTML tilida xatoliklar va tushunmovchiliklar e'tiborsiz qoldiriladi. Bu esa hamma dasturlovchiga o'zi istaganday kod yozish imkoniyatini berish bilan birga o'zi shundoq ham tartibsiz internetning yanada besaranjomlashishiga olib keladi, ya'ni ayni bir ma'lumot har xil brauzerda turli shaklda namoyon bo'ladi, izlovi-chi tizimlarda esa kutilgan xabar topilishiga to'sqinlik vujudga keladi. Bunday muammolarning oldini olish maqsadida 2000-yilda kengaytirilgan HTML (XHTML) tili yaratilgan. Unda ancha qat'iy talablar qo'yilgan bo'lib, xatolik uchragan holda kutilgan natijaga erishilmaydi. Bunday qat'iylik dasturiy ta'minot yordamida kodlarni tahlil qilishga qulaylik yaratadi. Afsuski, HTML5da talablarni qat'iylashtirishga muhim e'tibor qaratilgandan so'ng XHTMLni qo'llash va takomillash-tirish to'xtatildi.

2-BO'LIM

HTMLning ASOSIY TARKIBI

2.1. Asosiy tushunchalar

HTML dasturlash tili bo'lmagani uchun ham uning kodlariga buyruq emas, belgilanish sifatida qaraladi. Har bir belgilanish «<» belgisi bilan boshlanib, «>» belgisi bilan yakunlanadi va u «**teg**» deb nomlanadi.

Ushbu kitobda teg va boshqa kodlarning sintaksisi ort rangi alohida ajrab turuvchi maxsus maydon ichida keltiriladi. Ularni bayon etishda tagiga chizilgan egri yozuvdagi o'zbekcha izoh kelgan bo'lsa, uning o'rniда ifoda etilgan kod keltirilishi inobatga olingan bo'ladi. Agar qiymatlar «|» belgisi bilan ajratilgan holda berilsa, keltirilgan ro'yxatdagidan faqat bittasi qo'llanilishini anglatadi. Agar «[» va «] » qavslari ichida ifoda kelgan bo'lsa, u yozilishi shart bo'lмаган kodligini bildiradi. Yozilishi shart bo'lмаган ifodalar (teg yoki uning alovati) o'zining odatiy qiymatiga ega bo'lib, ixtiyoriy brauzer o'sha qiymat bo'yicha akslantiraveradi.

Sintaksisdan boshqa joylarda – kitobning odatiy yozuvlari ichida «[2.1.]» kabi belgilanish uchrasa, bayon etilgan fikrning to'ldiruvchi ma'lumoti ushbu kitobning 2.1-bo'limida (ya'ni mos raqamlar ostidagi

mavzuda) kengroq yoritilganligini anglatadi. Biror tushunchaning tafsiloti keltirilayotganda «¹» sinragi raqam qo'yilgan bo'lsa, demak, uning to'la izohi sahifaning quyi qismida keltirilganligidan bo'ladi. Biror tushunchani batafsilroq o'rganish kerak bo'lsa, o'sha so'z kitobning oxiridagi atamalar ko'rsatkichidan nechanchi sahifada keng yoritilganligini topib, o'qish mumkin.

Shu o'rinda ushbu kitobda qo'llaniladigan ba'zi tushunchalarga oydinlik kiritib o'taylik:

Brauzer — belgilanishlar tillarida (HTML, XHTML, XML va boshqalar) yozilgan kodlarni mazmuni bo'yicha namoyon etuvchi dasturiy ta'minot (*Chrome*, *Safari*, ...).

Ishorat — hujjatning butunjahon to'ridagi manzilini aniqlovchi yozuv. U ko'p hollarda URL, URN, URI ko'rinishlarida ham nomlanadi. [2.3.]

Element — tegning brauzerde akslanadigan ko'rinishi. Teg «<» va «>» belgilari orasidagi yozuvni ifodalasa, «**element**» deganda yozuv emas, brauzerde namoyon bo'ladigan shakl tushuniladi. Element brauzerde to'g'ri to'rtburchak ko'rinishidagi yoki qatorli yozuv sifatidagi maydonni egallaydi, bu maydon ko'rinvuchchi yoki ko'rinxmas bo'lishi mumkin. [2.5.]

Obyekt — yagona yoki bir nechta elementlar majmuasidan hosil bo'lgan shakl.

Veb-sahifa — brauzerning bitta oynasida ifodalana-digan barcha obyektlar majmuyi. Kelgusida uni oddiy «sahifa» deb ham yuritamiz.

Lotin harflari — «A»dan «Z»gacha va «a»dan «z»ga-cha hamda «_» belgisidan iborat jami 53 ta belgi.

Ajratuvchi belgilar — harfni ifodalamaydi, odatda, so'zlarni bir-biridan ajratish uchun qo'llaniladi. Ular «», «.», « », «(», «)», «+», «-», «/», «*» va hokazo.

Identifikator — elementlarni bir-biridan farqlash uchun qo'llaniladigan nomlanish. Unda faqat lotin harflari, pastki chiziqcha (_) va raqamlardan foydalaniladi, ajratuvchi belgilar ishlatalmaydi. Bundan tashqari, identifikatorning birinchi belgisi raqamdan iborat bo'lmasligi shart. Identifikatorda katta-kichik harflar e'tiborga olinadi hamda uni nomlashda tilda mavjud bo'lgan «zaxiralangan kalit so'zlar», ya'ni buyruq yoki boshqa ma'noni anglatuvchi so'zlar qo'llanmasligi kerak. Faqat HTML tilidagina identifikatorni qo'shtirnoq ichida berish imkoniyati mavjudligi uchun ham ba'zi ajratuvchi belgilardan foydalanish mumkin. Bunday belgilanishlarni «**kelishilgan identifikator**» deb nomlaymiz, unda birinchi harfdan keyin chiziqcha (-) qo'llanilishi mumkin. Ammo biz kelajakda boshqa dasturlash tillarini ham o'zlashtirishimiz zarurligini inobatgan olgan holda identifikatorda umuman ajratuvchi belgilardan foydalanishni tavsiya etmaymiz. HTML dasturlash tili bo'lmagani uchun hech qanday matematik amallar kod ichida yozilgan deb qaralmaydi, aks holda, chiziqcha ayirish amali sifatida qabul qilinishi mumkin bo'lardi.

Yuqorida kelishuvlarga asoslanib tegning mavjud ikki xil sintaksisini keltiramiz:

<teg nomi [teg alomatlari]>

<teg nomi [teg alomatlari]> [teg asosi] </teg nomi>

Yuqorida ta'kidlab o'tganimizday, ayrim veb-sahifalar dasturiy ta'minotlar yordamida o'qilib, undagi ma'lumotlarni o'zlashtirish qulay bo'lishi uchun XHTML standartlari bo'yicha yoziladi. Unda birinchi sintaksis bo'yicha yozilgan teg xato hisoblanadi. Albatta yopuvchi teg nomi ko'rsatilishi kerak, agar u mavjud bo'lmasa, quyidagicha ifodalanadi:

<teg nomi [teg alomatlari] />

XHTMLni rivojlashtirish ustida faoliyat to'xtatilganligi va ushbu kitob aynan HTMLni yoritishga asoslangani uchun biz teglarni HTML5 talablari bo'yicha yozishni tavsiya etamiz.

Shu o'rinda kichik bir misol ko'rib o'taylik. HTMLda satrlarni bo'lувчи teg: **br**. U yopuvchi qismiga ega emas, shunchaki **
** shaklida qo'llaniladi. Demak, yuqoridagi fikrni inobatga olgan holda uni **
** yoki **
** yoxud **
** ko'rinishlarida yozish mumkin. Har uchalasi ham to'g'ri hisoblanadi va bir xil natija beradi. Biz oddiygina **
** shaklida yozishni tavsiya etamiz, bu HTML ning asl sintaksisiga mos kelish bilan birga yozilayotgan kod hajmini qisqartirishga ham yordam beradi. Ammo qo'shimcha fraymworklardan foydalanuvchilar uning talabiga binoan yopuvchi qismi yo'q teglarni « />» tarzida yakunlashga odatlansalar ham zarar qilmaydi.

HTML tilida harflarning katta-kichikligiga e'tibor qaratilmaydi. Yuqoridagi tegning **
**, **
**, **
** va **
** shakllari birdek qabul qilinadi. «Pascal» kabilardan tashqari dasturlash tillarida faqat kichik harflarni

qo'llash talab etilgani uchun dasturlovchilar orasida kichik harflardan foydalanish ommalashgan. Teg nomlarini qanday yozish dasturlovchiga havola, ammo biz ham kichik harflardan foydalanishni maqsadga muvofiq deb bilamiz.

Tegning alomatlari quyidagi ko'rinishda yoziladi:

alomat nomi = qiymat

Qiymat, odatda, qo'shtirnoq ichida yoziladi, unda ajratuvchi belgilari, xususan, bo'shliq (probel) qatnashmagan bo'lsa, qo'shtirnoq yozish shart emas. HTML tilida ham ko'pgina dasturlash tillari singari ikki xil qo'shtirnoq ishlataladi: «"» va «'». Ammo ko'pchilik dasturlovchilarga odat tusiga kirganligi uchun ikkitalik qo'shtirnoqni doim qo'llash tavsiya etiladi. Kitobning keyingi qismlarida sintaksis shu tariqa beriladi, dasturchining o'zi ixtiyoriga qarab istagan qo'shtirnog'ini qo'yishi yoki ulardan umuman foydalanmasligi mumkin.

Matn ichida yoziladigan ba'zi maxsus kodlanishlarni keltirib o'tamiz:

ifodasi	kodi	belgi	izoh
<	<	<	Kichik belgisi
>	>	>	Katta belgisi
"	"	"	Qo'shtirnoq
	'	'	Birtirnoq (opostrof)
‘	‘	'	Chap birtirnoq
’	’	'	O'ng birtirnoq
&	&	&	Ampersand, mantiqiy ko'paytirish ishorasi
 	 		Ajratmas bo'sh joy

ifodasi	kodi	belgi	izoh
	‒	-	Chiziqcha
	‐	-	Qisqa chiziqcha
–	–	-	Qisqa chiziq, «n» harfi kengligida
−		-	Matematik ayirish belgisi
—	—	-	O'rtacha chiziq, eni «m» harfi kengligicha
	―	-	Uzun chiziq
­			Bo'g'in ko'chiruvchi, qator oxirida-gina (bo'g'in ko'chirilsa) ko'rindi

Barcha maxsus kodlanishlarni keltirib o'tmaymiz, qolgan belgilar o'z ko'rinishiga ega va u hujjat ichida qanday yozilsa, kodlama to'g'ri tanlanganda barcha brauzerlar aynan o'zidek akslantiradi. Yuqorida chalkashliklarga sabab bo'luvchi belgilanishlar keltirilgan, agar ularni chop etishga zaruriyat tug'ilsa, jadvalda keltirilgan «ifodasi» yoki «kodi»dan foydalanish tavsiya etiladi.

Chop etiladigan matn katta bo'lgan hollarda ulardagi so'zlarning egallaydigan hajmiga qarab bir necha satrlarga bo'lib akslantiriladi. Bunday hollarda yonma-yon turmasa mazmuni o'zgarib ketadigan so'zlar ketma-ket qatorlarga tushib, jumla mohiyatini buzishi mumkin. Shunday anglashilmovchilikka yo'l qo'ymaslik uchun «ajratmas bo'shliq» ishlatiladi, u maxsus iboralarni anglatuvchi qo'shni so'zlarning turli qatorlarga tushib qolmasligiga xizmat qiladi va bir qatorda yonma-yon kelishi shart bo'lgan so'zlar orasiga qo'yiladi. Brauzerlar esa ushbu so'zlarni ajratilgan holda namoyon etaveradi. Aksincha, juda uzun so'zning ma'lum bir qismini boshqa satrga ko'chirib tushirmsa, sahifaning chiroyi

buziladigan bo'lsa, «­» belgisidan foydalaniladi. Agar brauzer bo'g'in ko'chirishga hojat yo'q deb hisob-lasa, uni so'zdagi harflar orasida ko'rsatmaydi; keyingi qatorga so'zning ajratilgan qismini tushirishga to'g'ri kelsa, qatorning oxirida uni chiziqchaga almashtiradi.

HTML tilida bo'sh joylar, tabulyatsiya va qator yakunini bildiruvchi belgililar nechta va qanday ketma-ketlikda yozilgan bo'lsa ham, yagona bo'sh joy sifatida akslantiriladi. Matn qanday ko'rinishda yozilgan bo'lsa, aynan shunday namoyon bo'lishi uchun **pre** [8.8.] yoki **xmp** [3.3.] tegi ichida keltirish talab qilinadi. Sababi, HTML – belgilanishlar tili, matnning qaysi bo'lagi qanday ko'rinishda chop etilishi kerak bo'lsa, ularni shunga mos teglar ichida yozish lozim.

Barcha tillarda bo'lgani kabi, HTMLda ham kodning brauzer tahlil qilmasligi kerak bo'lgan qismi ajratib qo'yilishi mumkin. Bunday bo'lak **izoh** deb nomlanadi va quyidagi teg ichida yoziladi:

<!-- izoh matni -->

2.2. Ilk bosqich

Veb-sahifa yaratish uchun biz oddiy matn tahrirlovchi dasturdan foydalanamiz. Aslida, turli-tuman HTML yaratuvchi qulay muhitlar serob, ammo biz o'rganuvchi bo'lganligimiz va har bir kodni o'zimiz nima uchun

yozayotganimizni anglab yetishimiz uchun «notepad» ishlatishimiz maqsadga muvofiq. Yozishni osonlashtirish va kodlarni alohida o'ziga xos ranglar bilan farqlab ko'rish uchun «**notepad++**» yoki «**Sublime**»ni qo'llasa ham bo'ladi. Biroq boshlang'ich qadamda HTML kodlarni o'zi yozib beradigan muhitlardan foydalanilmagani ma'qul, sababi, o'zi qo'shib qo'yadigan kodlar nimani anglatishini hali to'liq o'rganib chiqmadik.

Birinchi qadamda matn tahrirlovchi dastur (*sublime, notepad++ yoki boshqa, bundan keyin MTD deb yuri-tamiz*) yordamida yangi fayl yaratamiz. Uni «UTF-8» shaklida kodlamaga o'tkazib olamiz. Sababi, butun dunyoda mavjud belgilarning hammasi unda jamlangan, aynan u veb-sahifalar yaratishga yo'naltirilgan.

Ikkinci qadamda faylga nom berib, kengaytmasini «htm» (yoki «html», *ko'pgina kengaytmalar uchta harfdan iborat bo'lgani uchun hamda HTML5ning yo'riqnomasini inobatga olgan holda «htm»ni tanlash tavsiya etiladi*) deb saqlash kerak. Keyingi qadamlarda bevosita kod yozib borilaveradi.

5-talqin tatbiq etilmasdan avval HTMLda resurslarning yetishmasligi va aniq yagona kelishuvga amal qilinmasligi kod yozishda juda ko'p muzokarali qarashlarni keltirib chiqarar edi. Yozilgan kodni yagona qolipga solish uchun juda murakkab va esda qolishi mushkul sintaksis asosiga hujjat turini ko'rsatish kerak bo'lgan. Agar sahifaning tarkibi keskin o'zgartirilsa, unga mos ravishda hujjat turi ham o'zgartirilishi taqozo etilgan. Hujjat turini aniqlovchi DTD fayl W3C saytidan

olingan. Agarda sayt bilan to'g'ridan to'g'ri aloqa mavjud bo'lмаган joyda ishlash uchun veb-sahifa yozilishi kerak bo'lsa, yana muammo ochiqligicha qolavergan. Hozirda barcha brauzerlarning yuqori talqinlarida HTML5 qo'llanishi inobatga olinsa, veb-sahifani DOCTYPEning soddagina ko'rinishidan boshlayveramiz. Demak, ilk qator doim quyidagicha bo'ladi:

<!DOCTYPE html>

Bu va tarkibni aniqlovchi boshqa teglar yozilmasa ham, brauzerlar ularni bor deb hisoblayveradi. Ammo biz yozayotgan kodimizga «ishlasa bo'ldi» degan nuqtayi nazar bilan qaramaymiz, to'g'ri va talab bo'yicha bexato kod yozishni o'rganamiz. Zero, biz yozgan ixtiyoriy kodni tahlil qiluvchi tashqi dastur yoki o'ziga indeksatsiya qiluvchi (*ma'lum qolipda ko'chirib oluvchi*) izlash tizimlari (*masalan: Google, Yandex, Rambler, Bing va hokazolar*) ham nima yozganimizni to'g'ri va to'liq tushunsin.

HTMLning ko'pgina teglari hozirda eskirgan deb qaraladi va foydalanish tavsiya etilmaydi. Ammo o'z vazifasini hali ham bajaraveradi. Kelgusida bunday «eskilik sarqitlari»dan ham butunlay voz kechiladi. Sababi, bir necha yillardan keyin yangi brauzerlar bunday kodlarni akslantirmay qo'yish ham mumkin.

Yana bir muhim jihat, har bir elementning paydo bo'lish joyi va shakliga mos ravishda teglarni o'z o'rniда yozishga katta ahamiyat berish lozim, aks holda, turli chalkashliklar kelib chiqib, keyinchalik kamchilikni bartaraf etish mushkullashishi mumkin.

61645

2.3. Asosiy qismlar

HTML ishoratlar bilan ishlovchi til bo'lgani uchun ham, avvalo, ishorat o'zi nima va qanday ko'rinishga egaligini aniqlab olishimiz kerak. Ingliz tilida uni URL — «Uniform (*avval uning o'rnida Universal so'zi ishlatalig'an*) Resource Locator» deb nomlash ommalashgan, ya'ni manbaning yagona (universal) ko'rsatkichi. Bu global (butunjahon) tarmoq — internetda joylashgan manbaning manzilini aniqlovchi ishorat. Uning umumiy ko'rinishi quyidagi shaklda bo'ladi:

[uslub://taxallus:kalitso'z@xost:port/yo'l?omillar#bo'lak](http://taxallus:kalitso'z@xost:port/yo'l?omillar#bo'lak)

Bunda belgilangan so'zlar quyidagi ma'nolarni anglatadi:

- **uslub** — veb-manbaga murojaat qilish uslubi, ko'p hollarda tarmoq qaydnomasi nazarda tutiladi. Massalan: http, ftp, https, telnet, mailto, file, data, news va hokazo.

- **taxallus** — manbaga kirish imkoniyati bo'lgan foydalanuvchining taxallusi.

- **Kalit so'z** — foydalanuvchining maxfiy kirish so'zi. Shuning uchun ham ba'zida kalit so'zlarda «@» belgisini ishlatish tavsiya etilmaydi.

- **xost** — domen (sayt, server kompyuter) nomi yoki uning IP manzili.

- **port** — ulanish amalga oshiriladigan port raqami.

- **yo'l** — manbaning xostda joylashgan manzilini aniqlashtiruvchi yo'l. Bunda aniq serverdag'i imkon

berilgan papka manzili yoki qiymatlar ketma-ketligi taqsim («/») bilan ajratilgan holda ko'rsatilishi mumkin. Server o'zining sozlanishga qarab, qiymatlarni mos tarzda qabul qiladi.

▪ **omillar** — sahifani ko'rish uchun qo'shimcha ravishda kiritilishi kerak bo'lgan qiymatlar, ular «&» belgisi bilan ajratilgan holda o'zgaruvchi nomiga beriladi.

▪ **bo'lak** — sahifa uzun bo'lgan hollarda aynan qaysi bo'lak ekran boshida ko'rinishini ta'minlovchi belgilanish.

Agar manba hamma uchun ochiq bo'lsa, unda taxallus va kalit so'zlar yozilmaydi. Port raqami ko'rsatilmagan hollarda uni qaydnomaning odatiy qiymatida berilganday qabul qiladi. Ishoratga misol ko'rib o'tamiz:

<http://yurt.uz/?a=16>

Omillarda ko'rsatilgan «a»ning qiymati o'zgartirilsa, sahifa ham mos ravishda o'zgaradi.

HTML uchta asosiy qismdan tashkil topadi.

Birinchisi – DOCTYPE. U haqida oldingi mavzuda to'xtalib o'tdik. [2.2.] U hujjat XML, XHTML yoki butunlay boshqa tilda emas, aynan HTMLda yozilganligini anglatadi. Hujjatni o'qiyotgan brauzer yoki boshqa dastur aynan qanday kelishuv asosida tahlil qilish kerakligini anglab olishi uchun bu juda muhim.

Ikkinchisi – bosh qismi. U **head** tegi ichida yoziladi va hujjatni shakllantirish uchun kerakli ma'lumotlarni o'zida jamlaydi. Unda hujjat aynan qaysi kodlamadaligi,

nishonchasi, sarlavhasi hamda sahifaga qo'shimcha yulkanadigan fayllar ro'yxati va boshqa muhim ma'lumotlar yoziladi. Har bir qo'shimcha ko'rsatilgan fayl yuklanib bo'lishi bilan ishga tushadi, ya'ni undagi kodlar tahlil qilinib akslantiriladi. Shuning uchun ham bosh qismida brauzerda ko'rinvuchi ifodalar yozib bo'lmaydi deya olmaymiz. Ammo bunday yozish mantiqiy xato bo'lganligi bois tavsiya etilmaydi.

Uchinchisi – asosiy qism yoki tana qismi. U **body** tegining ichida yoziladi va brauzerning akslantiruvchi maydonini to'la egallaydi. Barcha ko'rinvuchi (hatto ko'rinxaydigan) tasvirlar undan o'rinni oladi. Shu o'rinda aniqlik kiritib o'tish kerakki, hozirda head va body teglarining keskin farqi qolmagan. Birida yozilishi kerak bo'lgan teg boshqasida yozilsa ham, hech xatosiz bajarilaveriladi. Bu, bir tarafdan, juda chalkashlikka olib kelsa-da, ikkinchi tarafdan, kerakli faylarni faqat unga murojaat bo'lgandagina yuklab olib bajarishni ta'minlaydi va shu bilan tezkor ma'lumot almashinib, saytlarning tez ishlashiga zamin yaratadi. Modomiki shunday ekan, biz **head** va **body** orasida mantiqiy chegara yasab, «bosh qism»ga faqat hujjatni shakllantirishda kerak bo'ladigan tavsif-xabarlar¹ yig'adigan bo'lim sifatida; «asosiy qism»ga esa brauzerda ko'rinish va ma'lumot akslantirishga xizmat qiladigan bo'lim deb qaraymiz.

¹**Tavsif xabar** – uni meta-ma'lumotlar deb ham yuritiladi. Ammo biz xabar deganda katta hajmli ma'lumotni emas, faqat asosiy mazmunni ifodalovchi qisqa axborotni tushunamiz. “Meta” degan old qo'shimchaning aniq tarjimasi mavjud emas, **metadata** – hujjat aslida qanday va nima-dan iboratligini aniqlovchi tavsif-xabar degan ma'noni bildiradi.

Keyinchalik kiritilgan DOCTYPEni inobatga olma-ganda, barcha teglar, aniqrog'i, ikkinchi va uchinchi qismlar yagona **html** tegi ichida yoziladi. DOCTYPE, HTML, HEAD, BODY – bu kodning bo'limlarini bildi-rib, bitta veb-sahifa uchun faqat bir marta yoziladi.

2.4. Sahifaning bosh qismi

Kelishib olgan mantiqiy chegaramizga binoan bosh qismga, ya'ni **head** tegida faqat veb-hujjatimizni akslan-tirishda kerak bo'ladigan ma'lumotlarni yozamiz. Quyi-da keltiriladigan tavsif xabarlarni, faqat va faqat bosh qismda yozish tavsiya etiladi. Aks holda, barcha brau-zerlarda bir xil namoyon bo'lmashligi mumkin. Masalan, kodlamani albatta hujjatning ilk 512 bayti ichida e'lon qilish shart. Aks holda, brauzer o'zining odatiy qiymati bo'yicha yoki ochilayotgan hujjatning operatsion tizim aniqlagan kodlamasida ko'rsatilganday akslantiradi.

HTMLning 5-talqinida kodlamani ko'rsatish sintak-sisi ancha soddalashtirilgan. U quyidagicha yoziladi:

```
<meta charset="UTF-8">
```

Tashqi fayllarning ko'rsatishda to'liq URL ishlatalishi yoki joriy hujjatning joylashgan o'rniiga nisbatan yo'li ko'rsatilishi mumkin. Masalan, biror rasmni ko'rsatish uchun uni joriy fayl yonidagi «img» jildiga joylagan bo'lsak, oddiygina «/img/rasm.png» ko'rinishida unga

murojaat qilinadi. Bu, albatta, yozilayotgan kodning qisqa bo'lishini ta'minlash bilan birga domen o'zgarganda faylni topa olmay qolish xatoligidan ham saqlaydi. Agar veb-sahifa yaratish jarayonida bir necha manbadan ishorat ko'rsatish zaruriyati bo'sa, unda tashqi jildlar ro'yxati **base** tegida ko'rsatilishi kerak, sintaksisi:

```
<base href=ishorat [target=nomlanish]>
```

Bunda:

- **ishorat** — tashqi manbaning katalog manzili.
- **nomlanish** = blank | self | parent | top — ochilayotgan ishorat yuklanishi kerak bo'lgan oyna yoki freym nomi.
 - blank — ishoratni brauzerning yangi oynasiga ochadi.
 - self — odatiy qiymat, ishoratni brauzerning joriy oynasiga ochadi. Freym [5.7.] mavjud bo'lmasa, quyidagi alomatlar ham xuddi shu singari ishlaydi.
 - parent — freym ichidagi ishoratni tashqi sahifada ochadi.
 - top — freymlarni bekor qilib, ishoratni joriy oynada ochadi.

Tashqi fayllarni bog'lash uchun **link** tegidan foydalaniladi. Uning to'la sintaksisi quyidagicha:

```
<link href=ishorat [type=fayl turi] [title=matn]  
[charset=kodlama] [hreflang=til belgisi]  
[rel=munosabat] [sizes=o'lcham]  
[media=qurilma turi]>
```

- **ishorat** — bog'lanayotgan faylning manzili.
- **fayl turi** — faylning MIME (*Multipurpose Internet Mail Extension* — *internet pochtasining ko'p maqsadli kengaytmasi*) turi: text/html, text/css, image/gif va hokazo ko'rinishda bo'ladi. Ko'rsatilgan fayl aslida nima ekanligini anglatadi. Nomlanish bir qaraganda tu-shunarli, lekin uning to'liq va to'g'ri ro'yxatini aniqlab bo'lmaydi, u doim kengayib boraveradi. HTML5da agar **rel** ko'rsatilgan bo'lsa, fayl turini ko'rsatish shart emas.
- **kodlama** — kodlama nomi. Agar boshqasini ish-latishga jiddiy sabab bo'lmasa, UTF-8 dan foydalanish tavsiya etiladi. Bu turli tillardagi harf va belgilanishlar ni to'g'ri akslantirishga kafolat beradi.
- **munosabat** — veb-sahifaga nisbatan unga bog'lanayotgan fayl qanday munosabatdaligini aniqlaydi. Bunda yoziladigan kalit so'zlar HTMLda mavjud tu-shuncha bo'lsa, bog'lanayotgan faylni o'sha tushunchada ifodalanganday tahlil qilib yuklanadi. Masalan, unga «stylesheet» deb yozilsa, CSS jadvali sifatida qaraladi; «alternate» deb yozilsa, u HTMLda emas, unga bosh-qacha belgilanishlar tilida (XML yoki boshqada) yozilgan hujjat sifatida qaraladi.
- **matn** — bog'langan fayl sarlavhasi.
- **til belgisi** — ikkita harfdan iborat til belgilanishi, masalan: uz, ru, en.
- **o'lcham** — bu alomat HTML5da kiritilgan, lekin brauzerlar uni qo'llashni boshlagani yo'q. Odatda, ko'rsatilgan rasm qanday kattalikda bo'lmasin, uni brauzer 16x16 o'lchamgacha kichiklashtirib veb-sahifa nishon-chasi sifatida akslantiradi. Matematik nuqtayi nazardan

umumjahon kelishuviga asosan, birinchi eni, keyin lotincha «x» harfi, so'ogra bo'yi nuqtalar hisobida yoziladi. Hozircha ushbu alomatdan foydalanish tavsiya etilmaydi.

▪ **qurilma turi** — fayl qanday qurilmalar uchun ishlashini aniqlaydi:

- **all** — barcha qurilmalar (odatiy qiymati).
- **braille** — ko'zi ojizlar uchun Brayl tizimi asosida ishlovchi qurilma.
- **handheld** — ekrani kichik smartfon yoki shunga o'xhash qurilmalar.
- **print** — chop etish qurilmalari.
- **screen** — monitor ekrani.
- **speech** — ovozli qurilmalar, matnni o'quvchi brauzerlar.
- **projection** — proyektorlar.
- **tty** — imkoniyati cheklangan ekranlar, terminallar, ixcham qurilmalar.
- **tv** — televizorlar.

LINK tegi joriy veb-sahifa akslanishi uchun kerak bo'ladigan qo'shimcha fayllarni biriktirish uchun ishlatilganligi sababli ham, u HTMLning bosh qismida yoziishi kerak bo'lgan teg bo'lgani uchun ham unga yozilgan ma'lumotlar veb-sahifaga nishoncha qo'yishdan boshqa hollarda hech bir ko'rinishda brauzerda akslan-tirilmaydi. Faqat yangiliklardagi qisqacha daraklarnigina ko'rsatib boruvchi RSS (*Really Simple Syndication*) yoki boshqa shunga o'xhash XML fayllari ko'rsatilgan bo'lsa, faqat Opera brauzeri uni ishorat yoziladigan oynachaning o'ng qismida **title** alomatida yozilgan nom

bo'yicha ko'rsatadi. Qolgan hollarda yozilgan ma'lumotlar faqat brauzer va HTMLga ishlov beruvchi dasturlar uchun muhim ahamiyat kasb etadi.

Bosh qismda yozilishi kerak bo'lgan **meta** tegida sahifaning qaysi tildaligi, muallifi va boshqa xususiyatlarni aniqlash mumkin. Ammo HTMLning 5-talqinida bunday xususiyatlarni aniqlash vazifasi yangi kiritilgan teg va alomatlarga yuklatilganligi sababli unga ko'p to'xtalmaymiz, faqat sintaksisini keltirib o'tamiz:

```
<meta [name=nom] [http-equiv=tarkib turi]  
[content=tarkib]>
```

Veb-sahifa uchun tilni ko'rsatish, izlash tizimlari yoki boshqa dasturlar lotin harflarida yozganimizda uni turkcha, kirill harflarida yozganimizda esa ruscha deb xato aniqlab olmasligi uchun juda muhim. Til meta tegida emas, hujjat turini aniqlovchi «doctype»dan tashqari barcha kodni o'zi ichiga oluvchi **html** tegida aniqlanadi. U quyidagi alomatlarga ega:

- **lang** — til belgilanishi.
- **manifest** — sahifa yuklanayotganda bajarilib turuvchi dastur manzili.

Veb-sahifaning sarlavhasi, aniqrog'i, nomlanishi **title** tegi yordamida beriladi. Ushbu nom brauzerning eng yuqori qismida nishonchadan so'ng ko'rindi.

Yuqorida bayon etilgan bilimlarga asoslanib bosh qismga kattaroq e'tibor qaratgan holda eng kichik HTML kodini yozamiz:

```
<!DOCTYPE html>
<html lang="uz">
    <head>
        <meta charset="utf-8">
        <base href="http://yurt.uz/" target="_self">
        <link rel="alternate" href="/rss"
              type="application/rss+xml"
              title="RSS daraklar">
        <link rel="alternate icon" href="ico.gif">
        <title>HTML 5 va CSS 3 saboqlari</title>
    </head>
    <body>
        HTML5dan ilk saboq
    </body>
</html>
```

2.5. Sahifaning asosiy qismi

Asosiy qism, ya'ni **body** bo'limida yoziladigan har bir teg brauzerda ma'lum obyekt yaratadi. Aslida, body tegining o'zi ham brauzerning to'g'ri to'rtburchak shaklidagi namoyish maydonini to'liq egallaydi. Ammo barcha obyektlar ham ko'rinishdigan bo'lmasligi, hattoki ular alohida-alohida joylangan bir necha boshqa obyektlarining ustidan o'rinni olishi mumkin, ya'ni u ustki qatlamda akslantirilgan bo'ladi. Nima bo'lganda ham, barchasi body tegining ichki elementlari hisoblanib, body eng ostki qatlamdan o'rinni oladi. Uning

barcha imkoniyatlarini o'ziga ko'chirib olgan div mukammal teg hisoblanib, belgilanishlar tilida muhim ahamiyat kasb etadi.

Veb-sayt yaratish ekanmiz, avvalo, u qanday bo'limlardan tashkil topgan bo'lishi kerakligini aniqlab, unga mos ravishda ekranni to'g'ri to'rtburchak shaklidagi bo'laklarga bo'lib olish lozim. Eskiz tayyor bo'lgandan keyin brauzer o'zi ishlayotgan kompyuter va monitor, operatsion tizim va foydalanuvchilarning tanloviga qarab turli o'lchamda bo'lishi mumkinligini inobatga olgan holda, har bir obyektning hajmini va barcha o'lchamlarning o'zgaruvchanligini hisobga olib belgilanishlar tilida kod yozishni boshlagan ma'qul.

Sayt umumiy holda quyidagicha ko'rinishda bo'ladi:

Sarlavha qismi		
Sayt bo'limlari va xizmatlari		
Chap bo'lim	Asosiy bo'lim Bo'lim yuqori tavsiflari	O'ng bo'lim
Bo'lim matni	Bo'lim matni	Bo'lim matni
Bo'lim quy'i tavsiflari		
Holat satri va dasturlovchilar ma'lumotlari		

HTMLning avvalgi talqinlarida har bir joylashuvni aniqlovchi teg bo'limgan, shuning uchun dasturlovchilar **div** tegini turlicha xususiyatlar bilan hamma yerda qo'llashga majbur bo'lishgan.

5-talqinda joylashtirishni qulaylashtiruvchi va yagona qolipga soluvchi maxsus teglar kiritilgan. Yuqoridagi ko'rinishga ega veb-saytimizni mavjud teglar asosida chizib chiqamiz. Qaysi qism kerak bo'lmasa, uni hosil qiluvchi teglar yozilmasligi kifoya.

Saytning bosh sarlavhasi, kichik mavzusi, sohasi va boshqa xususiyatlari, odatda, **h1**, **h2**, **h3**, **h4**, **h5** va **h6** teglari ichida yozilishi tavsiya etiladi. Ularning yagona alomati mavjud:

<hN align = left | center | right | justify></hN>

- **left** — odatiy qiymat, chap tarafga tekislaydi.
- **center** — satr maydon o'rtasida joylashadi.
- **right** — o'ng tarafga tekislaydi.
- **justify** — bir necha satrli yozuvlarni ikki tarafga ham tekislaydi.

Eng katta sarlavha **h1** da, eng kichigi **h6** da yoziladi. Ularni umumlashtirish uchun **hgroup** tegi ichida yozish tavsiya etiladi.

Sayt sarlavhasi, tamg'asi, shior va targ'ibotlari bilan birgalikda uning bo'limlari, izlash va boshqa asosiy xizmatlarini o'z ichiga olgan qism – saytning bosh qismi hisoblanadi va u **header** tegi ichida yoziladi.

Asosiy qismda ishoratlar «a» tegi yordamida beriladi. [3.5.] Saytning bo'limlari va xizmatlarini ko'rsatuvchi ishoratlarga joylashgan qismi **nav** tegi ichida yozilishi tavsiya etiladi. Boshqa ishoratlarni nav ichiga olib umumlashtirish taqilanganadi.

O'ng va chapda joylashgan bo'limlar **aside** tegi ichida yozilishi kerak.

Veb-sahifaning asosiy, takrorlanmaydigan, urg'u beriladigan qismi **main** tegi ichida yozilishi tavsiya etiladi. Ushbu teg bitta sahifa uchun yagona bo'lishi va mos ishorat bo'yicha o'ziga xos ma'lumotlarni ichiga olishi shart.

Sahifaning asosiy maqola chiqadigan qismi **article** tegida berilishi kerak. Holat sarti, dasturlovchilarning mualliflik huquqi, sayt qachon tashkil topgani va shunga o'xshash ma'lumotlar **footer** tegida joylashishi talab etiladi. Boshqa zarur bo'ladigan bo'laklarning teglari **section** yordamida ajratilishi lozim.

Yuqoridagi kelishuvlarni inobatga olgan holda sahifaning asosiy qismini quyidagicha tarkibda yozishimiz mumkin:

```
<body>
  <header>
    <hgroup>
      <h1 align="center"> HTML5dan ilk saboq </h1>
      <h4 align="right">
        Mukammal dasturlash kitobidan
      </h4>
      <h6 align="right"> 1-kitob </h6>
    </hgroup>

    <nav>
      <a href="html.htm">HTML5</a>
      <a href="css.htm">CSS3</a>
      <a href="script.htm">JavaScript</a>
    </nav>
  </header>
```

```
<aside id="left">
  <p>HTML5</p>
  <p>CSS</p>
  <p>JavaScript</p>
</aside>

<aside id="right">
  <nav>
    <a href="abook.htm">Kitoblar</a> <br>
    <a href="sbook.htm">Qo'llanmalar</a> <br>
    <a href="hbook.htm">Ma'lumotnomalar</a>
  </nav>
</aside>

<main>
  <article>
    Ushbu joyda sahifaning asosiy urg'u beriladigan
    ma'lumotlari joylashadi.
  </article>
</main>

<footer> © 2021 </footer>

</body>
```

Albatta, u kutilganday chiroyli shaklda namoyon bo'lmaydi. Uni o'zimiz xohlaganday chiqadigan qilishimiz uchun CSSdan foydalangan holda (*8-bo'limda batatsil bayon etilgan*) maxsus jilolar bilan yozish kerak, keyingi mavzularda obyektlarga bezak berishni misollar bilan tanishib o'tamiz.

2.6. Alomatlar qiymatlarining turlari

Ba'zi istisnolardan tashqari umumiy holda teglar quyidagicha ko'rinishdagi sintaksisda yozilishi lozimligi bilan tanishdik:

<teg nomi [alomat nomi = qiymat] [...]> ... </teg nomi>

Bunda «**qiymat**» birtirnoq ichida berilishi yoki ajratuvchi belgilari qatnashmagan holda umuman qo'shtirnoqsiz yozilishi mumkin, ammo turli chalkashliklarga sabab bo'lmaslik uchun, yuqorida ta'kidlaganimizday, ikkitalik qo'shtirnoq ichida yozishni tavsiya etamiz.

Ba'zi teglarning qiymat berish majburiy bo'lgan alomatlari bo'ladi. Masalan, rasmni ko'rsatuvchi tegga, albatta, rasmning manzilini berish shart, aks holda, u tegning samarasi bo'lmaydi. Boshqa teglarda esa alomatning nomini yozish kifoya qiladi, ularga qiymat berish shart emas. Ushbu hollar faqat mantiqiy «rost» yoki «yolg'on» qiymat qabul qiladigan alomatlargagina taalluqli. Bunday teg va alomatlar bilan kelgusida bat afsil tanishamiz, hozir alomatlarga beriladigan qiymatlar qanday ko'rinishlarda bo'lishini aniqlab olishimiz lozim. Buning uchun qiymatlarni bir necha turlarga ajratamiz:

- **Mantiqiy tur** (boolean) — faqat «rost» yoki «yolg'on» qiymatlaridan birini qabul qiladi. Rost qiymat «true», «on», «yes» singari so'zlardan biri orqali ifodalanadi. Yolg'on qiymat esa «false», «off» yoki

«no» so'zlari bilan belgilanadi. Agarda rost ham emas, yolg'on ham emas qiymat berib, natija qabul qilishni brauzerning o'ziga qo'yib berishga to'g'ri kelsa, unday holda «auto» so'zini qo'llash lozim. Bunda natija yana qandaydir qo'shimcha shart bajarilishiga qarab hisoblanadi.

▪ **Butun** (integer) — bu elementar matematikada qo'llaniladigan butun son.

▪ **Haqiqiy** (real) — bu faqat o'nlik, ya'ni nuqta bilan yoziladigan matematik haqiqiy son. Cheksiz va davriy kasrlar ushbu turga kirmaydi.

▪ **Son** (number) — bu butun va haqiqiy sonlar umumlashmasi, o'z o'rnidagi ular «-» va «+» ishoralarini bilan yozilishi mumkin.

▪ **O'lcham** (size) — bu, odatda, «son»dan keyin uning qaysi nisbiy kattalikda ekanligini aniqlash uchun qo'llaniladi. Nol har qanday kattalikda birday nol (0)ni anglatgani uchun undan keyin «o'lcham» qo'yish shart emas. Nisbiy kattaliklar quyidagilardan iborat:

- **px** — piksel, qurilmaning eng kichik o'lchamli elementi.
- **em** — mos husnixatdagi «m» harfi enichalik kattalik.
- **ex** — mos husnixatdagi «x» harfi balandliga teng o'lcham.
- **cm** — santimetr.
- **mm** — millimetrr.
- **in** — dyuym, taxminan 2,54 santimetr.
- **pt** — nuqta, bu o'lcham Macintosh tizimlari uchun. Masalan, 12 pt Macintoshda 12 ta nuq-

taga teng bo'lsa, Windows tizimida 16 dan 22 gacha oraliqda bo'lishi mumkin.

- **pc** – pika, o'lchami 12 ta «nuqta» (pt) kattaligiga ($\approx 35/83$ smga) teng.

▪ **Foiz** — sondan keyin «%» belgisini qo'yish bilan ifodalanadi. Uning qiymati tashqi element o'lchamiga nisbatan aniqlanadi.

▪ **Qisqa o'lcham** — hech qanday kattaliksiz, faqat son orqali ifodalanadi. Agar sonning o'zi berilsa, o'lchamni shu soncha piksel sifatida qabul qiladi. Agar sondan keyin «*» qo'yilgan bo'lsa, obyektning o'lchami yonidagilardan shuncha marotaba farqlanishini anglatadi. Masalan, jadvalning bitta ustuni ikkinchisidan uch marta katta bo'lishi talab etilsa, birinchisiga «1*» (yoki «*»ning o'zi), ikkinchisiga «3*» qiymat berilishi kerak.

▪ **Ishorat** — URL, uning qandayligi bilan avval tanishganmiz. [2.3.]

▪ **Burchak** — burchak va aylana yoqlarini aniqlash uchun sondan keyin quyidagi kattaliklar ishlataladi:

- **deg** – gradus. Geometriyada o'rganganimizday, u 0 dan 360 gacha qiymat qabul qiladi. Albat-ta, manfiy qiymat berish ham mumkin, masalan: $-10\text{deg} = 350\text{deg}$
- **grad** – gradian. 0..400 oraliqdagi butun sonlarni qabul qiladi. 100 «grad» – to'g'ri burchak.
- **rad** – radian. 0.. π oraliqdagi haqiqiy sonlarni qabul qiladi. $1.5707\text{rad} \approx \frac{1}{2}\pi$ to'g'ri burchakni anglatadi.
- **turn** – qancha qism aylantirishni bildiradi, 0..1 oraliqdagi haqiqiy sonni qabul qiladi.

- **Vaqt** — sondan keyin quyidagi kattaliklar bilan aniqlanadigan o'lchov:
 - **s** — soniya.
 - **ms** — millisoniya.
- **Chastota** — o'lchovi sondan keyin quyidagi kattaliklar bilan aniqlanadi:
 - **hz** — gers.
 - **khz** — kilogers.
- **Belgi** — ASCII jadvalida keltirilgan bir bayt hajmli yagona kod.
- **Satr** — qo'shtirnoq ichida yoziladigan bir qator dan iborat belgilar ketma-ketligi. Satr bittalik yoki ikkitalik qo'shtirnoqqa olinishi mumkin, qay biri bilan ochilsa, o'sha bilan yopiladi. Shuning uchun ham biring ichida ikkinchisidan foydalanish mumkin, ammo ichma-ich qo'llanilmaydi. Satr ichida ayni qo'shtirnoq ishlatalishi kerak bo'lsa, «\»dan («orttaqsim»dan) so'ng yoziladi yoki uning maxsus kodidan foydalaniadi. [2.1.]
- **Matn** — xatbosilar bilan biriktirilgan «satr»lar majmuasi.
- **SDATA** — bu oddiy (teg va buyruqlarsiz) matning brauzer akslantiradigan ko'rinishi demakdir. Unda satrlarni bo'lувчи belgilar, tabulyatsiyalar umumlash tirilib, bitta bo'sh joy sifatida yoziladi. Maxsus kodlar akslanadigan ko'rinishida keltiriladi. U qo'shtirnoq ichida yozilmaydi, tegning yakunini anglatuvchi «</>» belgilari juftligi uchraganda CDATA ham o'zining poyoniga yetganini tushunadi.
- **Aniqlangan tur** — bunday tur bir qancha qabul qilinishi mumkin bo'lgan «satr»larning to'plamidan

tashkil topadi. Noto‘g‘ri yoki aniqlanmagan qiymat berilgan holda brauzer odatiy qiymatni oladi. Masalan, matnning yotiq joylashishini aniqlovchi **align** alomati faqat «left», «justify», «center» hamda «right» qiymatlaridan birinigina qabul qila oladi. Unga qiymat berilmaganda yoki ushbu to‘rtta so‘zdan boshqa narsa yozilganda odatiy qiymati «left»ni qabul qilgandagidek ishlaydi. [2.5.]

▪ **Rang** — tasvirlanish rangini belgilovchi kattalik. U standart nomi bilan yoki RGB (qizil, yashil, ko‘k) taqsimoti yordamida, o‘noltilik sanoq tizimi bo‘yicha har uchala rang uchun noldan FF gacha qiymat berilib, «#» belgisidan so‘ng ularni birlashtirib yozish orqali ifodalanadi. Qizil (#FF0000) – red; Yashil (#00FF00) – green. (*Muqovaning orqasida ommabop rang nomlari bilan RGB kodlanishlari keltirib o‘tilgan*)

▪ **MIME** — fayl turini aniqlovchi satr. [2.4.]

▪ **Zarurmas qiymat.** Ba’zi alomatlarga qiymat bermasa ham, uning mavjudligining o‘zi kutilgan natijani beradi. Masalan, obyektdagi «checked» – tanlashni anglatuvchi alomati, uning teg xususiyatlarida ishtirok etishining o‘zi obyektning qiymati rost (*belgilangan*) ekanligini anglatadi. Agar boshqa alomatlar singari bir xil andozada yozish lozim bolsa, qiymat sifatida o‘zining nomi beriladi: **checked="checked"**

▪ **Identifikator.** Har bir tegning «id» alomatiga sahifa ichida takrorlanmasligi talab etiladigan identifikator berish mumkin, faqat «**name**» alomatini nomlashda «kelishilgan identifikator»dan [2.1.] foydalansa bo‘ladi.

Sondan keyin ko'rsatiladigan kattaliklar hech qanday bo'sh joy va ajratuvchilarsiz, sonning oxirgi raqamidan so'ng unga biriktirib yoziladi.

2.7. Universal alomatlar

HTML tilida hamma teglar (*odatda, body tegida yozilishi kerak bolgan teglar nazarda tutiladi*) qo'llaydigan universal alomatlar mavjud. Ularni quyida alifbo tartibida tavsiflari bilan batafsil keltirib o'tamiz:

- **accesskey** — qiymati: belgi, «0»dan «9»gacha raqamlar va «a»dan «z»gacha lotin harflarini qabul qiladi. Tugmalar birikmasi (qo'shtugma) bosilganda brauzerdag'i obyekt faollashadi va uning ustida amal bajarish imkoniyati paydo bo'ladi. Ammo qo'shtugmalar hamma brauzerde ham bir xil emas. Har bir brauzerning o'z foydalanuvchilari unga biriktirilgan qo'shtugmalarga ko'nikma hosil qilib olgan. Shu o'rinda eng ommabop brauzerlar uchun **accesskey="x"** qiymat berilganda qanday qo'shtugma bosish lozimligini keltiramiz:

- **Internet Explorer:** Alt + X
- **Microsoft Edge:** Alt + X
- **Chrome:** Alt + X
- **Opera:** Shift + Esc, X
- **Safari:** Alt + X
- **Firefox:** Shift + Alt + X

Unutmashlik lozimki, ushbu birikmalar Windows ope-ratsion tizimi uchun berilgan MacOsda «Ctrl» tugma-sini qo'shib bosish talab etiladi. Linuxda esa ayrim birikmalar ishlamasligi ham mumkin.

▪ **class** — qiymati: aniqlangan turlar. CSS yorda-mida aniqlangan jilo nomi beriladi. Bir nechta nomlar bo'sh joy bilan ajratib yoziladi. Nomlar faqat lotin harflari, raqamlar, chiziqcha va tagchiziq (_) yordamida berilishi mumkin. CSS bilan keyingi bo'limlarda bat afsil tanishamiz.

▪ **contenteditable** — qiymati: aniqlangan mantiqiy tur (true yoki false). Brauzerdagi obyekt qiymatini foydalanuvchi o'zgartirishi mumkinligini aniqlaydi. Alo-matning o'zi yozilib, qiymati berilmagan holda uni rost (true) deb qabul qiladi. Odatiy qiymatni esa tashqi tegnikidan o'zlashtirib oladi.

▪ **contextmenu** — qiymati: aniqlangan tur, unga menu tegi yordamida yaratilgan ro'yxatning IDsi beriladi. Shunda obyekt ustida o'ng tugma bosilganda chiquvchi menyuga ko'rsatilgan ro'yxat qo'shilishi kerak, afsuski, HTMLning 5-talqinidagi ushbu ajoyib imko-niyat hozircha hech bir brauzerda tatbiq etilmagan.

▪ **dir** — qiymati: aniqlangan tur. Matnning namo-yon bo'lish yo'nalishini aniqlaydi: ltr — chapdan o'nga, rtl — o'ngdan chapga. Yunikod kodlamasida arab va yahudiycha tillar yo'nalishi o'ngdan chapga deb biriktirilgan, shuning uchun «dir» ularga ta'sir qilmay-di. Ammo boshqa hollarda matn yo'nalishini o'zgarti-rish uchun «rtl»ni qo'llasa bo'ladi, «ltr» esa uning odatiy qiymatidir.

▪ **hidden** — obyektning brauzerda ko'rinmasligini ta'minlaydi (zarurmas qiymat). Uni ishlatish tavsiya etilmaydi, bunday imkoniyat CSS yordamida berilgani ma'qul, shuning uchun ham ba'zi brauzerlar uni qo'llamaydi.

▪ **id** — identifikator. Obyektning takrorlanmas nomini berish uchun foydalaniladi. [2.1.]

▪ **lang** — obyekt ichida qaysi til belgilanishlaridan foydalanish kerakligini anglatadi, qiymati: aniqlangan tur. Shu o'rinda standart bo'yicha qaysi tillardan foydalanishini keltirib o'tamiz.

1-jadval. Muomala tillarining kodlari:

Til	kod	Til	kod	Til	kod	Til	kod
Abxaz	ab	Dzong-ke	dz	Maoriy	mi	Tojik	tg
Aymara	ay	Zulu	zu	Maratxi	mr	Tay (siam)	th
Alban	sq	Ivrit	he	Moldav	mo	Tamil	ta
Amxar	am	Idish	yi	Mongol	mn	Tatar	tt
Angliy	en	Indoneziy	id	Nauru	na	Tvi	tw
Arab	ar	Interlingva	ia	Nemetskiy	de	Telugu	te
Armyan	hy	Interlingve	ie	Nepal	ne	Tibet	bo
Assam	as	Yoruba	yo	Norvej	no	Tigrinya	ti
Afar	aa	Irland	ga	Ozarbayjon	az	Tongan	to
Afan (oromo)	om	Island	is	Oksitan	oc	Tsonga (shangaan)	ts
Afg'on (pushtu)	ps	Ispan	es	Oriya	or	Turk	tr
Afrikan	af	Italian	it	Pandjabi	pa	Turkmen	tk
Bask	eu	Kanada	kn	Persid (fors)	fa	O'zbek	uz
Bashkir	ba	Katalan	ca	Pol	pl	Uyg'ur	ug
Belorus	be	Kashmiri	ks	Portugal	pt	Ukrain	uk

Til	kod	Til	kod	Til	kod	Til	kod
Bengali	bn	Kechua	qu	Retoroman	rm	Urdu	ur
Birman	my	Kinyaruanda	rw	Rumin	ro	Farer	fo
Bislama	bi	Koreys	ko	Rus	ru	Fidjiy	fj
Bixari	bh	Korsikan	co	Saami	sz	Fin	fi
Bolgar	bg	Kosa	xh	Samoan	sm	Fransuz	fr
Breton	br	Kurd	ku	Sanskrit	sa	Friz	
Valliy	cy	Kurundi	rn	Sanyo	sg	Xausa	ha
Venger	hu	Kxmer	km	Serb	Sr	Hindi	hi
Volapyuk	vo	Qirg'iz	ky	Serboxorvat	sh	Xitoy	zh
Volof	wo	Qozoq	kk	Sesoto	st	Xorvat	hr
Vietnam	vi	Laos	lo	Setsvana	tn	Chex	cs
Galisiy	gl	Latviy	lv	Singal	si	Chivenda	ve
Golland	nl	Lotin	la	Sindxi	sd	Chishona	sn
Grenland	kl	Lingala	ln	Sisvazi	ss	Shved	sv
Greche	el	Litov	lt	Slovan	sk	Esperanto	eo
Gruzin	ka	Lujitskiy (sorb)	sb	Sloven	sl	Eskimos (yuit)	ik
Guarani	gn	Makedon	mk	Somali	So	Eskimos (inuit)	iu
Gudjarati	gu	Malagasiy	mg	Suaxili	sw	Eston	et
Gel	gd	Malay	ms	Sundan	su	Yavan	jv
Dat	da	Malayalyam	ml	Tagal	tl	Yapon	ja
Djuang	za	Maltiy	mt				

Ba'zi tillar bir nechta davlatlarda turlicha talqinda qo'llaniladi. Xatolik va chalkashliklar yuzaga kelmasligi uchun har bir tilni qo'llaniladigan davlatiga qarab belgilash kiritilgan. Ushbu kitob mukammal qo'llanma bo'lganligi bois o'quvchilarimizga qo'shimcha adabiyot tavsiya qilish o'rniiga tillar jadvalini to'laligicha keltirishni ma'qul topdik.

2-jadval. Davlatlar bo'yicha muomala tillarining kodlari:

Ingliz		Arab		Ispan	
Avstraliya	en-au	Jazoir	ar-dz	Argentina	es-ar
Beliz	en-bz	Bahrayn	ar-bh	Boliviya	es-bo
Buyuk Britaniya	en-gb	Misr	ar-eg	Venesuela	es-ve
Irlandiya	en-ie	Iordaniya	ar-jo	Gvatemala	es-gt
Kanada	en-ca	Iraq	ar-iq	Gonduras	es-hn
Karib oroli	en	Yaman	ar-e	Dominikan	es-do
Yangi Zelandiya	en-nz	Qatar	ar-qa	Ispaniya	es
AQSh	en-us	Kuvayt	ar-kw	Kolumbiya	es-co
JAR	en-za	Livan	ar-lb	Kosta-Rika	es-cr
Yamayka	en-jm	Liviya	ar-ly	Meksika	es-mx
Golland		Marokash	ar-ma	Nikaragua	es-ni
Belgiya	nl-be	BAA	ar-ae	Panama	es-pa
Niderlandiya	nl	Oman	ar-om	Paragvay	es-py
Gel		S.Arabiston	ar-sa	Peru	es-pe
Irlandiya	gd-ie	Suriya	ar-sy	Puerto-Riko	es-pr
Shotlandiya	gd	Tunis	ar-tn	Salvador	es-sv
Italian		Rus		Urugvay	es-uy
Italiya	it	Moldaviya	ru-mo	Chili	es-cl
Shveysariya	it-ch	Rossiya	ru	Ekvador	es-ec
Xitoy		Olmon		Fransuz	
Gonkong	zh-hk	Avstriya	de-at	Belgiya	fr-be
KNR	zh-cn	Germaniya	de	Kanada	fr-ca
Singapur	zh-sg	Lixtenshteyn	de-li	Lyuksemburg	fr-lu
Tayvan	zh-tw	Lyuksemburg	de-lu	Fransiya	fr
Portugal		Shveysariya	de-ch	Shveysariya	fr-ch
Portugal		Rumin		Shved	
Braziliya	pt-br	Moldaviya	ro-mo	Finlyandiya	sv-fi
Portugaliya	pt	Ruminiya	ro	Shvetsiya	sv

▪ **spellcheck** — qiymati: aniqlangan tur («yes» yoki «no»), brauzer yozuv to'g'rigilini tekshirish-tekshirmsligini aniqlaydi. Odatiy qiymatini brauzerda sozlasa

bo'ladi. Ushbu alomatni hamma teg uchun yozish mumkin, lekin samarasi faqat matnini tahrirlash mumkin bo'lgan obyektlardagina bilinadi. Alomat e'lon qilinib, qiymati berilmagan holda uni rost (yes) deb qabul qiladi. Afsuski, o'zbek tili grammatikasini tekshirish hali hech qaysi ommaviy dasturlarga kiritilmagan.

▪ **style** — obyektga CSS jilolarini berish uchun foydalaniadi, qiymati CDATA.

▪ **tabindex** — brauzerda «Tab» tugmasi bosilganda navbatma-navbat faollahish ketma-ketligini aniqlaydi, qiymati musbat butun son. «Tabindex» ko'rsatilmagan yoki «0» qiymat berilgan obyektlar yaratilish tartibi bo'yicha indekslangan obyektlardan keyin faollahadi. Yashirin va nofaol qilingan obyektlar bu ketma-ketlikda ishtirok etmaydi. Indeksi kichik obyektdan indeksi kattasi keyin faollahadi, shuning uchun qat'iy ketma-ket keladigan natural sonlar ishlatish shart emas, orada bir necha son tushib qolgan bo'lsa ham, navbat bo'yicha keyingisiga o'tib ketaveradi. Teskari tartibda ketma-ket tanlash uchun, odatda, Shift+Tab birgalikda bosiladi.

▪ **title** — obyekt ustiga sichqoncha borganda chiqadigan izohchani aniqlaydi, qiymati: satr. Uning ko'rinishini brauzer va operatsion tizimning o'zi aniqlaydi, unga uzundan uzoq va bir necha qatorli matnlar yozish tavsiya etilmaydi.

2.8. Hodisalar

Hodisalar — brauzerda qandaydir amal bajarilayotgan paytda yoki o'sha lahzadan avval, balki keyin bajariluvchi JavaScript kodlarini aniqlovchi buyruqlar majmuyi. Hodisalar obyektlarga tegning alomati sifatida yoziladi, qiymati esa satr ko'rinishida beriladi. Hodisalarga buyruqlar yozishni JavaScriptga bag'ishlangan kitobimizda bat afsil ko'rib o'tamiz, hozircha faqat HTML alomatlari hisoblanganligi bois hodisalar nomlarini sanab o'tamiz.

2.8.1. Sahifaga bog'liq hodisalar

Bular BODY tegida yoziladi:

- **onafterprint** — hujjat chop etilgandan keyin;
- **onbeforeprint** — hujjat chop etilmasdan avval;
- **onbeforeunload** — hujjat yuklanmasdan oldin;
- **onblur** — hujjat nofaol bo'lganda;
- **onerror** — hujjat yuklanayotganda xatolik bo'lsa;
- **onfocus** — hujjat faollashganda;
- **onhaschange** — hujjat tarkibi o'zgartirilganda;
- **onload** — hujjat to'la yuklanib bo'lganda;
- **onmessage** — hujjat xabar qabul qilganda;
- **onoffline** — server bilan aloqa uzilib qolganda;
- **ononline** — hujjat serverga (qayta) ulanganda;
- **onpagehide** — hujjat ko'rinmas qilib qo'yilganda;
- **onpageshow** — hujjat akslantirilganda;
- **onpopstate** — tashriflar tarixi o'zgarganda;

- **onredo** — tashriflar tarixidagi keyingi sahifaga o'tganda;
- **onresize** — hujjat oynasining o'lchami o'zgartirilganda;
- **onundo** — tashriflar tarixidagi oldingi sahifaga o'tganda;
- **onunload** — hujjat oynasi yopilganda.

2.8.2. Sichqoncha harakatiga bog'liq hodisalar

- **onclick** — sichqoncha bosilganda;
- **oncontextmenu** — matnbop menu chiqarilganda (odatda sichqonchaning o'ng tugmasi bosilganda);
- **ondrag** — sichqoncha yordamida element ushlanib harakatlantirilayotganda;
- **ondblclick** — sichqoncha ikki marta chertilganda;
- **ondragend** — sichqoncha bilan element harakatlantirilishi tugaganda;
- **ondragenter** — sichqoncha bilan obyekt ustiga element keltirilganda;
- **ondragleave** — sichqoncha bilan ushlangan element obyekt ustidan ketganda;
- **ondragstart** — sichqoncha bilan element ushlanganda;
- **ondrop** — sichqoncha bilan obyekt ustiga element keltirib tashlanganda;
- **onmousedown** — sichqoncha bosilganda;
- **onmousemove** — sichqoncha harakatlanayotganda;
- **onmouseout** — sichqoncha obyekt ustidan ketsa;

- **onmouseover** — sichqoncha obyektga kirganda;
- **onmouseup** — bosilgan sichqoncha tugmasi qo'yib yuborilganda;
- **onmousewheel** — sichqoncha g'ildiragi (surgichi) aylantirilganda;
- **onscroll** — obyektning surish tasmasi qo'zg'atilganda;

2.8.3. Klaviaturaga bog'liq hodisalar

- **onkeydown** — tugma bosilganda (bosilgan zahoti, hali qo'yib yuborilmasadn avval);
- **onkeypress** — tugma chertilganda (bosib qo'yib yuborilganda);
- **onkeyup** — bosilgan tugma qo'yib yuborilganda.

2.8.4. Forma hodisalari

Bular FORM tegida yoziladi:

- **onblur** — forma nofaol bo'lganda;
- **onchange** — formada o'zgartirish bo'layotganda;
- **onfocus** — forma faollashganda;
- **onformchange** — formada o'zgartirish bo'lganda;
- **onforminput** — foydalanuvchi formaga ma'lumot kiritayotganda;
- **oninput** — foydalanuvchi forma elementiga o'zgarish kiritayotganda;
- **onselect** — formaning elementi tanlanganda;
- **onsubmit** — formadagi ma'lumotlar yuborilganda.

2.8.5. Media hodisalar

- **onabort** — fayl yuklanishi uzilib qolganda;
- **oncanplay** — ijro etish uchun yetarlicha faylning qismi yuklanib bo'lganda;
- **oncanplaythrough** — oxirigacha uzlusiz ijro etish uchun yetarlicha yuklanganda;
- **ondurationchange** — fayl uzunligi o'zgarganda;
- **onemptied** — fayl yuklash imkoniyati yo'qolganda;
- **onended** — faylni oxirigacha ijro etib bo'lganda;
- **onerror** — fayl yuklanishida xatolik bo'lsa;
- **onloadeddata** — fayl yuklanayotganda;
- **onloadedmetadata** — tavsif-xabarlar [2.3.] yuklanayotganda;
- **onloadstart** — fayl yuklanishi boshlanganda;
- **onpause** — ijro to'xtamga keltirilganda;
- **onplay** — fayl ijro etishga tayyor bo'lganda;
- **onplaying** — ijro etish boshlanganda;
- **onprogress** — tavsif-xabarlarni hisoblayotganda;
- **onratechange** — ijro etish tezligi o'zgarganda;
- **onreadystatechange** — fayl holati o'zgarganda;
- **ontimeupdate** — vaqt bo'yicha o'zkazib ijro etilganda;
- **onvolumechange** — tovush balandligi o'zgarganda;
- **onwaiting** — ijro to'xtatilganda.

3-BO'LIM

MATNNI JOYLASHTIRISH

3.1. Xatboshi va bloklar

Matnlarni joylashtirishda bir muhim narsaga e'tibor qaratishimiz kerak. Barcha teglar, xususan, matnga shakl beruvchi va uning qanday akslanishini aniqlovchilar ham ikki toifaga bo'linadi: blokli va satrli teglar [8.8.]. Blokli teglar shunisi bilan farqlanadiki, ular o'zi joylashgan element maydonining barcha kengligini egalelaydi, balandligi esa ichidagi obyektlarga bog'liq ravishda aniqlanadi, shu bilan birga, doim yangi qatordan boshlanadi. Shuning uchun matnning davomidan ulanuvchi satr blokli teglarga olinmaydi.

Yuqorida [2.5.] **h1** dan **h6** gacha blokli teglarni ko'rib chiqdik. Ularni matn orasida sarlavhani ifodlashda qo'llasa bo'ladi. Lekin yagona sahifa bir necha mavzulardan iborat uzundan uzoq matndan iborat bo'lishi maqsadga muvofiq emas. O'quvchi bitta cho'zilgan sahifani aylantirib o'qishi juda zerikarli. Bitta mavzu ikkinchisidan yaqqol ajralib turmaydi. Shuning uchun har bir mavzuni alohida sahifada ochiladigan qilish lozim. Buni inobatga olgan holda ushbu teglarni faqat

sahifaning yuqorisidagi bosh qismida qo'llashni tavsiya etamiz.

Yuqorida **br** tegi bilan tanishganimizda uni keyingi qatorga tushirishni ta'minlovchi teg sifatida bayon etdik. Lekin bu degani yangi xatboshidan boshlashni anglatmaydi. Chunki, odatda, xatboshidagi birinchi satr qolganlariga nisbatan ichkarida joylashadi. BR tegi nafaqat satrlarni, balki obyektlarni ham pastki qatorga tushirishda qo'llaniladi. Satrlarning orasini ochish uchun bir nechtasini ketma-ket yozish hollari ham kuzatiladi. Aslida, bu noto'g'ri usul. Obyektlarning joylashishi faqat CSS yordamida ko'rsatilishi kerak. Shunday ekan, ushbu BR tegini matnlardagi satrlarni bo'lish uchungina qo'llash tavsiya etiladi. Agar xatboshi lozim bo'lsa, «
 » singari kod yozish, aslida, akslanish bo'yicha talabga javob bersa-da, mutlaqo noo'rin. Xatboshi maxsus teg – **p** (paragraf so'zidan olingan) blokli teg yordamida ifodalanadi. U matnni qanday usulda tekislashni ko'rsatuvchi alomatga [2.5.] ega:

<p [align="left | center | right | justify"]> ... </p>

Agar tegni yopishni unutgan bo'lsangiz, uning ta'sir doirasi keyingi blokli teg ochilgan joygacha bo'ladi.

Xatboshidagi birinchi satr qolganlaridan ichkarida bo'lishi qanday ta'minlanishini CSS bilan tanishganda bat afsil ko'rib chiqamiz.

3.2. Matnni jilolash

HTML tili matnni joylashtirishga mo'llanlanganligi bois ham aksariyat teglar matnga jilo berishga yo'naltirilgan. Har bir belgi, har bir so'z, har bir qator va xatboshilarni, qolaversa, butun sahifadagi matn qanday akslanishi lozimligini satrlar ichidagi teglar yordamida aniqlab yozib boriladi. (Yozilmagan taqdirda odatiy qiyamatlar asosida akslantiriladi.) Matnni jilolash uchun xizmat qiladigan teglarning ochiluvchi va yopiluvchi qismlari yozilishi qat'iy talab etiladi, ta'siri ham uning ichki qismida yozilgan matn uchun amal qiladi.

Quyida jilo beruvchi satrli teglarning bat afsil ro'y-xatini keltiramiz:

- **b** va **strong** — qalin matn.
- **mark** — matnnning muhim qismini ajratib ko'rsatish uchun foydalilanildi.
- **big** — odatdagidan bir birlik katta o'lchamli matn.
- **cite**, **q** — ko'chirmani ifodalovchi matn.
- **code**, **var** — dasturiy kodlarni anglatadi, barcha harflar eni bir xil bo'ladi.
- **em**, **dfn** va **i** — egri matn.
- **ins** va **u** — tagiga chizilgan matn.
- **kbd**, **tt** va **samp** — enli matn, har bir harf eni bir xil o'lchamda bo'ladi.
- **small** — odatdagidan sal kichikroq o'lchamli matn.
- **strike**, **del** va **s** — (ustidan) chizilgan matn.
- **sub** — indeksga tushirilgan matn.
- **sup** — darajaga chiqarilgan matn.

Yuqoridagi ro'yxatdan ko'rinish turibdiki, ba'zi bir teglar bir xil vazifani bajarar ekan. Bu turli xil matn tahrirlovchilarida ishlovchilar uchun ham tushunarli bo'lishi uchun ulardagi qisqartmalar bilan olingan. Ushbu teglarning ayrimlari eksirgan, qo'llash tavsiya etilmaydi. Ammo hech biridan butkul voz kechilmagani uchun ularning barchasini keltirib o'tishni lozim topdik. Amaliyotda qo'llashning o'rni kelib qolishi mumkin, aslida, bularning hammasini CSS yordamida ifodalash mumkin. HTML tili ilk yaratilgan vaqtida CSS tu-shunchasi bo'lмаган, shuning uchun ham matnlarga jilo berish uchun teglardan foydalaniлган. Matnning rangini va husnixatini (shriftini) o'зgartirish uchun font tegi qo'llanilgan.

Uning sintaksisi quyidagicha:

** matn **

Bunda:

- **o'lcham** matnning kattaligini aniqlab, faqat 1 dan 7 gacha butun son qabul qiladi. Agar sahifaning bosh qismida **basefont** yordamida husnixat o'lchami ko'rsatilgan bo'lsa, unga nisbatan FONTning hajmini mos ravishda «+1», «+2», ... qiymat berib matnni kattalishtirish yoki manfiy sonlar yordamida kichiklashtirish mumkin. Albatta, bu yerda qo'shish yoki ayirishdan so'ng matn kattaligi 1 va 7 ning orasida bo'ladi deb qaraladi. Boshqacha qiymat berilgan vaziyatda esa HTMLning o'z qoidasi bo'yicha odatiy qiymatda matn akslanaveradi.

- **rang** — maxsus so'z yoki kodi bilan aniqlangan [2.6.] ranglardan biri.
- **nom** o'rniliga operatsion tizimda mavjud husni-xatlarning nomi ko'rsatiladi. Misol:

```
<font size="+1" color="blue" face="Arial">
    ko'k matn
</font>
```

Veb-sahifalarda e'lon qilingan ma'lumotlar qachon kiritilganligini ko'rsatish uchun joriy vaqtga nisbatan yozuv sifatida namoyon etish odat tusiga kirib ulgurdi. Masalan, Ovro'padagi saytda chiqqan maqola ularning vaqtini bilan ko'rsatilgandan ko'ra o'quvchining brauzeri joriy mahalliy vaqtga nisbatan falon soat yoki daqiqa avval deb ko'rsatib tursa, aniqroq va tushunarliroq bo'ladi. Umuman olganda, vaqtini ko'rsatish uchun **time** tegi kiritilgan. Uning «pubdate» degan alomati mavjud bo'lib, ma'lumot kiritilgan vaqt ko'rsatilishi nazarda tutilgan edi. Lekin amaliyot natijasida ushbu alomat ortiqchaligi ma'lum bo'ldi va uni qo'llash tavsiya etilmaydi. Vaqtini ko'rsatish uchun yagona «datetime» alomati kifoya. Maxsus skript orqali time tegi ichidagi yozuvni joriy vaqtga nisbatan hisoblab ko'rsatish mumkin. Misol:

```
<time datetime="2021-03-10T18:17:11+05:00">
    5 daqiqa avval
</time>
```

Ba'zi qadimiy tillarda, xususan, arab yozuvida ham o'ngdan chapga qarab yoziladi, bu esa biz ko'nikkan

yo'nalishga teskari. Bunday matnlarni akslantirish uchun teskari tartibda har bir qatorni aniq yozib chiqish juda murakkab. Bu muammoni bartaraf etish uchun **bdo** tegi kiritilgan:

```
<bdo dir="ltr | rtl"> ... </bdo>
```

- **ltr** — odatiy qiymat, matn chapdan o'ngga qarab chop etib boriladi.
- **rtl** — matn (right to left) o'ngdan chapga qarab chop etib boriladi.

Agar yo'nalishi o'zgartirilgan yozuv ichida biror jumla asl holida ko'rinishi shart bo'lsa, uni **bdi** tegi ichida yozish lozim. Masalan, arab yozuv ichida inglizcha so'zni keltirish zaruriyati tug'ilsa, HTMLda qo'shilgan ushbu teg orqali ortiqcha murakkabliksiz buni amalgalashish mumkin. (Afsuski, Internet Explorer va Edgelarning hozirgi talqinlari uni qo'llamaydi.)

Umuman olganda, CSS yordamida jilo berib, span tegi ichida satrni istagan ko'rinishda akslantirish mumkin.

3.3. Eskirgan teglar

1996-yildan HTMLda kodlarni qisqartirish, sodda-tushunarliroq yozuv uslibiga o'tish uchun ham CSS kiritilib, keng ko'lamda qo'llanila boshladi. Uni takomil-

lashtirish bilan matnlarni joylashtirish va ularga jilo berishga mo'ljallangan ko'pgina HTML teglarni qo'llashga ehtiyoj qolmadi. Bundan tashqari, ba'zi teglarning o'rnnini bosadigan, ommabop tarzda nomlangan va yagona standartga keltirilgan yangi teglarga qo'shildi. Ushbu taraqqiyot bosqichlari ba'zi teglarni eskirgan deb qarashni taqozo etadi. Ulardan foydalanish tavsiya qilinmaydi, ammo oldin yozilgan kodlar o'z vazifasini bajarishi uchun ham hamma brauzerlar ularni qo'llaydi. Shunday bo'lsa ham, yaqin kelajakda ulardan butunlay voz kechilishi yoki ular yagona standartga keltirilmaganligi uchun ham turli brauzerlarda turlicha akslanishi mumkin. Agar yangi DOCTYPEda inobatga olinmagan bo'lsa, ba'zi qurilmalarda xato kod sifatida e'tiborsiz qoldirilishi ham mumkin. Shuning uchun W3C tavsiyalarini inobatga olishimiz muhim.

Quyida eskirgan (HTML5da qo'llash umuman tavsiya etilmaydigan) teglarni ko'rib chiqamiz, ulardan foydalanmasligimizni inobatga olgan holda to'liq sintaksisi keltirilmaydi:

- **acronym** — qisqartmalar (HTML, CSS, JS va boshqalar). Uning o'rniiga **abbr** qo'llaniladi.
- **applet** — Java dasturlash tilida yoziladigan kodlar. Ushbu tegning ochiluvchi va yopiluvchi qismlari orasiga, ya'ni teg asosiga matn yozilsa, u appletni qo'llamaydigan brauzerlarda akslanadi. Qo'llovchi brauzerlar kodning natijasini ko'rsatadi. Uning o'rniiga universal hisoblangan **embed** tegidan foydalanish tavsiya etiladi.

▪ **center** — ichidagi elementlarini tashqi obyektga nisbatan yotiqlichizig'i markazida joylashtiradi. Bu tegni barcha brauzerlar qo'llagani uchun ko'pgina dizaynerlar hali ham undan keng foydalanishadi. Aslida, o'mniga CSS qoidasini kiritish lozim. [8.]

▪ **dir** — OT (operatsion tizim) fayl va jildlari ro'y-xatini ko'rsatish uchun mo'ljallangan. Uning o'mniga umumiylashgan, ro'yxat ko'rsatuvchi UL tegi qo'llaniladi.

▪ **frame** — boshqa HTML faylni joriy sahifada ko'rsatish. Bunday yo'l, aslida, noto'g'ri, lekin ba'zan sahifaning bir qismida butunlay boshqa jiyo va qonuniyatlar bilan yozilgan manbadagi ma'lumotlarni ko'rsatishga to'g'ri kelib qolishi mumkin. Agar undagi kodlar joriy sahifaga, aytaylik, oddiy DIV tegi ichiga joylashtilsa, tabiiyki, o'z jozibasini yo'qotadi va ko'rinishi buzilib ketadi. Shunga o'xshash noiloj hollarda freymdan foydalanishga majbur bo'linadi. Lekin **frame** o'mniga **iframe** qo'llash mumkin.

▪ **frameset** — bir qancha freymlardan tashkil topgan sahifalarda qo'llanilgan. Agar u ishlatilsa, BODY tegidan foydalanmaslikka to'g'ri keladi.

▪ **isindex** — ma'lumot kiritiladigan oyna oldiga yoziladigan jumlaning bitta satrda bo'lishini ta'minlovchi, indekslangan matn. U sahifaning bosh (HEAD) qismida e'lon qilib qo'yiladi. Keyin brauzer ushbu jumlaniga chiziq chizib akslantirishi kerak. Ya'ni bitta jumla ikki marta yozilishi talab qilinadi, bu esa mukammal

dasturlash talablariga mos kelmaydi. Ma'lumot kiritish oynasi qo'llanilishi lozim bo'lsa, input va label teglari dan foydalanish kifoya. **ISINDEX** tavsiya etilmaydigan teg bo'lganligi bois ko'pgina brauzerlar kutilganidek namoyish etmaydi.

▪ **menu** — ushbu teg ham ro'yxatni akslantirishga mo'ljallangan va uning o'rniga «ul»dan foydalaniladi. HTML5da **menu** tegi butunlay boshqa maqsadda kiritilgan.

▪ **nextid** — brauzer va HTML tahrirlovchilari uchun keyingi sahifani topib olish uchun qo'llanilgan. Hozirda umuman ishlatilmaydi.

▪ **nobr** — ichiga yozilgan matn va elementlarning faqat bitta qatorda chiqishini ta'minlaydi. U ma'no jihatidan **br** ga teskari vazifani bajaradi. O'rniga CSS qo'llash tavsiya etiladi.

▪ **noembed** — agar brauzer tegishli plagiini (brauzerning imkoniyatlarini oshiruvchi qo'shimcha dasturcha, masalan, Flash, QuickTime va boshqalarni) qo'llama-sa, teg asosidagi tasvir ko'rindi, aks holda, teg e'tiborsiz qoldiriladi.

▪ **noframes** — ichida yozilganlar, freymlarni qo'llamaydigan brauzerlarda akslanadi. Barcha zamonaviy brauzerlar freymni qo'llagani bois uni e'tiborsiz qoldiradi. Shuning uchun ham uni ishlatishning ma'nosи yo'q.

▪ **object** — brauzer tushunmaydigan kodlar (tovushlar, lavhalar, dasturlash kodlari va boshqalar) qanday akslantirilishi lozimligini aniqlaydi. Uning o'rniga universal **embed** tegi ishlataladi.

- **plaintext** — matn qanday yozilsa, shunday, HTMLning shartli o'zgartirishlarisiz, bo'sh joy va bo'sh satrlarni o'z holicha akslantirish. Uning o'rniga PRE tegi yoki alohida fayl sifatida yuklansa, «text/plain» MIME-turi qo'llaniladi.

- **xmp** — kod qanday yozilgan bo'lsa, shundayligicha akslantiradi. Uning PREdan farqi shundaki, yopilma-guncha ichidagi teglarni ham yozuv sifatida ko'rsata-veradi. Ehtimol, shuning uchun ham buni ishlatish tavsiya etilmas, ammo HTML va dasturiy kodlarni chiqarish uchun qulay vosita. Boshqa hollarda «<» va «>»larni «<» va «>»lar yordamida yozib chiqishga to'g'ri keladi.

Quyidagi teglar bevosita CSS yordamida ifodalanadi:

- **big** — husnixatni bir birlikka kattalashtiradi.
- **listing** — kodlarni chop etish uchun mo'ljallangan, husnixatni kengaytirilgan va sal kichiklashtirilgan holda ko'rsatadi.
- **font** — husnixatni o'zgartirish. [3.2.]
- **rb** — ruby tegidagi asosiy matnni ajratish uchun foydalanilgan. Hozirda umuman qo'llanilmaydi.
- **blink** — pirpiraydigan (yonib-o'chadigan) matn.
- **basefont** — sahifaning bosh (HEAD) qismida FONTdan tashqari barcha teglar uchun qo'llaniladigan husnixat nomi, o'lchami va rangini aniqlaydi.
- **strike, s** — ustiga chizilgan matn.
- **u** — tagiga chizilgan matn.
- **tt** — «tengenli» yozuv. Har bir harf yoki belgining eni bir xil o'lchamda bo'ladi. Uning o'rniga «Consolas»

yoki «Courier New» kabi husnixatlarni qo'llash tavsiya etiladi.

Bundan tashqari, eskirgan alomatlar ham mavjud. Ularning barchasi CSS yordamida ifodalanadi. Masalan, BODY tegida sahifaning qanday shakllanishini aniqlovchi background, bgcolor, link,alink va vlink singari alomatlarni qo'llash tavsiya etilmaydi. Har bir teg uchun qaysi alomati eskirgan ekanligini to'liq ro'yxat sifatida keltirib o'tmaymiz. Tegning sintaksisini to'la bayon etayotganimizda ularni izoh sifatida keltiramiz yoki umuman tushirib qoldiramiz. Foydalanilmaydigan bilim uchun bat afsil to'xtalishga hojat yo'q.

3.4. Ro'yxat

Sahifada ba'zan nimalarningdir ro'yxatini sanab ko'rishishga to'g'ri kelib qoladi. Umuman olganda, ketma-ket yoziladigan satrlarni ro'yxat sifatida ifodalash ko'p hollarda qulaylik tug'diradi. HTMLda 2 xil ko'rinishda ro'yxat mavjud: sanoqli ro'yxat va belgilangan ro'yxat.

3.4.1. Sanoqli ro'yxat

Bunda keltiriladigan elementlar har biri yangi satrda tartib raqam bilan akslantiriladi. Sintaksisi quyidagicha:

```
<ol [type="A | a | I | i | 1"]  
    [reversed] [start="butun son"]>  
    <li> satr 1 </li>  
    <li> satr 2 </li>  
    . . . . .  
</ol>
```

Bunda type ro'yxat qanday tartib raqamiga ega bo'lishini aniqlaydi:

- A – lotin alifbosining bosh harflari bilan;
- a – lotin alifbosining kichik harflari bilan;
- I – Rim raqamlari yordamida;
- i – Rim raqamlari bilan faqat kichik harflar yordamida;
- 1 – odatiy qiymat, arab raqamlari bilan o'nlik sanoq tizimida.
- start – alomatiga tartiblashning boshlang'ich qiymatini berish mumkin.
- reversed alomati qiymatga ega emas, agar u ko'rsatilsa, raqamlashni teskaridan kamayish tartibida akslantiradi. Masalan: 1, 2, 3 o'rniga 3, 2, 1.

HTML

```
<ol type="A" reversed start="C">  
    <li> sanoqli ro'yxat </li>  
    <li> belgilangan ro'yxat </li>  
</ol>
```

- Natija
- D. sanoqli ro'yxat
 - C. belgilangan ro'yxat

3.4.2. Belgilangan ro'yxat

Ammo har doim ham tartib raqamini ko'rsatish zarur bo'lavermaydi. Sanab o'tilayotgan elementlarning barchasi bir xilda ahamiyatga ega bo'lsa, ularni xatboshidan ajratib ko'rsatish kerak bo'lgan hollarda belgilangan ro'yxatdan foydalaniladi. Uning sintaksisi quyidagicha:

```
<ul [type="disc | circle | square"]>
    <li> satr 1 </li>
    <li> satr 2 </li>
    . . .
</ul>
```

Bunda type ro'yxat qanday belgi bilan ko'rsatilishini aniqlaydi:

disc — doira; **circle** — aylana; **square** — kvadrat maydoncha.

HTML	<pre><ul type="square"> sanoqli ro'yxat belgilangan ro'yxat </pre>
Natija	<ul style="list-style-type: none">▪ sanoqli ro'yxat▪ belgilangan ro'yxat

HTML5da ro'yxatni menu tegi orqali ham ifodalash mumkin, lekin uni shunchaki ro'yxat emas, haqiqiy menyu sifatida foydalanish uchun kiritilgan. Ya'ni tugmani bosganda chiqadigan ketma-ket yoki yonma-yon turuvchi (odatda, oynalar tepasiga qo'yiladigan) tugmalarni namoyon etishga mo'ljallangan. Oddiy hollarda

«ul»ning o‘rniga «menu» yozish kifoya, faqat uning «type» alomati boshqacha qiymatlar qabul qiladi:

- **context** — ketma-ket chiquvchi menu (sich-qoncha o‘ng tugmasi bosilgani singari);
- **toolbar** — yonma-yon chiquvchi menu (bosh-qaruv paneli singari);
- **list** — shunchaki ro‘yxat, odatiy qiymat, uning «ul»dan farqi yo‘q.

3.4.3. Izohli ro‘yxat

Izohli ro‘yxat asosan atamalar va ularni tavsiflash uchun qo’llaniladi. Ro‘yxat **dl** tegi yordamida yaratiladi. Nomlanish **dt** tegi bilan beriladi, unga izoh esa **dd** yordamida ifodalanadi. Misol:

```
<dl>
  <dt> Firefox </dt>
  <dd> Eng qulay brauzer </dd>
  <dt> Chrome </dt>
  <dd> Eng tezkor brauzer </dd>
</dl>
```

Ro‘yxat yasovchi ushbu uchala teg alohida alomat-larga ega emas, ularning ko‘rinishini CSS yordamida o‘zgartirish mumkin.

3.4.4. Izohi yashirin ro‘yxat

Ko‘p hollarda sarlavhalarning o‘zini ro‘yxat sifatida keltirish kifoya bo‘ladi, ammo istalgan paytda mavzu

nima haqidaligini bilib olish uchun yashirin izohni ochib ko'rish imkoniyatini ham qoldirishga to'g'ri keladi. Bunday hollarda HTML5da yangi kiritilgan, izohi yashirin mavzularni yozishga mo'ljallangan **details** tegidan foydalanish maqsadga muvofiq. Uning ichida qo'llaniladigan **summary** tegida sarlavha yoziladi, qolgan barcha yozuv yashirin tarzda bo'ladi. Foydalanuvchi istagan paytda «batafsil» degan ma'noni bildiruvchi milni bosib, yashirin izohni ko'rish mumkin. Sarlavha bilan izohi ko'rinish turishi uchun details tegida zarurmas qiymatli «open» alomatini ko'rsatish kerak. Misol:

```
<details>
  <summary><b>HTML</b></summary>
  <i>Belgilanishlar tili</i>
</details>
<details open>
  <summary><b>JavaScript</b></summary>
  <i>Dasturlash tili</i>
</details>
```

► HTML
▼ JavaScript
Dasturlash tili

3.5. Ishorat

Ishorat HTML tilida alohida o'ringa ega muhim teg hisoblanadi. Shuning uchun ham u alifboning eng birinchi harfi bilan nomlangan. Aslida ishorat boshqa sahifani yuklash uchun xizmat qiladi, lekin zamонавиј

talablar ayni bir sahifani birligida bo'lgan o'zgarishi uchun qaytadan yuklash samarasiz ekanligini ko'rsatdi. Shuning uchun ayrim sahifalarda talabga qarab ishorat umuman uchramasligi ham mumkin. JavaScript yordamida sahifani to'liq qayta yuklamay, serverga so'rov yuborib, aynan qaytgan ma'lumotlarni akslantirish amaliyoti mavjud. Bu xususida keyingi kitobda batafsil yoritiladi.

Ishorat baribir o'z dolzarbligini yo'qotmagan, brauzerlar ularni maxsus ko'rinishda akslantiradi, ularning foydalanuvchiga ko'rinarli bo'lishini taminlaydi. Ayni bir sahifaning manzilini statik ravishda ko'rsatib qo'yish uni tezda topib ochishda samara beradi. Shularni inobatga olgan holda ishoratdan voz kechib bo'lmaydi.

Ishorat qo'yish sintaksisi:

< a name=nom href="ishorat" [alomatlar] > matn </ a >

Ushbu teg asosan ikki maqsadda qo'llaniladi:

1. Ko'rsatilgan ishoratdagi sahifani ochish uchun.
2. Sahifaning shu ayni joyiga langar (anchor) qo'yish uchun.

Birinchi holat eng ko'p qo'llaniladi, unda «matn»ga bosilganda «ishorat»dagi sahifa yuklanadi. Ikkinci holatda esa sahifaning o'sha joyiga «langar tashlanadi». Agarda sahifa uzun bo'lib, bitta ekranga sig'mayotgan bo'lsa, aytaylik ikkinchi bobni o'qish uchun bir qancha surishlar natijasida pastga tushishga to'g'ri kelsa, bunday noqulaylik tug'ilmasligi uchun «langar»dan foydalaniladi.

** Ikkinch bob **

singari yozilsa, brauzerdagi manzil kiritish darchasiga **http://...#bob2** deb murojaat qilinganda ekranda paydo bo'lgan sahifa tepaga surilib, aynan shu «Ikkinch bob» ko'rindigan holga keladi. Agar belgilangan langardan keyingi namoyon bo'ladigan tasvirlar bitta ekran maydonidan kichkina bo'lmasa, ushbu yozuv ekranda aks lantirilayotgan birinchi satrda ko'rindi, aks holda, surgich (scroll) eng pastga tushiriladi. Odatda, sahifa ochilganda boshida mundarija sifatida mavzular keltiriladi, ularda ishoratlar quyidagicha yoziladi:

** Ikkinch bob **

Shu mavzuga bosilganda ushbu sahifa tepaga surilib, quyirog'ida joylashgan yuqorida yozilgandagidek langar qo'yilgan qismi ekranga ko'rinarli tarzda namoyon boladi. Ta'kidlab o'tishimiz joizki, ushbu imkoniyatdan foydalanishdan hali butunlay voz kechilmagan bo'lsa ham, HTML5 uchun eskirgan hisoblanadi. Sababi, zamonaviy veb-sahifalarda «#»dan keyin umuman langarni anglatmaydigan muhim qiymatlar berish odatga kirgan.

Ushbu teg «**href**» va «**name**»dan tashqari yana bir qancha alomatlarga ega:

- **charset** — ishoratdagi matn qaysi kodlamadaligini aniqlaydi.
- **href** — o'tish kerak bo'lgan hujjatga ishorat, URL yoziladi.
- **hreflang** — ishoratdagi matn tilini aniqlaydi.
- **media** — «**href**»da ko'rsatilgan hujjat qanday qurilma uchun mo'ljallanganligi [2.4.].

- **name** — nomi, ushbu nom bilan brauzer uning joylashgan o'rnini topadi.
- **rel** — joriy hujjatga nisbatan ishoratda ko'rsatilgan fayl munosabati.
- **rev** — ishoratdagiga nisbatan joriy hujjat munosabati («rel»ning aksi).
- **tabindex** — «Tab» tugmasi bosilgandagi faolla-shish tartibini aniqlaydi.
- **type** — ishoratdagi hujjatning MIME turi. [2.4.] Ushbu alomat faqat «href» bilan birga qo'llaniladi. Tur ko'rsatilmagan holda odatiy qiymatini ishoratdagi fayl kengaytmasidan yoki uning bosh qismidagi tavsif-xabar-da ko'rsatilgan turdan oladi.
- **coords** — faol maydonning koordinatasini aniqlaydi. Ushbu alomatni ishlatish uchun avval **map** tegi yordamida sahifadan maydon ajratib olish lozim.
- **shape** — faol maydon shaklini aniqlaydi. Odatda, bu alomat bilan coords birga qo'llaniladi. Ular ko'pincha sahifaga rasm (yoki obyekt) qo'yib, uning qaysidir sohasidagina ishorat faol bo'lishini aniqlashda qo'llaniladi. Bunda ko'rsatilgan shakl bilan beriladigan koordinata mutanosib bo'lishi kerak. [5.1.]
- **download** — ishoratda ko'rsatilgan faylni yuklab olish shartligini anglatadi. Odatda, brauzerlar o'zida mavjud va qo'shimcha o'rnatilgan dasturiy ta'minotlar asosida media fayllarni o'zida ko'rsatishni ma'qul ko'radi, bu esa har doim ham maqbul yechim emas.
- **hreflang** — ishoratda ko'rsatilgan hujjatning qaysi tildaligini ko'rsatadi.
- **target** — yuklanayotgan sahifa qayerda akslani-shini aniqlaydi. [2.4.]

4-BO'LIM

JADVALLAR

4.1. Jadvalni e'lon qilish

Jadval — hisobotlarni namoyon qilish uchun eng keng qo'llaniladigan obyekt, akslanishi lozim bo'lgan matn yoki tasvirlarni maxsus satr va ustunlar yordamida joylashtirish uchun juda qulay vosita. Agar jadvalning joylashishi va o'lchamlari uchun aniq, qat'iy chegara qo'yilmasa, u ichidagi ma'lumotlar hajmiga qarab o'z ko'rinishini o'zgartirish xususiyatiga ega. Ba'zan butun boshli veb-sahifa bir jadval ichiga joylanganiga ham guvohi bo'lamiz (2.5. dagi andoza ko'zda tutilyapti). Bunday usulni qo'llash umuman tavsiya etilmaydi.

Jadval **table** nomli teg yordamida e'lon qilinadi, unda universal alomatlar [2.7.] bilan birga faqat o'ziga xos quyidagi maxsus alomatlardan foydalanish yoki o'rniga CSSni qo'llash mumkin:

- **align** — yotiq joylashishini aniqlaydi. Aniqlangan tur, qiymatlari: left, center yoki right. Agar «center» qiymati berilsa, jadvalning eni qanday kattalikda bo'lmasin, ikki tomonidan chetgacha teng masofa qoladi. Boshqa hech bir obyekt bunday dinamik xususiyatni o'zi namoyon qila olmaydi (eskirgan center tegi bundan mustasno), buning uchun maxsus CSSlardan foydalanishga to'g'ri keladi.

- **background** — jadvalning orqasiga qo'yiladigan rasm. Qiymati: ishorat.
- **bgcolor** — jadvalning ortrangi. Qiymati: rang.
- **border** — chegaraviy chiziqlar qalinligi. Qiymati: son.
- **bordercolor** — chegaraviy chiziqlar rangi. Qiymati: rang.
- **cellpadding** — chegaraviy chiziqdan katakdagi obyektgacha masofa, qiymati: son.
- **cellspacing** — kataklar orasidagi masofa. Qiymati: son.
- **cols** — ustunlar soni. Qo'llash shart emas, o'zi aniqlaydi. Qiymati: son.
- **frame** — jadval atrofidagi hoshiya qanday ko'rinishda bo'lishini ko'rsatadi. Qiymati — aniqlangan tur:
 - **void** — jadval atrofiga chegara chizilmaydi.
 - **border** — jadval atrofiga chegara chiziladi, odatiy qiymat.
 - **above** — jadvalning yuqori qismiga chegara chiziladi.
 - **below** — jadvalning quyi qismiga chegara chiziladi.
 - **hsides** — jadvalning quyi va yuqori qismiga yotiq chegaralar chiziladi.
 - **vsides** — jadvalning yon taraflariga tik chegaralar chiziladi.
 - **rhs** — jadvalning faqat o'ng tarafiga chegara chiziladi.
 - **lhs** — jadvalning faqat chap tarafiga chegara chiziladi.

- **height** — balandligi. Qiymati: son, foizda yoki pikselda ko'rsatiladi.

- **rules** — kataklarning qaysi tomonlarida chegaralar chizilishini aniqlaydi. Qiymati aniqlangan tur:

- **all** — kataklarning barcha chegaralari ko'rindi, `border > 0` bo'lsa, odatiy qiymat.
- **groups** — faqat umumlashtirilgan kataklar guruhlarining chegarasi chiziladi.
- **cols** — ustunlar orasidagi chegaraviy tik chiziqlarga ko'rindi.
- **none** — hamma chegaraviy chiziqlar yashirin, `border = 0` bo'lganda odatiy qiymat.
- **rows** — satrlar orasidagi chegaraviy chiziqlargina ko'rindi.

- **summary** — jadvalning qisqa tavsifi, bu izoh uchun, brauzerda aks etmaydi, ammo izlovchi dasturiy ta'minotlar bundan keng foydalanadi. Qiymati: satr.

- **width** — jadval eni, qiymati son, foiz yoki piksel hisobida. Agar jadval kataklaridagi obyektlarning umumiy hajmi ko'rsatilgan qiymatdan oshib ketsa, ushbu alomat e'tiborga olinmaydi va brauzer yangi qiymatni o'zi tanlaydi.

4.2. Jadvalning sarlavhasi

Jadval **table** tegi asosida yaratiladi, faqat uning ichidagina qo'llaniladigan bir necha teglar yordamida

qo'shimcha imkoniyatlari aniqlanadi. Jadvalning sarlavhasi **caption** tegi yordamida yoziladi. Aslida, ushbu sarlavha table tegidan oldin yozilgan satr ko'rinishida namoyon bo'lsa ham, uni alohida yozish tavsiya etilmaydi, chunki akslanish jarayonida sarlavha sifatidagi yozuv kutilganidan boshqa joyda paydo bo'lishi mumkin.

Aslida, jadval ichidagi barcha obyektlar akslanib bo'lgandan keyin ularning hajmiga qarab jadval o'z o'lchamini o'zgartirib, brauzerga moslaydi. Ba'zan ustunlar xususiyatini o'zimiz aniq kiritib qo'yishimiz lozim bo'la-di. Bunda jadval to'liq yuklanishini kutmay ustunlar ko'rsatilgan shaklda namoyon bo'la boshlaydi. Buning uchun har bir ustunga mos ravishda **col** tegi yordamida omillar belgilab berilishi kerak. COL tegi quyidagi alomatlarga ega:

- **align** – yotiq joylashishini aniqlaydi.

- **char** – qiymati: belgi, ustundagi barcha qiymatlardagi aynan shu belgi bitta tik chiziqqa to'g'irlab yoziladi. Masalan, haqiqiy sonlardan iborat ustun bo'lsa, **char=''** singari yozilganda ustundagi barcha sonlar shunday joylashadiki, ulardagi hamma «» bir tik chiziqda yotadi. Faqat yuqoridagi align alomatiga odatdagidan tashqari char qiymat berilgandagina (**align='char'**) bu holat inobatga olinadi. Afsuski, hozirda ushbu ajoyib imkoniyatni hech qaysi brauzer kutilganidek namoyon etmaydi. Kelajakdagi brauzerlarning yangi talqinlarida buni kutib qolamiz.

- **charoff** – ustundagi qiymatlarni qancha belgi kengligida surishni aniqlaydi. Qiymati butun son qabul

qilib, manfiy bo'lishi ham mumkin (buning taqdiri ham «char» singari).

▪ **span** – ko'rsatilgan alomatlar nechta ustunga tatbiq etilishini aniqlaydi.

▪ **valign** – tik ko'rinishda tekislash, qiymati aniqlangan tur:

- **baseline** – qatordagi yotiq chiziqqa (balandliklari bir xil bo'lgandagina) tekislaydi.
- **bottom** – katakning quyi qismiga tekislaydi.
- **middle** – katakning o'rta qismiga tekislaydi, odatiy qiymat.
- **top** – katakning tepe qismiga tekislaydi.

▪ **width** – ustun eni, qiymati son, foiz yoki piksel hisobida.

```
<table cellpadding="5" cellspacing="0">
  <col width="200" align="center" valign="top" span="3">
```

Aynan **col** bilan deyarli bir xil vazifani bajaruvchi **colgroup** tegi ham mayjud. Ularning yagona farqi shundaki, jadvalga rules="groups" alomati berilganda col bilan guruhlangan ustunlar chegaralari chizilmaydi, faqat colgroup bilan guruhlangan ustunlar chegaralari ko'rindigan tarzda akslantiriladi.

Bir necha qatorni sarlavha sifatida ajratib olish uchun **thead** tegidan foydalaniladi. Asosan ustunlarning nomlari yoziladigan satrlar uning ichida qo'llaniladi. Jadvalning oxirgi satrlari, odatda, jamlovchi qiymatlar yoziladigan qatorlar **tfoot** tegi ichida yoziladi. Qolgan jadvalning asosiy satrlari **tbody** tegiga olinadi. Bu uchala teg jadvalni uch qismga ajratib, ularga alohida

tus berish uchun qo'llaniladi. Odatda, bittasi ishlatilgan jadvalning ichida qolgan ikkitasi ham qo'llanilishi kutiladi. Bu teglarning qulayligi shundaki, ularning ichidagi har bir katak uchun alohida xususiyatlar berish talab qilinmaydi. Agar CSS yordamida bir marotaba ushbu tegiarga jilolar biriktirilsa, jadval tarzida bir qancha hisobotlar chiqarayotganimizda ularning barchasi birday shamoyilda sarlavha, yuqori, quyi va asosiy qismi ajragan holda bejirim namoyon bo'ladi.

4.3. Jadvalni shakllantirish

Jadvalni yaratishda eng muhim qadam – undagi satrlarni aniqlab olishda. Barcha jadvaldagi kataklar ma'lum aniqlangan satrda joylashadi. Jadvaldagi satr **tr** tegi yordamida aniqlanadi. U ham yuqorida qayd etib o'tilgan qiymatlar qabul qiluvchi quyidagi alomatlarga ega: align, bgcolor, bordercolor, char, charoff, valign.

Ta'kidlab o'tish joizki, satrning xususiyatlariga nisbatan unda joylashgan kataklarning xususiyatlari ustuvor ahamiyatga ega. Ya'ni, misol uchun, bitta qatorga **tr** alomatlari orqali umumiylrang berilsa-yu undagi kataklarning har biriga o'ziga xos boshqa rang berilsa, kataknинг rangi akslantiriladi.

Jadvaldagi kataklar **td** tegi yordamida ifodalanadi va ular albatta **tr** tegi ichida joylashishi shart. Shunga

qarab katakning qaysi qatorda akslantirilishi aniqlanadi. Jadvaldagи barcha ko'rinadigan obyektlar faqat va faqat katak ichiga yoziladi. Bir qator tugagandan so'ng keyingi qator boshlanguncha nimadir yozib, uning o'sha satrlar orasida akslanishini kutish kerak emas. Bu «table» tegining sintaksisiga zid va biz bilganimizday shu kabi xato kodlar brauzerlar tomonidan e'tiborsiz qoldiriladi yoki sahifaning boshqa joyida namoyon boladi.

Eng sodda, faqat bitta katakdan iborat jadval chizish uchun ham kamida uchta teg ichma-ich quyidagi tartibda yozilishi shart:

```
<table> <tr> <td> obyekt </td> </tr> </table>
```

Aslida, faqat bitta katakdan iborat bo'lsa, jadval o'rninga div tegi ishlataladi. Table tegi yordamida esa ikki o'lchamli jadvallar tushunarli, ko'zimiz o'rgangan tarzda akslantiriladi. Demak, td tegi jadval elementining asosi hisoblanadi. U qiymati butun son bo'lgan quyidagi alovatlarga ega:

- **colspan** — satrdagi bir necha kataknini birlashtiradi.
- **— bir ustundagi tagma-tag kataklarni birlashtiradi.**

Bundan tashqari, CSS yordamida ifodalasa bo'ladi-gan, eskirgan deya hisoblanuvchi, tavsiflari yuqorida keltirilgan omillarga ham ega: **align**, **char**, **background**, **bgcolor**, **bordercolor**, **charoff**, **height**, **nowrap** (yozuv faqat bitta satrda chiqishini ta'minlaydi), **valign**, **width**.

Yana deyarli ishlatilmaydigan va brauzerda akslanmaydigan katak ichidagi ma'lumotlarga izoh yozish va ularni o'zaro yoki sarlavha bilan bog'lash uchun mo'l-jallangan quyidagi eskirgan teglar ham mavjud: **abbr**, **axis**, **headers**, **scope**. Bularni qo'llash samarasiz va tavsija etilmagani bois, har biriga alohida to'xtalmaymiz. Agar veb-sahifa ma'lum dasturiy ta'minot yordamida tahlil qilinsa, kataklardagi ma'lumotlarni guruhash uchun ushbu omillar kerak bo'lishi mumkin.

Agar katak ichidagi yozuv sarlavha sifatida namoyon bo'lishi kerak bo'lsa, boshqalaridan ajrab turishi uchun, odatda, matn katak o'rtasiga qalin yozuvda joylashtiriladi. Buning uchun alohida alomatlardan foydalanib yotmasdan, shunchaki **td** o'rniغا **th** tegini qo'llash kifo-ya.

Jadvalni yaratishdan oldin unda nechta katak bo'lishini va qanday tarzda joylashishini aniqlab olish lozim. Har bir satrdagi kataklarning umumiyligi soni bir xil bo'lmasa, brauzer kutilganidek akslantirmaydi. Agar hamma katak o'lchami bir xil bo'lмаган jadval chizish kerak bo'lsa, avval bir xil o'lchamli kataklardan iborat jadval tuzib, keyin **colspan** va **rowspan**lar yordamida bir necha kataklarni umumlashtirishga to'g'ri keladi. (*Ummat olganda, jadval chizishdan avval Microsoft Excel dasturini tasavvurga keltirish loyihalashtirishda yaxshi samara beradi.*)

4.4. Jadval chizish

Jadval chizishda har bir katagi o'zining alohida chegaraviy chiziqlariga ega ekanligini unutmaslik lozim. Ya'ni kataklar orasidagi yoki chetki katak bilan jadval chegarasi orasidagi chiziqlar, aslida, ikki qatlamdan iborat. Bu «cellspacing»ga noldan katta qiymat berilganda aniq ko'rindi. Agar ikki katakn ni umumlashtirmagan holda, faqatgina ularning orasidagi chegara ko'rinnmasligini istasak, har ikki katakning mos chegaraviy chiziqlarini yashirishga to'g'ri keladi. Bunga misolni CSSga bag'ishlangan [8.] bo'limda ko'rib o'tamiz. Brauzerlarning oldingi talqinlarida agar katak ichida ko'rindigan obyekt bo'limasa, uning chegarasi chizilmas edi, bu esa jadvalni anchagini xunuk akslantirardi. Shuning uchun eski brauzerlarda chizilishi lozim bo'lgan bo'sh kataklar ichiga «ko'rinvuchi bo'shliq» () qo'yish odait tusiga kirgandi. Yangi talqindagi brauzerlarda bu shart emas, faqat sahifa eski brauzerlar bilan ishlashga ham mo'ljallangan bo'lsa, buni inobatga olish kerak.

Nº	Tur	Mahsulot	Dona	Vazn
1.	Meva	Olma	120 ta	23,1 kg
2.		Nok	60 ta	13,4 kg
3.		Behi	36 ta	17,6 kg
4.		Anor	24 ta	11,2 kg
5.	Poliz	Qovun	4 ta	22,6 kg
6.		Tarvuz	3 ta	19,7 kg
7.		Handalak	10 ta	22,8 kg
Urnumiy:		257 ta	130,4 kg	

Ushbu bo'limda tanishgan bilimlarimizni mustahkamlash uchun misol tariqasida yuqoridagi jadvalni ko'rib o'taylik. Undagi har bir ustun, satr va katakka alohida e'tibor qaratgan holda ayni jadvalni ifodalovchi HTML kodni yozamiz:

```
<table border="1" cellspacing="0">

  <caption>Mahsulot yuki</caption>
  <col width="48"></col>
  <col width="96" span="2"></col>
  <col width="72" span="2"></col>

  <thead bgcolor="#D0D0D0" align="center">
    <tr>
      <th>Nº</th>
      <th>Tur</th>
      <th>Mahsulot</th>
      <th>Dona</th>
      <th>Vazn</th>
    </tr>
  </thead>

  <tfoot bgcolor="#B2B2B2">
    <tr>
      <td colspan="3" style="text-align:right">
        Umumiy:
      </td>
      <th>257 ta</th>
      <th>130,4 kg</th>
    </tr>
  </tfoot>

  <tbody align="center">
    <tr>
      <td>1.</td>
```

<td rowspan="4" valign="top">Meva</td>	Meva
Olmak	
120 ta	
23,1 kg	

<td>2.</td>	2.
Nok	
60 ta	
13,4 kg	

<td>3.</td>	3.
Behi	
36 ta	
17,6 kg	

<td>4.</td>	4.
Anor	
24 ta	
11,2 kg	

<td>5.</td>	5.
Poliz	
	Qovun
	4 ta

<td>6.</td>	6.
Tarvuz	
3 ta	
19,7 kg	

```

<tr>
  <td>7.</td>
  <td>Handalak</td>
  <td>10 ta</td>
  <td>22,8 kg</td>
</tr>
</tbody>

</table>

```

E'tibor qilish kerakki, «Meva» so'zi yozilgan katakda **rowspan="4"** qiymat berilgan. Bu uning to'rtta odatiy satrga joylashishini anglatadi. Ya'ni keyingi qatorlarda kamida uchta tr qatnashishi kerak va ularda shu katak tagiga to'g'ri keluvchi «td»lar yozilmaydi. Shuning uchun «Meva» yozilgan qatordagi «tr» tegi ichida beshta «td», uning ostidagi «tr»larda esa to'rttadan «td». «Umumiy» so'zi yozilgan «tr» ichida esa bor-yo'g'i uchta «td». Sababi, birinchi katak uchtasining joyini egallayapti.

«thead», «tfoot», «tbody» va «col»larni ishlatish katklarni guruhlarga ajratib, ularga bir xil tus berishda juda qulaylik tug'dirishini ham misolda yaqqol ko'rdik. Yana e'tibor berish kerakki, «tfoot» tegi «table»ning qaysi qismida yozilmasin, undagi satr jadvalning eng quyida akslantiriladi.

5-BO'LIM

MEDIA FAYLLARNI BOG'LASH

5.1. Xaritali rasm qo'yish

HTML boshidanoq matn va rasmni akslantirishga qaratilgan, shuning uchun ilk talqinlaridanoq rasm qo'yish uchun maxsus teg kiritilgan. Agar rasm bir necha qismlardan tashkil topgan xarita sifatida namoyon bo'lsa va undagi har bir qismga bosganda alohida manzilga o'tish yoki amal bajarishga to'g'ri kelsa, avval rasmni maxsus shakllar asosida obyektlarga taqsimlab olish lozim bo'ladi. Buning uchun **map** tegidan foydalaniлади:

```
<map name="nomi"> maydonlar </map>
```

Bunda «nomi» identifikator bo'lib, bitta sahifa uchun takrorlanmas bo'lishi talab qilinadi, «maydonlar» esa «area» tegi yordamida ko'rsatiladi. Rasmning ma'lum bo'lagini ko'rsatuvchi **area** tegi quyidagi alomatlarga ega:

- **alt** — rasm ko'rinxagan holda (ishoratdagi fayl mavjud bo'limganda) akslantiriluvchi muqobil matn.

▪ **shape** — ajratiladigan maydon shakli. Qiymati — aniqlangan tur:

- **rect** — to‘g‘ri to‘rburchak, odatiy qiymat.
- **circle** — doira.
- **default** — odatiy qiymati, rasmni to‘laligicha qamrab oladi.
- **poly** — ko‘pburchak.

▪ **coords** — matematik Dekart koordinatalar tizimi asosida rasmga nisbatan uning ma’lum qismini ajratib olish uchun ko‘rsatiladigan sonlar. Piksel hisobida chap yuqori burchakdan boshlab birinchi son eniga va ikkinchi son bo‘yiga nisbatan hisoblanadi. Agar «shape»da aynala ko‘rsatilgan bo‘lsa, koordinatalar uchta sondan iborat bo‘lib, markazning eni va bo‘yiga nisbatan qayerda joylashganini hamda radiusini anglatadi. Agar «shape»da to‘g‘ri to‘rburchak nazarda tutilgan bo‘lsa, koordinata to‘rtta sondan iborat bo‘lib, chap yuqori va o‘ng quyi burchak qirralari joylashgan o‘rnini ko‘rsatadi. Ko‘pburchakni ifodalash uchun uning qirralaridagi nuqtalari soat mili yo‘nalishiga teskari tartibda joylashishi bo‘yicha koordinatalari ko‘rsatiladi.

▪ **href** — maydonga «bosilganda» qaysi manzilga sahifa yo‘naltirilishini ko‘rsatadi, qiymati: ishorat.

▪ **hreflang** — «href» yozilgan hujjat qaysi tildaligini ko‘rsatadi. Ushbu hujjatni akslantiruvchi brauzer yoki dasturiy ta’milot interfeysi qaysi tilda bo‘lishini aniqlashga yordam beradi.

▪ **nohref** — bu eskirgan teg, «href» o‘z vazifasini bajarmasligini ta’kidlash uchun qo’llanilgan. HTML5da «href» yozilmasa, «nohref» ko‘rsatilgandagidek ishlaydi.

- **target** — «href» yozilgan hujjat qayerga yuklanishi ko'rsatadi. [2.4.]

- **type** — «href» yozilgan hujjatga MIME tur ko'rsatadi. [2.4.]

Rasm qo'yishga mo'ljallangan **img** tegining quyidagi alomatlari mavjud:

- **align** — rasm akslanishi kerak bo'lgan tashqi obyektning maydoniga qanday tarzda joylashishini aniqlaydi. Qiymati aniqlangan tur bo'lib, matn bilan birga kelgan rasmning bir-biriga nisbatan qanday ko'rinishda bo'lishini ham aniqlaydi. Bu eskirgan alomat hisoblansa ham, amalda ko'p qo'llaniladi, shuning uchun qiymatlarini batafsil ko'rib chiqamiz:

- **bottom** — rasmning quyi qismi matnning ostiga tekislanadi.

- **top** — rasmning yuqori qismi matnning tepasiga tekislanadi.

- **left** — rasm maydonning chap qismiga tekislanadi.

- **middle** — rasmning o'rtasi matn satrining o'rta-siga tekislanadi.

- **right** — rasm maydonning o'ng qismiga tekislanadi.

- **alt** — agar ko'rsatilgan manzildagi rasmni akslantirish imkonи bo'lmasa, ushbu alomatdagi matn namoyon bo'ladi. Qiymati: satr.

- **border** — rasm atrofidagi hoshiyaning qalinligi (eskirgan).

- **height** — rasm balandligi.

- **width** — rasm eni.

- **hspace** – rasmning yon tomonlarida qoldiriladigan oraliq.
- **vspace** – rasmning tepa va pastidan qoldiriladigan bo'shliq.
- **src** – rasm faylining manzili, qiymati: ishorat.
- **srcset** – ushbu alomat HTML5da qo'shilgan bo'lib, brauzerga keltirilgan ro'yxatdagi ma'qul rasmni namoyon qilishga imkon yaratadi. Masalan, ayni veb-sahifa kompyuterda ochilganda boshqa rasm, planshetda ochilganda boshqa, telefonda ochilganda boshqa o'ziga mosini undagi brauzer aniqlab ko'rsataveradi. Hozircha ushbu ajoyib imkoniyatni «Android»ning o'z brauzeri va «Internet Explorer» qo'llamasligini ham ta'kidlab o'tish joiz. Bunda ro'yxat vergullar bilan ajratilgan rasmlarning manzil va o'lchamlaridan iborat bo'ladi. O'lcham manzildan bo'sh joy bilan ajratilgan bo'lib, eni necha piksel bo'lishi yoki rasm necha marta kattalashtirilishi lozimligini ko'rsatadi. Agar sondan keyin «w» qo'yilsa, enini, «x» qo'yilsa, necha marta kattalashtirishini anglatadi.
- **sizes** – o'zgaruvchan o'lchamli sahifalar uchun rasmlarning qanday oraliqda kegayishi yoki torayishi mumkinligini aniqlaydi. Tashqi obyektga nisbatan foizda qiymat berish uchun «vw» harflar birikmasidan foydalaniadi, «%» belgisi ishlatilmaydi. Misol: **sizes="100vw"**. Qiymat sifatida hisoblovchi funksiyadan [8.4.10.] ham foydalanish mumkin:

```
<img srcset="a1.jpg 2x,a2.png 150w,a3.gif 200w"
      sizes="(max-width: 300px) 100px,
             (min-width: 50em) 50vw, calc(33vw-100px)">
```

- **usemap** – rasmning bo'laklaridan xarita sifatida foydalanish uchun qo'llaniladi. Yuqorida ko'rib chiqilgan **map** tegi yordamida bog'lanadi:

```
<map name="shakllar">
    <area shape="circle" href="circle.htm"
          coords="320,120,100" title="Doira">
    <area shape="poly" href="poly.htm"
          coords="9,4,13,0,79,0,7,24" title="Maydon">
    <area shape="rect" href="rect.htm"
          coords="60,20,260,420" title="To'rtburchak">
</map>

```

5.2. Tovushli fayl qo'yish

HTML5ning eng katta yutuqlaridan biri tovushli va lavhali fayllarni o'zi tezkor ijro etishni ta'minlashidadir. Hozirda barcha brauzerlarning yuqori talqinlari ushbu tegni to'laqonli namoyon etadi.

Tovushli faylni ijro etish uchun **audio** tegidan foydalaniladi. U quyidagi alomatlarga ega:

- **autoplay** — sahifa to'la yuklanib bo'lishi bilan tovush ijro etilishini ta'minlaydi. Zarurmas qiymat.

- **controls** — ijroni boshqarish (to'xtatish, qayta qo'yish, ovoz balandligini sozlash) tugmalari ko'rinarli bo'lishini ta'minlaydi. Zarurmas qiymat.
- **loop** — ijro etib bo'lingandan so'ng qaytadan boshlaydi, zarurmas qiymat.
- **muted** — ovoz o'chirilgan holda namoyon bo'ladi, zarurmas qiymat.
- **src** — ijro etiladigan fayl manzili, qiymati: ishorat.

Keyingi kitobimizda JavaScript bilan tanishganimizda tovushlarni obyektlar va ular ustida amal bajarilganda mos ravishda yuklash va ijro etilishini ta'minlash xususida batafsil to'xtalamiz. Bunday imkoniyatlar zamona-viy veb-saytlarni yaratishda juda keng qo'llaniladi va ommabop sanaladi.

Mabodo qaysidir brauzer ushbu tegni akslantira olmasa, uning ichidagi yozuvni shundoq ko'rsatadi. Ommabop brauzerlar ichida bundaylari qolmagan, lekin har ehtimolga qarshi faylni ijro etadigan dasturiy ta'minot mavjud bo'lмаган hollar uchun ham ko'chirib olishga imkon beruvchi izoh qoldirish maqsadga muvofiq bo'ladi:

```
<audio autoplay loop src="music.ogg">  
Ushbu faylni ijro etishga imkon bo'lмади  
<a href="/media/music.ogg">  
Marhamat, yuklab oling.  
</a>  
</audio>
```

5.3. Lavhali fayl qo‘yish

Endi zamonaviy brauzerlarda video lavhalarni tomosha qilish uchun qo‘srimcha dasturiy ta’milot yoki «Flash» singari kengaytmalarni yuklash shart emas. HTML5 video ko’rsatishni barcha standart vositalari bilan o‘z zimmasiga oladi. Buning uchun alohida **video** tegi ishlatiladi. U quyidagi alomatlarga ega:

▪ **preload** — sahifa yuklanib bo‘lishi bilan lavha faylini qanday tarzda ko‘chirish lozimligini ko’rsatadi. Qiymati — aniqlangan tur:

- **metadata** — faqatgina bosh ma’lumot qismi (tavsif-xabarlar: lavha hajmi, birinchi kadr, davomiylik vaqtiga va boshqalar) yuklanadi.
- **none** — odatiy qiymat, faylni yuklamaydi.
- **auto** — lavha to’laligicha yuklab olinadi.

▪ **autoplay** — sahifa to’la yuklanib bo‘lishi bilan lavha namoyish etilishini ta’minlaydi. Bu alomat ko’rsatilganda «preload» ishlatilmaydi. Zarurmas qiymat.

▪ **controls** — ijroni boshqarish (to’xtashish, qayta qo‘yish, ovoz balandligini sozlash) tugmalari ko‘rinarli bo‘lishini ta’minlaydi; zarurmas qiymat.

▪ **loop** — ijro yakuniga yetgandan so‘ng qaytadan boshlaydi. Zarurmas qiymat.

▪ **poster** — lavha namoyish etilmasdan tasvir maydonida ko‘rinadigan rasmning manzili, qiymati: ishorat. Agar bunda manzil ko’rsatilmasa yoki ko’rsatilgan rasmni ochish imkonи bo‘lmasa, tasvir maydonida birinchi kadr akslantiriladi.

▪ **src** — ijro etiladigan fayl manzili, qiymati: ishorat.

▪ **height** — lavha namoyon bo‘ladigan maydon balandligi, qiymati: son yoki foiz. Agar qiymat berilmasa, lavha xususiyatlarda ko‘rsatilgan balandlik olinadi, uni ham aniqlash imkonsiz bolsa, «poster»dagi rasm o‘lchamidagi bo‘yi olinadi. Rasm ham mavjud bo‘lmagan-da 150 piksel balandlikda ko‘rsatiladi.

▪ **width** — lavha namoyon bo‘ladigan maydon eni, qiymati: son yoki foiz. Odatiy qiymatini «height»niki singari aniqlab bo‘lmaganada 300 piksel kenglik o‘rnatalidi.

Ushbu tegda tasvirning o‘lchami qat’iy berilmasa, lavhaning asl hajmi qanday bo‘lsa, shunday taqsimot bilan namoyon etiladi. Albatta, bu doim ham samarali natija bermaydi. Eni yoki bo‘yi qat’iy belgilanganda unda ijro etilayotgan lavha balandligi yoki kengligi bo‘yicha unga moslashadi, ammo proporsionallik saqlanib qoladi. Ya’ni kadrlar cho‘zilib yoki aksincha, siqilib qolmaydi.

5.4. Tanlangan faylni yuklash

Albatta, fayl yuklanar ekan, uning hajmiga ham e’tibor berish lozim. Internetda har bir bayt hajm o‘z hisob-kitobiga ega. Shuning uchun media (rasm, lavha, tovush) fayllarning ham hajmi kichik, ammo o‘z sifatini

yo'qotmaganidan foydalanish tavsiya etiladi. Barcha toifada media fayllarni har doim ham hamma brauzerlar qo'llayvermaydi. Ba'zan maxsus kengaytmalar o'rnatishga to'g'ri keladi. Foydalanuvchilarga ma'lum dasturlarni yuklamasang, bizning veb-sahifamizni to'la-qonli ko'ra olmaysan deya dakki berish nojoyiz. Foydalanuvchilar veb-sahifani qanday brauzer orqali ochishmasin, doimo u to'liq imkoniyatini namoyon eta olishi kerak.

Masalan, eng sifatli va ixcham tovushli fayl «ogg» (kengaytmalisi) hisoblanadi. Ammo «Internet Explorer» va «Safari» bunday turdag'i faylni qo'llamaydi. Bu ikki brauzer uchun «AAC» yoki «MP3» shaklida, qolganlar ga hajmi kichik «ogg» ko'rinishidagi faylni yukla deb HTMLning o'zida qo'shimcha tekshiruv shartlarini qo'ymasdan yozish uchun **source** tegidan foydalaniladi. Bu teg yordamida faylning bir necha muqobil manzili keltiriladi. Brauzer o'zi qo'llaydigan birinchi faylni namoyish etadi. Quyidagi alomatlarga ega:

- **media** — qurilma turi. [2.4.]
- **src** — fayl manzili, ishorat.
- **srcset** — ishoratlar ro'yxati. [5.1.]
- **sizes** — o'lchamlar ro'yxati. [5.1.]
- **type** — faylning MIME turi. [2.4.]

Ushbu teg «audio» yoki «video» teglari ichida yozilganda ularning **src** alomatiga qiymat berish shart emas, ichki «source» teglarining «src» alomatlarida ko'rsatilgan ro'yxatdan birinchi o'zi tushungan faylni ijro etaveradi.

Misol:

```
<video width="400" height="300" controls="controls">
  <source src="lavha.ogv"
    type='video/ogg; codecs="theora, vorbis"'>
  <source src="lavha.mp4"
    type='video/mp4; codecs="avc1.42E01E,mp4a.40.2"'>
  <source src="lavha.webm"
    type='video/webm; codecs="vp8, vorbis"'>
</video>
```

Rasmlar ham ba'zan vektor sifatida, ba'zan harakatli «gif» yoki boshqa shaklda beriladigan bo'lsa, hamma brauzerlarning ham to'liq qo'llashiga kafolat yo'q. Bunday hollar ham muqobil fayl sifatida ommabop rasm shaklini «source» tegi yordamida ko'rsatish mumkin. Muammo shundaki, «audio» va «video» teglari HTML5da kiritilgan bo'lib, ularning ochiluvchi va yopiluvchi qismlari mavjud va ichida «source» tegini yoki qolla-maydigan brauzerlarda akslantiradigan matnni yozish imkoniyati yaratilgan. Lekin «img» HTMLning birinchi talqinidayoq kiritilgan va uning yopuvchi qismi mavjud emas. Bu holda qo'shimcha ravishda kiritilgan picture tegidan foydalaniadi:

```
<picture>
  <source srcset="tasvir.svg">
  
</picture>
```

Ayni bir sahifa ekranning turi, o'lchami, tik yoki yotiqligiga qarab turli suratlar ichidan mos rasmni tanlab ko'rsatishda bunday usul yaxshi samara beradi.

Ijro etilayotgan lavha yoki tovushga qo'shimga ravishda izoh matnini ko'rsatish joiz bo'lsa, unda **track** tegidan foydalilanadi. Ayniqsa, kino namoyish bo'layotganda qahramonlarning gaplarini, ayni paytda, yozuv sifatida ko'rsatib turish ko'p hollarda talab etiladi. Bunday imkoniyatni taqdim etish uchun «audio» yoki «video» teglari ichiga «track» tegi yoziladi, u maxsus tavsifnomali matn (subtitr) yozilgan faylga murojaat qiladi. Unda yozilgan satrlarni ko'rsatilgan vaqtin bo'yicha ekranda namoyish etadi.

Ushbu **track** (yo'lakcha) tegi quyidagi alomatlarga ega:

▪ **kind** — yo'lakchaning turini ko'rsatadi, qiymati aniqlangan tur:

- **subtitles** — tavsifnomali matn lavhaning ustiga chiqadi. Bunda har bir aytilayotgan gapning yozuvi o'sha paytda akslantiriladi. Boshqa tilda yozuvni o'qib bormoqchi bo'lganlarga, talaffuz etilayotgan so'zlarning yozuvda ifodalanishini o'rganmoqchi bo'lganlarga yoki qulog'i yaxshi eshitmaydiganlarga ham bu juda qulay vosita hisoblanadi.
- **captions** — bunda to'liq barcha gap yozilmaydi, ayrim qismlarga, aytaylik, boshqa tilda gaplashgan vaqtida yoki qo'shiq kuylanayotganda uning matni ko'rindi, xolos.
- **descriptions** — bu, odatda, ekrandagi voqelikka to'ldiruvchi izoh berilayotganda, ya'ni faqat qo'shimcha fikr yozuv tarzida akslanishi lozim bo'lganda qo'llaniladi.

- **chapters** — lavhaning ayni paytda qaysi mavzuga oid qismi yoki bo'limi ijro etilayotganini ko'rsatib turish uchun ishlataladi.
- **metadata** — bunda matn lavhada ko'rsatilmaydi. Maxsus qo'shimcha kodlar yordamida tahlil qilish uchungina foydalaniladi.
- **src** — faylning manzili. To'liq yoki nisbiy ishorat.
- **srclang** — yo'lakchaning tili. [2.7.]
- **label** — yo'lakchaning nomi, agar u ko'rsatilmasa, sozlashlar panelida «untitledN» sifatida ko'rindi. Shuning uchun bu alomatga satr sifatidagi qiymat berish tavsiya etiladi.
- **default** — zarurmas qiymatga ega bu alomat bitta lavha yoki tovush tegi ichida qatnashgan yo'lakchalarning faqat bittasiga yoziladi va odatda aynan ushbu yo'lak foydalanishini anglatadi.

Ushbu bo'limda olgan bilimlarimizni mustahkamlash uchun bitta mukammalroq misol ko'rib o'tamiz:

```
<video width="500" height="400" controls>
  <track kind="subtitles" src="lavha.en.vtt"
    srclang="en" label="English">
  <track kind="subtitles" src="lavha.uz.vtt"
    srclang="uz" label="Ӯzbekcha" default>
  <track kind="subtitles" src="lavha.ru.vtt"
    srclang="ru" label="Русский">
  <source src="video/lavha.ogv"
    type='video/ogg; codecs="theora, vorbis"'>
  <source src="video/lavha.mp4"
    type='video/mp4; codecs="avc1.42E01E,mp4a.40.2"'>
```

```
<source src="video/lavha.webm"
        type='video/webm; codecs="vp8, vorbis"'>
<video> tegi ishlamadi, HTML5ni qo'llaydigan
ommaviy brauzerdan foydalaning!
</video>
```

5.5. Nostandard media qo'yish

Ba'zan shunday media fayllarni akslantirishga to'g'ri keladiki, ularning formatini ommabop brauzerlar tu-shunmaydi. Namoyish etish uchun «qo'shilgich» deb ataluvchi (plugin) maxsus qo'shimcha dasturlardan foy-dalanishga to'g'ri keladi. Yoki alohida dasturiy kodlar yozish talab etiladi. Brauzer qo'llaydigan dasturiy kodlarni «**skript**» deb yuritamiz. Ular maxsus **script** tegi ichida yoziladi. Keyingi kitobda aynan «JavaScript» bilan batafsil tanishamiz. Hozircha ushbu kitobda bayon etilgan kodlarni namoyish etish uchun skriptni qo'llamaydigan yoki sozlashlar bo'limidan bunday imkoniyati o'chirilgan brauzerlar ham kifoya qiladi. Skript brauzerde qo'llanilishini tekshirish uchun **noscript** tegi ishlatiladi. Agar uning ichida yozilgan matn veb-sahifada akslantirilsa, demak, brauzerde yozilgan skriptlar ishlamaydi.

Brauzerlar va ularga qo'shilgichlar qandayligiga qarab yuklanayotgan media faylning ko'rinishi turlicha

bo'lishi mumkin. Ular **embed** tegi yordamida biriktiriladi va quyidagi alomatlar yormadida sozlash mumkin:

- **height** – obyekt balandligi, qiymati: o'lcham;
- **hidden** – obyektning ko'rinishmasligini ta'minlaydi, mantiqiy qiymat;
- **pluginspage** – «qo'shilgich»ni qayerdan yuklab olishni ko'rsatadi, qiymati: ishorat;
- **src** – fayl manzili, qiymati: ishorat;
- **type** – fayl MIME-turi;
- **width** – obyektning eni, qiymati: o'lcham.

Misol:

```
<embed width="400" height="300"
       type="application/x-shockwave-flash"
       src="/flash/anim.swf"
       pluginspage="https://get.adobe.com/flashplayer">
```

5.6. Obyektlarni biriktirish

Media fayl qo'yilganda unga yoziladigan izohlar va qo'shimcha teglar orqali bog'langan elementlar yagona obyekt sifatida mujassamlashtirib akslantirilishi lozim bo'ladi. Ya'ni rasm yoki boshqa media faylni bir joydan boshqa yerga ko'chirishga to'g'ri kelsa, uni tavsiflovchi matn va boshqa elementlar umumiy ko'rinishi o'zgar-magan holda birgalikda siljishi kerak. Shunday vaziyat-

larda bir necha teglar orqali yaratilgan elementlarni biriktirish uchun ularni HTML5da yangi kiritilgan **figure** tegi ichida yozish talab etiladi. Ushbu majmuaga izoh **figcaption** tegi yordamida qo'yiladi, faqat u «figure» tegi ichidagi birinchi yoki oxirgi element bo'lishi shart. Misol:

```
<figure>
  <p>
    
  </p>
  <hr>
  <audio autoplay loop src="kuy.mp3"></audio>
  <figcaption> Eski shahar </figcaption>
</figure>
```

5.7. Tashqi HTML faylni akslantirish

Butunlay boshqa, o'ziga xos «JavaScript» va «CSS» kodlariga ega tashqi HTML faylni sahifaning bir qismida namoyon etish uchun **iframe** tegidan foydalaniladi. Imkonи boricha uni ishlatmaslik tavsiya etiladi, ammo ba'zan boshqa manbadagi ma'lumotlarni (mualliflik huquqini buzmagan holda) birgalikda akslantirishga to'g'ri kelgan vaziyatda freym qo'llashga majbur bo'linadi. Ushbu teg quyidagi alomatlarga ega:

- **name** — freym nomi;

- **allowfullscreen** – freymning to'liq ekranda aks-lanishini ta'minlaydi (zarurmas qiymat);
- **allowtransparency** – freymning ort rangini shaf-sof qiladi (zarurmas qiymat);
- **frameborder** – freym atrofida chegaraviy chiziq bo'lishini aniqlaydi (mantiqiy qiymat);
- **height** – freym balandligi;
- **marginheight** – freym tepa va pastidagi hoshiya qalinligi (o'lcham);
- **marginwidth** – freymning yon tomonlaridagi hoshiya qalinligi (o'lcham);
- **sandbox** – freymdagi sahifaga cheklov o'rnatadi. Masalan, siz ijtimoiy tarmoqning qaysidir imkoniyatini saytingizga ulash uchun freym orqali bog'ladingiz, lekin ma'lum bo'lishicha, siz ulashishni istamaydigan ma'lumotlarni ham olishga uning imkoniyati mavjud. Shunday hollarda cheklovlar qo'yishga ehtiyoj seziladi. HTML5da qo'shilgan ushbu alomatga qiymat ko'rsatilmagan taqdirda barcha cheklovlar o'rnatilgan deb hisoblanadi. Quyidagi qiymatlarni ko'rsatish bilan (bir nechtasini bo'sh joy bilan ajratib berish mumkin) ma'lum ruxsatlar beriladi:

- **allow-forms** – forma yordamida qiymatlarni yuborishga;
- **allow-modals** – freym ichidan modal oyna [10.1.] ochishga;
- **allow-orientation-lock** – ekranni qulflashni taqiqlashga;
- **allow-scripts** – suzib chiquvchi oyna yaratishni cheklagan holda skriptlarning ishlashiga;

- **allow-pointer-lock** – Pointer Lock API dan (sichqoncha harakatlarini nazorat qiluvchi skriptdan) foydalanishga;
 - **allow-popups** – suzib chiquvchi («open», «target=_blank»), «showModalDialog»lar yordamida yaratilgan) oyna ko'rinishiga;
 - **allow-presentation** – Presentation API (tashqi monitorlarni qo'llovchi skript)ni ishlatalishga;
 - **allow-same-origin** – suzib chiquvchi oynani taqiqlab, faqat tashqi sahifa manbasidan olin-gan ishorat bo'yicha yuklashga;
 - **allow-top-navigation** – ishoratlarni tashqi sahifada ochishga;
- **scrolling** – surgich qo'yilishini aniqlaydi. Qiymati: «auto», «yes», «no»lardan biri.
 - **seamless** – freymni tashqi sahifaning qismi sifatida akslantirish (zarurmas qiymat).
 - **src** – freymga yuklanuvchi sahifa manzili, ishorat.
 - **screddoc** – HTML kod. Sahifa manzilini ko'rsatish o'rniiga unda bo'lishi lozim bo'lgan kodni ushbu alomat orqali berish mumkin. Ushbu alomat HTML5da qo'shilgan va uni «IE» va «Edge»lar qo'llamasligi mumkin. Agar kod yozilsa, «src» alomatidagi manzil e'tiborsiz qoldirilib, ushbu kod namoyish etiladi.
 - **width** – freymning eni.

Misol:

```
<iframe scrolling="no" width="200" height="100"  
frameborder="1" screddoc="<!DOCTYPE html>  
    <div> Bu oddiy DIV emas </div>">  
</iframe>
```

6-BO'LIM

GRAFIK VA TASVIRLAR CHIZISH

6.1. CANVAS tegi

HTML5 imkoniyatlaridan foydalanib brauzerda istalgan tasvirni chizish mumkin. Tayyor suratdan foydalanishdan rasm chizishning qulayligi brauzerdagi harakatga mutanosib ravishda chiziladigan tasvirning o'zgarib borishidir. Rasm chiziladigan maydon **canvas** tegi bilan belgilanadi. Unga tasvir maxsus skriptlar yordamida chiziladi. Keyingi «JavaScript»ga bag'ishlangan kitobimizda suratlar chizish, ularni o'zimiz istaganday harakatga keltirish haqida batafsil to'xtalib o'tamiz. Hozircha uning blokli teg ekanligini va nima uchun qo'llanilishini bilishimiz kifoya. Uning bor-yo'g'i ikkita o'ziga xos alomati mavjud:

- **height** – balandligi, odatiy qiymati 150 px.
- **width** – kengligi, odatiy qiymati 300 px.

Tayyor rasmni qo'ymasdan yoki maxsus skriptlarni ishlatmasdan HTML5da tasvir hosil qilish mumkin emasmi degan haqli savol tug'ilishi tabiiy. Albatta, mumkin.

6.2. SVG tegi

SVG (Scalable Vector Graphics) XML shaklidagi vertorli o'zgaruvchan grafikni tasvirlash uchun qo'llaniladi. U qolip sifatida **canvas** tegiga o'xshaydi, farqi shundaki, uning ichida maxsus teglar yordamida shakllar chizish mumkin. SVG boshqa sohalarda ham keng ishlataladigan standart bo'lib, eng muhim jihatiga rasmning hajmi o'zgarganda ham sifati yuqoriligidicha qolaveradi. Sababi, unda XML shaklida yozilgan kodlar tahlil etilib, brauzerning o'zi tasvirni hosil qiladi. SVG tegining ichida geometrik shakllar chizishga mo'ljallangan maxsus teglar ishlataladi.

Umuman olganda, SVGga alohida XML til sifatida qarash mumkin, uning ichida ko'pgina HTML teglar o'ziga xos tarzda qo'llaniladi. Biz uning asosini o'rganib, faqat u yordamida ikki o'lchamli Dekart koordinatalar tizimida standart shakllarning maxsus xususiyatlarni belgilagan holda chizish va harakatlashtirishni ko'rib chiqamiz. SVG tegi quyidagi alomatlarga ega:

- **version** — talqini, haqiqiy son. Qaysi qoidalar asosida chizish kerakligini ko'rsatadi.

- **baseProfile** — SVG qonuniyatlar majmuasining ixcham jamlanmasini ko'rsatadi, bu tasvirning to'g'ri akslanishini ta'minlash uchun zarur. Doim ham barcha qonuniyatlarni to'laligicha yuklab olib foydalanish yaxshi natija beravermaydi. Qiymati – aniqlangan tur:

baseProfile = "none" | "full" | "basic" | "tiny"

Uyali qurilmalar uchun o'zak (basic) yoki ixcham (tiny) majmuani tanlash tavsiya etiladi.

- **width** – ichki maydonining eni. Qiymati: son yoki foiz. Agar unga **0** berilsa, undagi tasvirlar chizilmaydi.
- **height** – ichki maydonining bo'yisi. Xusisiyati eni-niki singari.
- **viewbox** – maydonning ichiga shaklning qanday masshtabda joylanishini aniqlaydi. Agar ushbu alomat ko'rsatilmasa yoki unga «none» qiymat berilsa, to'la maydonda o'lchami o'zgarmagan shakl akslantiriladi. Qiymati satr bo'lib, unda bo'sh joy yoki vergul bilan ajratilgan to'rtta butun son qatnashishi kerak. Mos ravishda «X» va «Y» bo'yicha boshlang'ich chegara (ya'ni tasvirning shuncha nuqtagacha bo'lgan qismi kesib yuboriladi) hamda eni va bo'yisi bo'yicha o'lchamini o'zgartirish foizi.

▪ **preserveAspectRatio** – masshtabni o'zgartirgan holda moslashtirish. Tasvirning o'lchami «viewbox»da ko'rsatilgan qolip bo'yicha maydonga biz xohlaganday joylashmagan taqdirda uning joyini va hajmini o'zgartirish uchun ushbu alomatdan foydalilaniladi. Qiymati satr, uchta qismdan iborat. Agar SVGning ichida tasvir mavjud bo'lsa, joylashtirishni aynan uning masshabiga qarab amalga oshirish uchun birinchi so'zning o'rniga «defer» yozilishi mumkin. Odatda, birinchi so'z ishlatilmaydi va ikkitasi bilan kifoyalaniladi. Asosiy qiymatda tasvir maydonining (to'g'ri to'rtburchak shaklida) bir uchi yoki markazi qayerda turishi ko'rsatiladi va quydagicha shablonda yoziladi:

X[Min | Mid | Max] Y[Min | Mid | Max] mosligi

Bunda «mosligi» quydagi qiymatlardan birini qabul qilishi mumkin:

- **meet** – proporsiya saqlanib, tasvir to'laligicha joylanadi, odatiy qiymat.
- **slice** – proporsiyani saqlagan holda maydonni to'ldirishga urinadi, tasvir tik yoki yotiq kesilishi mumkin.
- **none** – proporsiyani saqlamay maydonni to'l-diradi (tasvir buziladi).

Misol:

```
<svg viewBox="0 0 300 300"
      preserveAspectRatio="XMinYMax meet">
  . . .
</svg>
```

- **snapshotTime** – harakatlanuvchi suratlarda qaysi qadamdagи tasvirni akslantirish lozimligini ko'rsatadi, «animateColor» singari teglar ishlatilganda foydalaniadi. Afsus, hozircha brauzerlar buni qo'llamaydi.
- **playbackOrder** – o'z ichidagi elementlarga murojaat qilishda ularni izlab topish yo'nalishini ko'rsatadi. Qiymati aniqlangan tur:
 - **forwardOnly** – ketma-ketlikda faqat oldinga qarab izlaydi.
 - **all** – ikkala yo'nalishda ham izlashni ta'minlaydi.
- **timelineBegin** – harakatli tasvir bo'lsa, u qachon namoyish bo'lishini ko'rsatadi. Qiymati aniqlangan tur:
 - **onLoad** – sahifa yuklanishi bilan, odatiy qiymat;
 - **onStart** – ko'rsatilgan vaqtida;

- **contentScriptType** — sahifaning MIME [2.4.] turi. Qo'shimcha skriptdan foydalanilgan bo'lsa, ko'rsatilgan faylning tavsif turi.

- **zoomAndPan** — foydalanuvchiga tasvirning hajmini o'zgartirish imkonini (masalan, lupa yordamida) yaratadi. Qiymati: **magnify** yoki **disabled**.

- **focusable** — fokuslash (tugmalar bilan boshqarish uchun). Qiymati: aniqlangan tur:

- **true** — fokus faol bo'lishi mumkin, «Tab» tugmasida boshqariladi.
- **false** — fokus faollashmaydi.
- **auto** — agar ichida fokuslanuvchi element bo'lmasa, faollashmaydi.

Ushbu texnologiya ochiq kodda bo'lgani bois ham keng qo'llaniladi. Agar yangi talqinlari chiqib, tatbiq etiladigan bo'lsa, «version» alomatida u ko'rsatib qo'yilaveriladi. Avvallari bunday imkoniyatlar bo'lмаганлиги sababli «Flesh» kabi qo'shimcha dasturiy ta'minotdan foydalanishga ehtiyoj tug'ilardi, u esa tasvirlarni paydo qilishda o'zi patentlagan yopiq koddan foydalanardi. Flesh ishlashi uchun esa maxsus dastur operatsion tizimga o'rnatilishi yoki brauzer uchun kengaytma yuklanishi talab etilardi. Mutaxasislar bunday qo'shimcha dastur bilan tasvirlar hosil qilishda kompyuterlarning ko'p resurslari sarflanishini, ayniqsa, elektr energiyasi ortiqcha isrof bo'lishini hamda texnik qurilmalar zo'rqlikini isbotlab, HTML5da butunlay fleshdan voz kechishni ma'qul topishgan. Fleshning noyob imkoniyatlari lavha va tovushlarni ijro etish hamda harakatli tasvirlarni paydo etish edi. Ko'rib o'tganimizday, hozir-

da HTML5da bu imkoniyatlarning barchasi mavjud. Ular ochiq kodli, hamma uchun mo'ljallangan, ortiqcha dasturiy ta'minot va texnik imkoniyatlar talab qilmaydi.

6.3. Geometrik shakllar qo'yish

SVG tegining ichida bir necha xil geometrik shakllar hosil qilish mumkin. Ular o'ziga xos teglar yordamida yozilib, quyidagi umumiy alomatlarga ega:

- **stroke** — chegaralovchi chiziq rangi, qiymati: rang.
- **stroke-width** — chegaralovchi chiziq qalinligi, piksel hisobida, qiymati: son.
- **stroke-linecap** — chiziq qalinlashtirilganda yo'g'onligi asl chiziqqa nisbatan qanday kengayishini aniqlaydi. Qiymati — aniqlangan tur:
 - **butt** — ikki yonigagina bir xilda enlik qo'shiladi.
 - **square** — har tarafga birday kengayib boradi.
 - **round** — chiziqning uchlari yoysimon tarzda «qirrasizlantiriladi».
- **stroke-linejoin** — chiziqlar tutashganda hosil boladigan burchakning «o'tkirligi»ni aniqlaydi. Qiymati — aniqlangan tur:
 - **miter** — uchli (o'tkir).
 - **bevel** — qirrali (o'tkir uchi kesilgandagidek ikita qirrasi bor).
 - **round** — yoysimon (o'tmas).

- **stroke-miterlimit** – «stroke-linejoin="miter"» qiymat berilgan bo'lsa, ushbu alomatda ko'rsatilgan son chiziqlarning ichki tutashgan nuqtasi bilan chetki chegarasi orasidagi masofa eng ko'pi bilan qancha bo'lishini belgilaydi.

- **stroke-dasharray** – chegaralovchi chiziq uzilishlari. Qiymiti: satr. Unda bir necha sonlar bo'sh joy bilan ajratib yoziladi va ular mos ravishda takroriy ketma-ket keluvchi chiziqchalar yoki ular orasidagi uzilishlarning uzunligini aniqlaydi.

- **stroke-dashoffset** – «stroke-dasharray»da ko'r-satilgan bиринчи son chegaraviy chiziqning boshlang'ich chiziqchasi uzunligini anglatadi. Agar bиринчи chiziqcha qisqa, qolganlari ko'rsatilgan bo'yicha akslanishi kerak bo'lsa, ushbu alomat yordamida boshlang'ich chiziq-chani kesiladi. Qiymati: «stroke-dasharray»da ko'rsa-tilgan bиринчи sondan kichkina butun son.

- **stroke-opacity** – chegaraviy chiziqning shaffofligi. Qiymati 0 va 1 orasidagi haqiqiy son. 0 – ko'rmas darajada tiniq, 1 – o'z rangida (shaffofligi yo'q).

- **fill** — shaklning ichki rangi, qiymati: rang.

- **fill-opacity** — shakl ichining shaffofligi. Qiymati «stroke-opacity»nikiday.

Ta'kidlab o'tish joizki, SVG tegi XML tiliga mo'ljal-langanligi uchun, uning ichida qo'llanilgan shakl chizuvchi teglarning yopuvchi qismi mavjud bo'lmasa, ularni «/» bilan tugatish talab etiladi. Aks holda, xatolik yuzaga kelib, keyingi shakl akslanmasligi mumkin.

6.3.1. To‘g‘ri to‘rtburchak

To‘g‘ri to‘rtburchak **rect** tegi yordamida qo‘yiladi. U quyidagi alomatlarga ega:

- **x** – chap yuqori nuqtaning yotiq koordinatasi.
- **y** – chap yuqori nuqtaning tik koordinatasi.
- **width** – to‘g‘ri to‘rtburchak eni.
- **height** – to‘g‘ri to‘rtburchak bo‘yi.
- **rx** – qirrasining x o‘qi bo‘yicha egilish radiusi.
- **ry** – qirrasining y o‘qi bo‘yicha egilish radiusi.

Misol:

```
<svg width="120" height="120" viewBox="0 0 120 120">
    <rect x="6" y="8" width="92" height="98"
          rx="8" ry="8" fill="aqua" stroke="red" />
</svg>
```

6.3.2. Doira

Doira **circle** tegi yordamida chiziladi. U quyidagi alomatlarga ega:

- **cx** – chap yuqori nuqtaning yotiq koordinatasi.
- **cy** – chap yuqori nuqtaning tik koordinatasi.
- **r** – x o‘qi bo‘yicha qirrasining egilish radiusi.

Misol:

```
<svg width="120" height="120" viewBox="0 0 120 120">
    <circle cx="60" cy="60" r="48" fill="aqua"
           stroke="blue" />
</svg>
```

6.3.3. Ellips

Ellips (qovunsifat shakl) **ellipse** tegi yordamida chiziladi. U quyidagi alomatlarga ega:

- **cx** – markazining yotiq koordinatasi.
- **cy** – markazining tik koordinatasi.
- **rx** – x o'qi bo'yicha radiusi.
- **ry** – y o'qi bo'yicha radiusi.

Misol:

```
<svg width="120" height="120" viewBox="0 0 120 120">
    <ellipse cx="60" cy="60" rx="60" ry="20"
              fill="aqua" stroke="blue" />
</svg>
```

6.3.4. Kesma

Kesma **line** tegi yordamida chiziladi. U quyidagi alomatlarga ega:

- **x1** – kesma boshining x o'qi bo'yicha koordinatasi.
- **y1** – kesma boshining y o'qi bo'yicha koordinatasi.
- **x2** – kesma oxirining x o'qi bo'yicha koordinatasi.
- **y2** – kesma oxirining y o'qi bo'yicha koordinatasi.

```
<svg width="120" height="120" viewBox="0 0 120 120">
    <line x1="1" y1="1" x2="119" y2="119"
           stroke="blue" />
    <line x1="20" y1="100" x2="100" y2="20"
           stroke="blue" />
</svg>
```

6.3.5. Siniq chiziq

Siniq chiziq **polyline** tegi yordamida chiziladi. Uning **points** alomatida koordinatalar juftligi vergul bilan, nuqtalar esa bo'sh joy bilan ajratib satr sifatida beriladi. Misol:

```
<svg width="80" height="80" viewBox="0 0 80 80">
  <polyline points="40,0 80,80 0,40 40,0"
             fill="none" stroke="red" />
</svg>
```

6.3.6. Ko'pburchak

Agar siniq chiziq boshlang'ich nuqtasining koordinatasi berilgan satr oxiriga ham qo'shilsa, yopiq ko'pburchak shakli hosil bo'ladi. Siniq chiziq bilan ko'pburchak ifodalanadigan bo'lsa, tomonlari kesishib qolishiga ham imkon tug'ilishi mumkin, bunga yo'l qo'ymaslik maqsadida ko'pburchak chizish uchun maxsus **polygon** tegi qo'llaniladi. Misol:

```
<svg width="80" height="80" viewBox="0 0 80 80">
  <polygon points="40,0 80,40 0,80"
            fill="aqua" stroke="red" />
</svg>
```

6.4. Murakkab shakllar chizish

Murakkab shakllarni yagona teg yoki bir necha teglar majmuasi yordamida akslantirish mumkin. Dastlab bitta mukammal tegning imkoniyati bilan tanishib chiqamiz.

6.4.1. PATH tegi

Path tegi kesmalar, yoqlar va egri chiziqlar yordamida hosil qilinadigan murakkab shakl chizishga imkon beradi. Uning **d** alomati satr qiymatiga ega bo'lib, murakkab kelishuvlar asosida shakl chizish qonuniyatini aniqlaydi. Qanday qonuniyatlar asosida chiziqni boshqarib shakl hosil qilishni bat afsil ko'rib chiqamiz. Buni rassomning mo'yqalamini yuritishga qiyoslash mumkin. Qiymat sifatida **d** alomatiga beriladigan satrda quyidagi buyruqlar qatnashishi mumkin:

- **moveto** — boshlang'ich nuqta. Qisqacha qilib «M» yoki «m» harfi bilan belgilash mumkin. Tabiiyki, ko'rsatilgan harfdan keyin vergul bilan ajratilgan holda ikkita son – **x** va **y** bo'yicha koordinata yoziladi. Bu mo'yqalamni dastavval qaysi nuqtaga tekizish kerakligini ko'rsatadi.

- **lineto** — joriy o'rindan qaysi nuqtagacha kesma chizish kerakligini ko'rsatadi. Sintaksisi «moveto»niki singari.

- **horizontal** — joriy nuqtadan boshlab yotiq chiziq

chizish uchun **H** yoki **h** buyruqlari ishlataladi. Harfdan so'ng bitta **x** bo'yicha koordinata ko'rsatiladi.

▪ **vertical** — joriy nuqtadan tik chiziq chizish uchun **V** yoki **v** buyruqlari qo'llaniladi. Harfdan so'ng bitta **y** bo'yicha koordinata ko'rsatiladi.

▪ **curveto** — joriy nuqta va **S** yoki **s** harflaridan keyin ko'rsatilgan uchta koordinata bo'yicha hosil qilingan to'rburchak shaklidagi siniq chiziq asosida chiziladigan «Bezier» yoyi (1-rasm).

▪ **smooth** — bu ham «curveto» singari yoy chizadi, faqat unda **S** yoki **s** harfidan keyin ikkita nuqta beriladi, uchinchisi esa birinchi nuqtaning ikkinchisiga nisbatan simmetrik aksi sifatida qaraladi.

▪ **quadratic** — joriy nuqta va **Q** yoki **q** harflaridan keyin ko'rsatilgan ikki koordinatadan o'tadigan burchak asosida chiziladigan «Bezier» yoyi.

▪ **smooth quadratic** — joriy nuqta va **T** yoki **t** dan keyin yoziladigan bitta koordinata asosida hosil qilingan kvadrat bo'yicha chiziladigan «Bezier» yoyi.

▪ **elliptic** — ellips yoyi, bunda **A** yoki **a** harfidan keyin **x** va **y** o'qi bo'yicha radiuslar, keyin yoyning tugash nuqtasi beriladi.

▪ **closepath** — yopish, **Z** yoki **z** harfi bilan buyruq beriladi, bunda joriy (oxirgi) nuqta kesma yordamida boshlang'ich nuqtaga biriktiriladi.

Sanab o'tilgan qonuniyatlarda e'tibor qaratish kerak bo'lgan jihat — buyruqlar bitta harf orqali beriladi. Agar bosh harf ishlatsa, tasvir chizish uchun ajratilgan umumiy maydon bo'yicha Dekart koordinatalar tizimi-

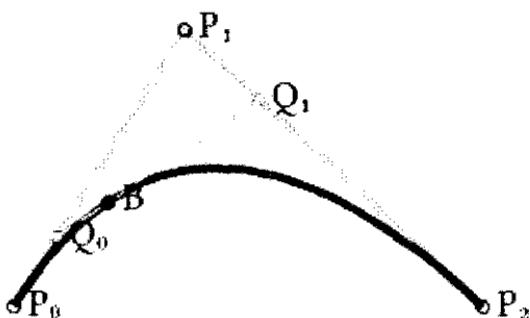
dagi **x** va **y** o'qi bo'yicha mutloq koordinata ko'rsatiladi. Kichik harf ishlatilsa, mazkur (oldingi) nuqtaga nisbatan **x** va **y** o'qi bo'yicha qancha masofaga surilishi lozimligini anglatadi. Sonlar bir-biridan bo'shliq yoki vergul bilan ajratiladi. Agar son manfiy bo'lsa, minusdan boshqa ajratuvchi ishlatish shart emas. Shunday ekan, umuman bo'shliq va vergul ishlatmasdan koordinatalarni faqat «+» va «-»lar yordamida ham ifodalash mumkin.

Kimki «Bezier» yoyi qandayligi haqida tasavvurga ega bo'lmasa, geometriyadan shu mavzu tanishib olish bilan tavsiya etiladi. U murakkab formula yordamida aniqlanadi, nuqtalar qancha ko'p ko'rsatilsa, ular yordamida hosil qilinadigan siniq chiziqqa qarab yoy chizish shuncha mushkullashadi. 1962-yili Pierre Bezier uni om-maga ma'lum qilgan bo'lsa-da, undan bir necha yillar ilgari Kasteljo tomonidan ham avtomobil kuzovlari uchun ushbu usul qo'llanilgan.

Shu o'rinda eng sodda, faqatgina uchta nuqtadan iborat siniq chiziq (burchak) uchun «Bezier» yoyi qanday chizilishi haqida tushuncha berib o'tishni lozim topdik. Bunda burchakning har ikki kesmasini teng (aytaylik, **n** ta) bo'lakka bo'linadi. Keyin shunday vaqtinchalik kesma chiziladiki, uning boshlang'ich uchi birinchi kesmaning bo'laklarga ajratuvchi ochiq uchiga nisbatan birinchi nuqtasi bo'ladi, ikkinchi uchi esa burchakni hosil qilgan ikkinchi kesmaning uchiga nisbatan birinchi nuqtasi bo'ladi. Vaqtinchalik kesmani ham **n** ta bo'lakka bo'lib, birinchi nuqtasi maxsus belgilab qo'yiladi. Keyin vaqtinchalik kesma o'chirilib, yangisi xuddi shu tariqa faqat ikkinchi nuqtalar asosida chizi-

ladi. Shu tariqa vaqtinchalik kesmalar qoldirgan n ta maxsus nuqtalar bitta yoyni hosil qiladi. Albatta, bunda **n** yetarlicha katta natural son bo'lishi kerak.

Rasm 1. Bezier yoyini hosil qilish



Demak, **path** tegining **d** alomatida kesma va yollar chizish orqali ixtiyoriy ikki o'lchamli shakl hosil qila olamiz. U shunchalik mukammal ekan, butun boshli tasvirni faqat bitta path tegi bilan chizilsa bo'ladimi degan savol tug'ilishi mumkin. Bo'ladi, agar shakl faqat bitta rangdan iborat bo'lishi qanoatlantirilsa. Odatda, bir necha ranglardan foydalanishga to'g'ri keladi, buning uchun bir nechta teg ishlataladi va ranglar bo'lim boshida ifoda etilgandagidek [6.3.] qo'llaniladi.

Quyida bir juft gilos qanday chizilishining kodini keltirib o'tamiz:

```
<svg width="215" height="274" viewBox="0 0 215 274">
  <path fill="none" stroke="#7AA20D"
    stroke-width="6" stroke-linecap="round"
    d="M55,174c0,0,77-73,82-104 c4-30,46,91,27,121"
    stroke-linejoin="round" />
```

```

<path fill="#7AA20D" stroke="#7AA20D"
      stroke-width="4" stroke-linejoin="round"
      stroke-linecap="round" d="M134,62c-1-6,0-12,5-19
                                c11-14,23-17,40-17c0,0,22-0,26-20c-2,12-5,24-9,36
                                c-6,19-22,48-47,38 C141,75,136,69,134,62z" />
<path fill="#ED2246" stroke="#ED0046"
      stroke-width="2" d="M164,193 c1-0,3-0,
                                5-1c8-2,21-6,28,3c15,21,5,55-18,66
                                c-27,12-58-16-56-45 c1-13, 8-35,26-27
                                C155,191,159,193,164,193z" />
<path fill="#ED2222" stroke="#CB1111"
      stroke-width="2" d="M55,176 c1+0+3+0+5-0c8
                                0,22-2,26,9c9,24-6,55-32,60 c-29,5-53-29-44-56c4-13,
                                15-33, 31-20C47,173,50,176,55,176z" />
</svg>

```

6.4.2. Markerdan foydalanish

Ko'pincha ayni bir tasvirni takroriy chizmaslik uchun uni ma'lum nom bilan saqlab qo'yish lozim bo'ladi. Keyin undan bir necha bor foydalanish mumkin. Buning uchun **marker** tegi ishlataladi. U quyidagi alomatlarga ega:

- **viewBox** — markerni surish yoki mashtabini o'zgartirish [6.2.];
- **markerWidth** — marker maydonining eni;
- **markerHeight** — marker maydonining bo'yisi;
- **refX** — markerni x o'qi bo'yicha surish (musbat son berilsa, chapga);

- **refY** — markerni y o‘qi bo‘yicha surish (musbat son berilsa, tepaga);
- **markerUnits** — chegaraviy chiziq qalinligiga qarab markerning mashtabini aniqlaydi. Odatiy qiymati «stroke-width»da berilgani kabi;
- **orient** — yotiq chiziqqaga nisbatan markerni burish burchagi. Qiymati son yoki «auto» (90^0);

```

<svg>
  <marker id="#mCircle" refX="4" refY="4"
          markerWidth="8" markerHeight="8">
    <circle cx="4" cy="4" r="2"
            style="stroke: none; fill: #000000;" />
  </marker>
  <marker id="#mArrow" refX="4" refY="2"
          markerWidth="8" markerHeight="8">
    <path fill="#000000" d="M2+2L6+2L4+6z" />
  </marker>
  <path d="M10+20L40+20S90+20,90+90" fill="none"
        stroke="#6666ff" stroke-width="4"
        marker-start="url(#mCircle)"
        marker-end="url(#mArrow)" />
</svg>

```

Bitta **marker** ichida bir necha teglar yordamida maxsus shakl hosil qilish mumkin. Qaysi shaklga marker ulash lozim bo‘lsa, unga mos ravishda **marker-start**, **marker-mid** yoki **marker-end** alomatlari yordamida avval e’lon qilingan nomdagisi biriktiriladi. Ushbu alomatlaridan biri yordamida marker chaqirilmasa, undagi tasvir

akslantirilmaydi. Markerning ichidagi teglar yagona tasvir sifatida guruhlangan hisoblanadi.

Umuman olganda, murakkab tasvirlarni bir necha teglar yordamida chizib, **g** tegi yordamida guruhlash mumkin.

6.4.3. Tasvirni o'zgacha shakllantirish

Har bir tasvirni yoki ularning guruhini shakllantirishda ma'lum qoida bo'yicha o'zgarishini ta'minlash mumkin. Masalan: ularni burish, hajmini o'zgartirish, teskari yoki qarama-qarshi akslantirish. Ushbu maqsadda barcha shakl hosil qiluvchi yoki ularni guruhlovchi teglarda **transform** alomatini qo'llasa bo'ladi. U 3×3 matritsa yordamida koordinatalarni qayta ishlab hosil qiladi. Quyidagi qiymatlarni qabul qiladi:

- **matrix(<a> <c> <d> <e> <f>)** — shu oltita songa yana 0, 0, 1 ni qo'shib 3×3 matritsa hosil qiladi va unga koordinatani matritsa sifatida ko'paytirib, yangi koordinatalar oladi.
- **translate(<x> [<y>])** — x va y o'qlari bo'yicha surish. Qiymat berilmasa, 0 deb oladi.
- **scale(<x> [<y>])** — x va y o'qlari bo'yicha massh tablash. Agar y berilmasa, x ning qiymatiga teng hisoblanadi.
- **rotate(<a> [<x> <y>])** — ko'rsatilgan (x, y) koordinata (*berilmasa, (0, 0) olinadi*) markaz sifatida hisoblanib, unga nisbatan a gradusga buriladi.
- **skewX(<a>)** — x o'qi bo'yicha a gradusga og'dirish.

- **skewY(<a>)** — y o'qi bo'yicha a gradusga og'dirish.

Misol:

```
<svg width="180" height="200">
  <text x="10" y="60"
    transform="translate(36) rotate(45 30 30)">
      Burilgan matn
    </text>
</svg>
```

6.4.4. Tuslantirish

Shakl rangini tuslantirish uchun **linearGradient** yoki **radialGradient** teglaridan foydalaniladi. Birinchisida to'g'ri chiziq bo'yicha bitta rangdan ikkinchisiga asta o'tib borsa, ikkinchisi markazdan doira tarzda rangning o'zgarib borishini ta'minlaydi. Ular quyidagi alomatlarga ega:

- **x1, y1, x2, y2** — mos ravishda boshlang'ich va oxirgi nuqtalar koordinatalari, birinchi rang qayerdan boshlanib, ikkinchi rang qayerda tugashini ko'rsatadi. Qiymati: son yoki foiz. Ularning qiymati yordamida chiziqli tuslagichning yo'nalishini ham aniqlash mumkin. (radialGradient tegida x1, y1 o'rniga fx, fy; x2, y2 o'rniga cx, cy va radius r ishlataladi.)

- **gradientUnits** — berilgan koordinatalarning qanday qabul qilinishini ko'rsatadi. Aniqlangan turdag'i quyidagi qiymatlarni qabul qiladi:

- **userSpaceOnUse** – koordinatalar (x_1, y_1, x_2, y_2) umumiy SVG obyektga nisbatan;
 - **objectBoundingBox** – (odatiy qiymat) ayni tegishli shaklga nisbatan;
- **spreadMethod** – tuslanuvchi ranglar maydoni shaklning hajmidan kichik bo'lgan taqdirda, bo'sh qolishi lozim bo'lgan qism qanday akslantirilishi lozimligini ko'rsatadi. Qiymati aniqlangan tur:
- **pad** – odatiy qiymat. Boshlang'ich va oxirgi ranglar bo'sh maydonga yoyiladi.
 - **repeat** – tuslanayotgan ranglar takrorlanadi.
 - **reflect** – tuslanayotgan ranglar ko'zgudagi aks singari takrorlanadi.
- **gradientTransform** – tuslanuvchi ranglar shaklini o'zgartirish. [6.4.3.]

Aynan ranglar emas, qandaydir shakllar hosil qilinib orqatasvir namoyon etilishi lozim bo'lsa, unda **pattern** tegidan foydalaniladi. U ham yuqorida tanishib o'tilgan standart alomatlarga ega: **width**, **height**, **patternUnits** (**gradientUnits** singari) va **x**, **y**.

Tuslanishni tasvirlashga umumiy misol:

```
<svg width="400" height="120" viewBox="0 0 400 120">
  <radialGradient id="cGradient">
    <stop offset="10%" stop-color="gold"/>
    <stop offset="95%" stop-color="green"/>
  </radialGradient>
```

```
<linearGradient id="rGradient" x1="0%" y1="0%"  
                x2="100%" y2="100%">  
    <stop offset="0%" stop-color="aqua"  
          stop-opacity="1" />  
    <stop offset="100%" stop-color="blue"  
          stop-opacity="1" />  
</linearGradient>  
<pattern id="tus" width="40" height="40" x="9"  
         y="9" patternUnits="userSpaceOnUse">  
    <circle cx="20" cy="20" r="20"  
          fill="#BBC42A" />  
</pattern>  
<circle cx="60" cy="60" r="50"  
          fill="url(#cGradient)" />  
<rect x="120" y="10" width="100" height="100"  
          fill="url(#rGradient)" />  
<rect x="230" y="10" width="100" height="100"  
          stroke="#333333" fill="url(#tus)" />  
</svg>
```

6.5. Harakatlanuvchi tasvirlar

Shakllarni harakatlantirish uchun bir necha maxsus teglardan foydalaniladi.

6.5.1. SET tegi

Eng oddiy harakatni ta'minlash uchun **set** tegidan foydalilanadi. U belgilangan vaqtdan so'ng shaklning ko'rsatilgan alomat bo'yicha qiymatini boshlang'ich holatidan o'zgartiradi. Ya'ni ma'lum vaqtdan so'ng shakl bir ko'rinishdan ikkinchisiga darhol o'tib qoladi. Ushbu teg quyidagi alomatlarga ega.

- **attributeName** — qaysi alomat bo'yicha o'zgarishi ni ko'rsatadi. Qiymati: satr.

- **attributeType** — ko'rsatilgan alomat qanday aniqlanganini bildiradi. Qiymati: aniqlangan tur, agar alomat sifatida ko'rsatilgan bo'lsa, «XML»; shamoyili faqat CSS yordamida aniqlangan bo'lsa, «CSS» qiymat berilishi kerak.

- **to** — «attributeName»da ko'rsatilgan alomatning qiymati nechaga o'zgarishini aniqlaydi. Qiymati unda ko'rsatilgan alomatniki singari bo'ladi.

- **begin** — qancha vaqtdan so'ng o'zgarishini aniqlaydi. Qiymati: vaqt.

Bir soniyadan keyin to'rt barobar kattalashib qoluvchi doiraga misol:

```
<svg width="160" height="160">
  <circle cx="80" cy="80" r="20"
    stroke="none" fill="#0000ff">
    <set attributeName="r" attributeType="XML"
      to="80" begin="1s" />
  </circle>
</svg>
```

6.5.2. ANIMATE tegi

Takroriy harakatni ta'minlash uchun **animate** tegi ishlataladi. U yuqorida [6.5.1.] izohlangan alomatlardan tashqari quyidagi qo'shimchalarga ham ega:

- **from** – qaysi qiymatdan harakat boshlanishini ko'rsatadi.
- **dur** – harakat qancha vaqt davom etishi kerakligini ko'rsatadi.
- **repeatCount** – qaytarilish soni. Qiymati butun son yoki «indefinite» (to'xtovsiz).

Yotiq chiziq bo'yicha suriluvchi doiraga misol:

```
<svg width="560" height="60">
  <circle cx="30" cy="30" r="25" stroke="none"
    fill="#0000ff">
    <animate begin="0s"
      from="10" attributeName="cx"
      to="520" attributeType="XML"
      dur="2s" repeatCount="indefinite" />
  </circle>
</svg>
```

Ko'p brauzerlar vaqt o'tgan sayin rangni o'zgartirib turishni ta'minlovchi «animateColor» tegini qo'llamaydi. Shuning uchun «animate» tegining o'zidan foydalangan holda «attributeName»ga rang qabul qiluvchi alomatning nomini berib qo'llash tavsiya etiladi.

6.5.3. AnimateMotion tegi

Shaklni ma'lum yo'nalishda harakatlantirish uchun **animateMotion** tegidan foydalilanadi. Buning uchun path tegidagi qonuniyatlar asosida maxsus chiziq chiziladi. Shakl shu chiziq bo'ylab uning boshlang'ich nuqtasidan oxirigacha harakatlanishi ta'minlanadi. Ushbu tegda ham «animate»niki singari alomatlar mavjud. Faqat qo'shimcha «path» alomatiga ega bo'lib, u orqali maxsus qoidalalar [6.4.1.] asosida egri chiziq beriladi. Misol:

```
<svg width="320" height="120">
  <path d="M10,50q60,50+100,0q60,-50+100,0"
        fill="none" stroke="green"/>
  <rect x="0" y="0" width="30" height="15"
        stroke="#ff0000" fill="none">
    <animateMotion begin="0s" dur="10s"
      path="M10,50q60,50+100,0q60,-50+100,0"
      repeatCount="indefinite" rotate="auto" />
  </rect>
</svg>
```

Bunda **path** tegi faqat qanday iz (trayektoriya) bo'yicha harakat bo'layotganini brauzerda ko'rsatish uchun yozilgan, xolos. Aslida, uni yozmasa ham bo'ladi. Istalgan darajadagi murakkab chiziq bo'ylab ko'rsatilgan vaqtda ixtiyoriy shaklni harakatlantirish mumkin, **rotate="auto"** qiymati harakatlanish paytida mos ravishda burilishni ham ta'minlaydi. Umuman olganda, ushbu alomatga butun son gradus hisobida berilsa ham bo'ladi.

Agar path tegi ifodalagan iz ko'rinishi shart bo'lsa, uni aniqlovchi satr yuqoridagidek ikki marta takror yozilishi mukammal dasturlash talablariga javob bermaydi. Aslida, «iz»ni bir marta aniqlab, kerak bo'lsa, unga boshqa yerda yana murojaat qilish mumkin bo'lishi kerak. Buning uchun **mpath** tegidan foydalaniladi. Misol:

```
<svg width="320" height="120">
    <path id="yol" fill="none" stroke="green"
          d="M10,50q60,50+100,0q60,-50+100,0" />
    <rect x="0" y="0" width="30" height="15"
          stroke="#ff0000" fill="none">
        <animateMotion begin="0s" repeatCount="10"
                        dur="10s" rotate="auto">
            <mpath href="#yol" />
        </animateMotion>
    </rect>
</svg>
```

6.5.4. AnimateTransform tegi

Shaklni **transform** alomati [6.4.3.] bo'yicha harakatlantirishda **AnimateTransform** tegi qo'llaniladi. U «to», «attributeName», «attributeType», «begin», «from», «dur», «repeatCount» singari standart alomatlarga qo'shimcha ravishda «type»ga ham ega. Bu «type» alomatida qanday o'zgarish sodir bo'lishi ko'rsatiladi. Qiymati – aniqlangan tur: matrix, translate, rotate, scale, skewX, skewY [6.4.3.].

Uning imkoniyatidan foydalanan shaklni harakat yordamida surish, hajmini o'zgartirish, og'dirish, burish va maxsus ko'rinishda akslantirish mumkin. Misol:

```
<svg width="200" height="200"
      style="border: 1px solid red;">
  <rect x="30" y="30" width="60" height="40"
        fill="none" stroke="green">
    <animateTransform attributeName="transform"
        type="rotate" repeatCount="indefinite"
        from="0 100 100" to="360 100 100"
        begin="0s" dur="3s" />
  </rect>
</svg>
```

Keyingi «JavaScript»ga bag'ishlangan kitobimizda nafaqat shakllarni, balki barcha obyektlarni harakatlantirish va aylantirish, rang va ko'rinishlarini o'zgartirish, yashirish va paydo qilish, o'chirish va yaratish haqida bat afsil ma'lumot beramiz.

Umuman olganda, SVG juda boy imkoniyatlarga ega alohida til sifatida shakllangan. Hozircha uning 1.1-tal-qinidagi ko'p imkoniyatlar brauzerlarda qo'llaniladi. Biz ularning asosini tashkil qiluvchi muhim teglar bilan tanishdik, ular orqali kerakli shaklni hosil qilish va harakatlantirish mumkin. Ushbu kitobda yoritilmagan SVGning qolgan teglari HTML va CSSning alomatlariga o'xshab ketadi. Shuning uchun har biriga alohida to'xtalishni lozim topmadik.

7-BO'LIM

FORMA ELEMENTLARI

7.1. Forma yaratish

Ma'lumot va qiymatlarni saqlash, kiritish, qayta ishlash, tahrirlash, muhimi, serverga yuborish uchun formadan foydalaniladi. Formaning ichida yoziladigan teglar asosida akslantiriladigan obyektlarni **elementlari**, serverga qiymat yuborish uchun xizmat qiladigan obyektlarni esa **darchalari** deb yuritamiz. Forma form tegi yordamida yaratilib, o'z ichiga ma'lum qiymatlarni qabul qiladigan darchalarni mujassamlashtirgan quti vazifasini bajaradi. Serverga ma'lumotlar shu quti xususiyatiga qarab jamlanib yuboriladi. Ushbu teg quyidagi alomatlarga ega:

- **accept-charset** — yuborilayotgan yozuvlar kodlamasining nomi, qiymati: satr.
- **action** — qiymatlarni yuboriladigan server manzili, qiymati: ishorat. Ko'rsatilmassa, joriy veb-sahifa manzili olinadi.
- **autocomplete** — foydalanuvchi darchalarga oldin qiymat kiritgan bo'lsa, veb-sahifa qayta yuklangandan keyin dastavval obyektlarda o'sha qiymatlardan birini qo'yishga imkon yaratadi. Mantiqiy tur qabul qiladi.
- **enctype** — forma ma'lumotlarni yuborishdan avval qanday kodlashni aniqlaydi, qiymati aniqlangan tur:

- **application/x-www-form-urlencoded** – odatiy qiymat, satrlardagi belgilarni maxsus keli-shuv asosidagi «urlencoded» funksiyasi yorda-mida kodlaydi. Unda bo'sh joy «+», lotin harflaridan boshqa belgilar esa «%» bilan o'n oltilik sanoq tizimidagi maxsus raqam orqali yoziladi. Masalan, kirill harflari bilan yozilgan «**китоб**» so'zi quyidagi tarzda akslantiriladi: «%D0%BA%D0%B8%D1%82%D0%BE%D0%B1».
 - **text/plain** – faqat bo'sh joy «+»ga almashtiri-ladi, ma'lumot kodlanmaydi.
 - **multipart/form-data** – ma'lumot kodlanmay-di, fayl yuborishni ta'minlaydi.
- **method** – qiymatlarni serverga yuborish usuli.

Aniqlangan tur:

- **get** – qiymatlarni darchalarning nomi ortidan «=> belgisidan keyin yozib, «&» belgisi bilan birlashtirib, «**action**»da ko'rsatilgan ishoratdan so'ng «?» belgisidan keyin qo'shib, yangi ishorat hosil qiladi va shu yo'sinda ma'lumot yuboradi. Masalan, «**a**» nomli darchaga «**1**», «**b**» nomli darchaga «**www**» qiymat berilgan bo'lsa, yuboriladigan manzil ishoratiga ushbu «?**a=1&b=www**» satri qo'shiladi. Bunda bar-cha yuborilayotgan ma'lumotlar foydalanuvchi tomonidan o'qish va o'zgartirishga qulay. Albat-ta, juda uzun va katta hajmdagi qiymatlarni bu yo'l bilan yuborish maqsadga muvofiq emas, unda ko'pi bilan 4Kb ma'lumot jo'natiladi.
- **post** – formadagi darchalarning qiymatlarni «**enctype**»da ko'rsatilgan kodlash asosida foy-

dalanuvchiga ko'rinxmaydigan tarzda yuborishini ta'minlaydi. Albatta, bunda serverda o'rnatilgan maxsus chegaragacha bo'lgan hajmdagi qiymatlarni, hatto fayllarni yuborish imkoniyati bor.

- **put** – ma'lumotlarni fayl tarkibida matn sifatida yuborish. Umuman olganda, forma yordamida serverga hamma ma'lumotlar bitta fayl sifatida jo'natiladi; «enctype=multipart/form-data» tarzida qiymat berish orqali ma'lumotlarni alohida fayl sifatida yoki «get» uslubida keltirilgani kabi server manziliga qo'shimcha tarzda yozish bilan hamda «post» uslubida yoritilgan kabi alohida biriktirish yordamida yuborishlardan tashqari; yo'naltirilayotgan faylning tavsif xabari [2.3.] ichiga qo'shish yoki tarkibini tashkil etuvchi matni orqali berish ham mumkin.

▪ **name** – formaning nomi. Agar veb-sahifada bir necha forma ishtirok etsa, ularning nomlari takrorlanmaydigan bo'lishi talab etiladi. Skriptlar yordamida formaga uning nomi bo'yicha murojaat qilishni ta'minlaydi. Qiymati: identifikator.

▪ **noValidate** – odatda, darchalarga kiritilgan qiymatlar yuborishdan oldin talab etilgan andozaga to'g'ri kelishi tekshiriladi, agar ushbu alomat ko'rsatilsa, tekshiruv o'chiriladi.

▪ **target** – serverdan qaytgan javob qayerda akslanshini ko'rsatadi. [2.4.]

Yuqorida ma'lumot yuborish «method»ining uch xil usuli bilan tanishdik. Aslida, ular o'ndan ortiq va yana qo'shib borilishi mumkin. Hammasining yuborish tex-

nologiyasi shu uchala uslubdan aynan biridir, faqat nomlanishi boshqacha. Sababi, server uslub nomiga qarab unga mos amal bajaradi. Aslida, hech qanday yangicha texnologiya yaratmasdan, server bilan ma'lumot almashishni ma'lum qolipga solish maqsadida HTTP qaydnomasi asoschilaridan biri Roy Tomas Filding 2000-yilga kelib **REST** (Representational State Transfer – taqdimot holatini yuborish) atamasini kiritdi. Unga ko'ra, yuborish uslublarining nomi kengaydi: post, get, put, delete, patch, view, copy, head, link, unlink, purge, propfind, lock, unlock, options. REST uslubi bilan to'laqonli ishlash keyingi kitobda batafsil yoritiladi.

7.2. INPUT darchasi

Forma elementlari ichida eng murakkabi **input** darchasi hisoblanadi, u ko'rsatilgan turiga qarab bir necha xil qiymatlarni qabul qiladi, obyektning ko'rinishi ham shunga mos ravishda o'zgaradi. U yopilish qism mavjud bo'lmagan input tegi orqali ifodalanadi, ya'ni barcha xususiyatlari faqat alomatlar yordamida beriladi.

Dastavval uning barcha, ya'ni HTML5da yangi qoshilgan hamda eskirgan deb hisoblanmagan alomatlari bilan tanishib o'tamiz:

- **accept** – faqat fayl yuborilayotgan holda ushbu alomatda MIME turi [2.4.] yoki uning andozasi ko'rsa-

tiladi. Masalan: tovush fayllar uchun «**audio/***», rasm fayllar uchun «**image/***» sifatida ko'rsatish mumkin. Bunda tanlangan fayl ko'rsatilgan andozaga mos kelmasa, u tanlovlardan oynasida namoyon bo'lmaydi. Qiymati satr, katta-kichik harflar ahamiyatga ega emas. Bir necha MIME tur ko'rsatilishi lozim bo'lsa, vergul bilan ajratiladi.

- **alt** — agar input turi rasm sifatida ko'rsatilgan bo'lsa, rasm akslantirilmaydigan hollarda ushbu alomatda yozilgan matn ko'rindi. Qiymati: satr.

- **autocomplete** — o'zi to'ldirish imkoniyatini yoqish yoki o'chirish. [7.1.]

- **autofocus** — darchani faollashtiradi («Enter» yoki («Return» singari kiritish tugmasi yoxud bo'shliq bosilganda ishlashi uchun shay holda turadi). Ushbu alomat bitta veb-sahifadagi elementlarning faqat bittasida berilishi tavsiya etiladi. Chunki doim kursov (ko'rsatkich) turgan, ya'ni ayni paytdagi faol darcha yagona bo'ladi. Aks holda, ushbu alomatli darchalarning eng oxirgisi faol bo'ladi. Zarurmas qiymat qabul qiladi.

- **checked** — input turi tanlanuvchi yoki belgilanuvchi sifatida ko'rsatilgan bo'lsa, yaratilayotganda rost qiymatni qabul qilishini anglatadi (zarurmas qiymat).

- **dirname** — qiymati: satr. Serverga unda ko'rsatilgan satr nomi bilan «ltr» yoki «rtl» qiymatlaridan biri [2.7.] yuboriladi. Ushbu alomatni Internet Explorer va Firefoxlar qo'llamaydi. Ayrim darchalarda arab yozuvni ishlatalishi mumkin bo'lgan hollardagina bu imkoniyat asqotishi ehtimoli bor va u ham juda kam.

- **disabled** — darchanining faolligini o'chiradi. Ma'lum

mot kiritish yoki o'zgartirish imkoniyati bo'lmaydi (zarrumas qiymat).

▪ **form** — ushbu alomatda formaning «id»sidagi qiymat beriladi. Agar darcha «form» tegining ichida joylashgan bo'lsa, bu shart emas. Ba'zi hollarda element formadan tashqarida (boshqa faylda) yaratilgan bo'lishi mumkin, shunda tashqaridagi darchani formaga biriktrish imkoniyatini yaratadi. Lekin ba'zi brauzerlar buni qo'llamaydi va bunday ishlatish tavsiya etilmaydi.

▪ **formaction** — ushbu alomat formadagi «action» bilan bir xil vazifani bajaradi. Bitta formada bir necha tugma qo'yilib, har birini bosganda o'ziga xos boshqa-boshqa manzilga murojaat qilish lozim bo'lsa, shu alomatdan foydalaniladi va u ustuvor hisoblanadi. Ya'ni «input»ning turi jo'natishga mo'ljallangan tugma bo'lsa, «formAction», «formEnctype», «formTarget», «formMethod» va «formNoValidate»lar mos ravishda formaning alomatlariga muqobil bo'ladi.

▪ **list** — «datalist» tegining [7.4.4.] «id»si. Ma'lumot kiritilayotganda ro'yxatdan mos satrni tanlash imkoniyatini beradi.

▪ **max** — son yoki sana sifatidagi kiritilishi mumkin bo'lgan eng katta qiymatni aniqlaydi.

▪ **maxlength** — kiritilayotgan satrdagi belgilarning eng ko'p miqdori nechta bo'lishini chegaralaydi, qiymati: son.

▪ **min** — son yoki sana sifatidagi kiritilishi mumkin bo'lgan eng kichik qiymatni aniqlaydi.

▪ **minlength** — kiritilayotgan satrdagi belgilar eng kamida nechta bo'lishi (son).

- **multiple** — qiymatlar bir necha bo‘lishi mumkinligini ko‘rsatadi. Bunda ular vergul bilan ajratilishi kerak. Sintaksis tekshirilayotganda bu muhim.

- **name** — darchaning nomi. Bitta formadagi darchalar nomi takrorlanmasligi shart. Ushbu nom bilan uning qiymati serverga yuboriladi. Skriptlar ham shu nom bilan murojaat qilishi mumkin. Qiymati: identifikator.

- **pattern** — muntazam ifodalar yordamida kiritiladigan qiymatga andoza berish. Qiymati: muntazam ifoda qoidalari bo‘yicha (bu haqida dasturlashga bag‘ishlangan kitobimizda batafsil to‘xtalamiz) yozilgan satr. Misol tariqasida faqat beshta bosh lotin harflari yozildigan ko‘rib chiqsak, u quyidagicha bo‘ladi:

```
<input pattern="^[A-Z]{5}$">
```

- **placeholder** — qiymati: satr. Ushbu alomatdagi matn xiraroq rang bilan bo‘sish darchaga yozilib turadi. Biror qiymat berilmaganda nimani kiritish lozimligi haqida xabar berib turishi uchun qo’llaniladigan juda qulay vosita. Misol:

HTML <input placeholder="Ismingizni kriting">

Natija

Ismingizni kriting

- **readonly** — qiymatni o‘zgartirish imkonidan mahrum etadi (zarurmas qiymat).

- **required** — darchaga ma’lumot kiritishni majburiy qilib qo‘yadi (zarurmas qiymat). Aks holda forma serverga yuborish uchun murojaat qilmaydi.

- **size** — darchaning kengligi. Qiymati: son, nechta belgi sig'ishini ko'rsatadi.
- **src** — input turi tasvir sifatida aniqlangan bo'lsa, ushbu alomatda rasmning manzili ko'rsatiladi.
- **step** — input turi son sifatida ko'rsatilgan bo'lsa, undagi qiymatni qo'shimcha paydo bo'ladigan millarni bosish orqali oshirish yoki kamaytirish qadami.
- **value** — darcha yaratilayotganda qanday qiymat bilan akslanishini ko'rsatadi.
- **type** — darcha qanday turdaligini aniqlaydi. (Quyida batafsil tanishamiz.)

7.3. INPUT darchasi turlari

INPUTning alomatlari bilan tanishib chiqilganda ma'lum bo'ldiki, ularning ko'pchiligi faqatgina ko'rsatilgan turdag'i darchalargagini aloqador. Masalan, matn yoziladigan darcha uchun «step» hech qanday vazifani bajarmaydi, vaqtini aniqlaydigan darchaga «checked» qo'yish ma'nosiz. Bunday ortiqcha alomatlar yozilgan holda, HTMLning qoidasiga muvofiq, shunchaki e'tiborsiz qoldiriladi.

Dastavval HTML4gacha mavjud bo'lgan turlar bilan tanishib o'tamiz, sababi, HTML5da qo'shilgan ko'pgina foydali darchalar ba'zi brauzerlarda hanuzgacha qo'llanmaydi. Xullas, **input**ning **type** alomati quyidagi qiymatlarni qabul qiladi:

- **button** – tugma hosil qiladi.
- **checkbox** – belgilanuvchi (bir nechta belgilash mumkin) darcha hosil qiladi.
- **file** – fayl tanlab yuborishni ta'minlaydi.
- **hidden** – yashirin darcha, foydalanuvchiga ko'rinxaydi.
- **image** – rasmdan iborat darcha, bosilganda forma serverga jo'natiladi.
- **password** – har bir belgisi yulduzchadan iborat yashirin satr. Kalit so'z yozish uchun.
- **radio** – tanlanuvchi darcha hosil qiladi. Odatta, bir necha yozuvlardan faqat bittasi tanlanadi, shuning uchun ushbu turdag'i darchalarning bir nechta ketma-ket keladi va bir xil nom (name) bilan ifodalanadi.
- **reset** – formadagi barcha qiymatlarni boshlang'ich holatiga keltiruvchi tugma.
- **submit** – forma(ma'lumotlari)ni serverga yuboruvchi tugma.
- **text** – bir qator matnga mo'ljallangan darcha.

Endi HTML5da qo'shilgan yangi turlar bilan tanishamiz, shu bilan birga, ommabop brauzerlarning qaysi talqinlaridan boshlab qo'llanishini ham ko'rib o'tamiz. Qaysi brauzerlarda haligacha tegishli turdag'i darchani akslantiruvchi imkoniyat mavjud bo'lmasa, unga «—» belgisi qo'yilgan va yaqin kelajakda bu kamchiliklar bartaraf etilishiga umid qilib qolamiz.

Qiymat								
color	—	14	20	11.5	—	29	4.4	—
email	10	11	5	10.1	5	4	3	3.1
date	—	13	20	10.1	—	—	4.4	5.0
datetime	—	11	20	10.1	—	—	4.4	5.0
datetime-local	—	11	20	10.1	—	—	4.4	5.0
month	—	13	20	10.1	—	—	4.4	5.0
number	10	11	6	10.1	5	29	4	4.0
range	10	11	4	10.1	3.1	23	4.3	5.0
search	10	11	5	10.1	5	4	2.3	4.0
tel	10	11	5	10.1	5	4	2.3	3.1
time	—	13	20	10.1	—	—	4.4	5.0
url	10	11	8	15	6	4	4	3.1
week	—	13	20	10.1	—	—	4.4	5.0

Darcha **url**, **tel** yoki **email** turida bo'lsa, kiritilgan satr o'zimiz bilgan mos qolipga tushishini brauzer formani yubormasdan oldin tekshiradi. Agar **search** turida bo'lsa, uning **text** turidan farqi – kiritilayotgan belgilar yozilganda «x» paydo bo'lib, uni bosganda matnni o'chirib yuborish imkonini ham yaratadi. Qolgan barcha turdagilar uchun brauzer ko'rsatilgan nomga mos va o'ziga xos andozada yozishini ta'minlovchi, tanlash imkoniyati bilan kiritiladigan qiymatlar hosil qiladi.

Muammo shundaki, vaqtin ko'rsatadigan turdagilar uchun sana shaklining qay tarzda akslanishi lozimligini ko'rsatish mumkin emas. Masalan, amerikaliklar sanani ifodalashda avval oy raqamini, keyin kun raqamini yozishadi, biz kunni birinchi yozishga odatlanganmiz. Shuning uchun ham eng ommabop foydalaniladigan

«Safari», «Firefox» haligacha ular uchun interfeys qil-magandir. Umuman olganda, qo'shimcha kutubxonalar yordamida (masalan: *JQueryUI*) ushbu muammoga yechim topish mumkin.

7.4. Forma darchalari

Endi formaning inputdan boshqa teglar yordamida yaratiladigan darchalari batafsил bilan tanishamiz. Ularning hammasi uchun yuqorida keltirib o'tilgan alomat-lardan «formaction», «formnovalidate», «formenctype», «formtarget», «formmethod», «form», «name», «value», «readonly», «required», «autofocus», «autocomplete» hamda «disabled»lar tavsiflanganday vazifani bajaradi. Quyida ushbu sanab o'tilganlar va universal alomat-lardan tashqari faqat o'ziga xoslari keltirib o'tiladi.

7.4.1. Tugma

Tugma yuqorida <input type="button | reset | submit"> kabi ko'rinishda yaratilish mumkinligini ko'rib o'tdik. Lekin undan tashqari alohida **button** tegi ham mavjud bo'lib, o'zining ichida boshqa teglar yozishga ruxsat etadi. Bu tugmaning ichida rasm qo'yish, yozuvining shakli va rangini o'zgartirish, qolaversa, boshqa kerakli

obyektlarni akslantirishga imkon beradi. U quyidagi o'ziga xos alomatga ega:

▪ **type** — tugmaning vazifasini anglatadi. Qiymati aniqlangan tur:

- **button** — oddiy tugma.
- **reset** — formadagi qiymatlarni boshlang'ich holatiga keltiradi.
- **submit** — forma ma'lumotlarini serverga yuboradi (odatiy qiymat).
- **menu** — **menu** tegi [3.4.2.] bilan yaratilgan ro'yxat «id»si, tugma bosilganda tanlash uchun biriktirilgan menyu namoyon bo'ladi.

Misol:

```
<button>
    
    Suratli tugma
</button>
```

7.4.2. Matn yozish darchasi

Bir necha qatordan iborat matnlar yozish uchun **textarea** tegidan foydalaniladi. U quyidagi o'ziga xos alomatlarga ega:

▪ **cols** — eni (kengligi), qiymati: son, belgi sig'ish hisobida.

▪ **rows** — bo'yi (balandligi), qiymati: son, belgi sig'ish hisobida.

▪ **maxlength** — matndagi belgilarning ko'pi bilan nechta bo'lishi, qiymati: son.

▪ **wrap** — keyingi satrga qay tarzda tushishni sozlaydi. Qiymati: aniqlangan tur:

- **soft** — odatiy qiymat. Bunda satrga sig'magan matnning davomi keyingi qatorga o'z-o'zidan tushadi va serverga yuborilayotganda ushbu qatorlar bo'linishi maxsus belgi sifatida jo'natilmaydi, faqat kiritish tugmasi (enter) yordamida qatorlar bo'lingan bo'lsa, ular yuboriladi.
- **hard** — bu qiymatni ko'rsatishdan avval «rows» alomatiga son berilishi talab etiladi. Bunda har bir qator bo'linishi maxsus belgi (\n) bilan serverga yuboriladi. Kiritish tugmasi (enter) atayin bosilganmi yoki avtomatik qo'shildimi, farqlash imkonsiz.
- **off** — keyingi qatorga o'tkazishni o'chiradi. Barcha matn bitta satrdan iborat bo'lib, maydonga sig'magan holda surgich (scroll) paydo bo'ladi.

7.4.3. Tayinlanuvchi (select) darcha

Yuqorida input tegi yordamida tanlanuvchi (radio) va belgilanuvchi (checkbox) darchalar qanday qo'shilishi bilan tanishib o'tdik. Ular keltirilgan ro'yxatdan bir yoki bir nechasini ajratib olish uchun foydalilanadi va unda ro'yxatdagi hamma elementlari ko'rinish turadi. Tayinlanuvchi darchada, odatda, faqat ro'yxatning belgilangan elementigina ko'rsatiladi. (Albatta, sozlash orqali bir necha qatorni ko'rsatadigan qilish mumkin, bunda uning maydoniga sig'magan elementlar uchun surgich

paydo bo'ladi.) Barcha ro'yxat yordamchi «mil» tugmasini bosish orqali ko'rildi. Bunday ochiluvchi ro'yxatni tayinlanuvchi darcha deb nomlaymiz va u **select** tegi yordamida yaratiladi. O'ziga xos alomatlari:

- **multiple** — bir nechtasini tanlashga imkon beradi, zarurmas qiymat.

- **size** — nechta element (qator) ko'rinishi kerakligini aniqlaydi, qiymati: son.

Tayinlanuvchi darchadagi ro'yxatning elementlari maxsus **option** tegi yordamida beriladi. U quyidagi alomatlarga ega:

- **disabled** — elementni nofaol qiladi.
- **selected** — yaratilgan payti qaysi element tanlanib turishini ko'rsatadi.
- **value** — elementning ko'rinnmaydigan asl qiymati. U berilmagan holda teg ichidagi yozuvni qabul qiladi. Ro'yxatdagi yozuvlar uzun bo'lishi mumkin, shuning uchun ularni qiymat sifatida qarash ma'qul emas. Dasturlovchilarga qaysi elementlar tanlanganligini bilish uchun ularning qiymati bo'yicha murojaat qilish qulay.

Elementlar ko'payib ketganda ularning ichidan saralab olish mushkullashadi. Alifbo tartibida ro'yxatni saralash ham doimo samara beravermaydi, sababi, nomlanishlar turli sohalarga oid bo'lishi mumkin. Bunday vaziyatlarda elementlarni ma'lum mavzularga ajratib guruhlash lozim bo'ladi. Guruhlash uchun **optgroup** nomli maxsus teg ishlataladi. Uning nofaollashtiruvchi «disabled» va guruh nomini ko'rsatuvchi «label» alomatlari mavjud. Misol:

```
<select size="2" multiple>

    <optgroup label="Group 1">
        <option> Option 1.1 </option>
    </optgroup>

    <optgroup label="Group 2">
        <option selected> Option 2.1 </option>
        <option> Option 2.2 </option>
    </optgroup>

    <optgroup label="Group 3" disabled>
        <option> Option 3.1 </option>
        <option> Option 3.2 </option>
        <option> Option 3.3 </option>
    </optgroup>

</select>
```

7.4.4. Tanlanuvchi satr

Satrni input darchasiga to'liq yozish ko'p hollarda malol keladi. Agar ko'rsatilgan darchaga kiritilishi mumkin bo'lgan satrlarning aksariyat matnlari ma'lum bo'lsa, ulardan mosini tanlab, faqat ististno tariqasidagi yozuvningina kiritish anchagini qulaylik yaratadi. Ushbu maqsadda **datalist** tegi qo'llaniladi, uning ichida ro'yxat «option» tegi yordamida beriladi. Mos input darchasiga «datalist»ning «id»si beriladi. Misol:

Brauzer nomi:

```
<input list="browsers" name="myBrowser">
<datalist id="browsers">
    <option value="Chrome">
    <option value="Firefox">
    <option value="Internet Explorer">
    <option value="Opera">
    <option value="Safari">
    <option value="Microsoft Edge">
</datalist>
```

7.4.5. Ma'lumotlarni shifrlash

Ko'p hollarda, ayniqsa, kalit so'zlarni va maxfiy ma'lumotlarni yuborishda ularni yashirin tarzda jo'natish dolzarb hisoblanadi. Hozirda barcha brauzerlar maxsus shirflash algoritmlarini qo'llaydi va **keygen** tegi orqali ifodalanadi. Shifrlash ochiq va yopiq kalitlar asosida amalga oshiriladi, ochiq kalit serverga yuboriladi, yopig'i brauzerning o'zida saqlanib, kelayotgan shirflangan ma'lumotlarni ochishda ishlatiladi. Ushbu kalitlarning qiymatiga qarab ayni bir xil ma'lumot turlicha ko'rnishda kodlangan holda kelishi mumkin. Odatda, shifrlashning RSA algoritmi qo'llaniladi, brauzer va server tushungan holda boshqa nomdagisini ishlatish mumkin, nomlanish **keytype** alomatida ko'rsatiladi. Forma yuborilayotganda qiymatlar qay tartibda o'zgarishi esa **challenge** alomatida ko'rsatiladi. Misol:

```
<form>
    <keygen name="name" challenge="__"
            keytype="rsa" disabled>
</form>
```

Ta'kidlab o'tish joizki, ushbu uslubdan foydalanish yaxshi samara bermaydi. Afsuski, bu ma'lumot almashtish jarayonini sekinlashtiradi va xavfsizlik jihatidan ham kutilgan natijani bermaydi. Albatta, yuborilgan ma'lumotni maxsus terishlar orqali ma'lum muddatda aniqlash imkoniyati mavjud. Hozirda barcha ma'lumotlarni, hatto ishoratni ham kodlab yuboradigan «<https://>» qaydnomasi keng qo'llaniladi. Ushbu texnologiya ishlatalganda ruxsat etilgan sertifikat ko'rsatilgandagina ma'lumotlarni ochishga ruxsat beriladi. Hozirda bu ma'lumotlar maxfiyligini ta'minlashning ommabop, qulay va ishonchli vositasidir.

7.4.6. Jarayonlarni ko'rsatish

Veb-sahifa ustida qandaydir amallar bajarilar ekan, ba'zan berilgan buyruq qanday natija bilan yakunlanganini ko'rish, agar u uzoq vaqt olsa, yana qancha kutish lozimligini ko'rsatib borish talab etiladi. Agar kutilgan natija bir soniyada akslantirilmasligi ehtimoli mavjud bo'lsa, bu haqida foydalanuvchini ogohlantirib, maxsus kuzatib borish darchasini ko'rsatish tavsiya etiladi. Buning uchun eng oddiy **progress** tegidan ham foydalanish mumkin. Uning son qabul qiluvchi ikkita alomati bor: «max» va «value»; «max»ta qadamdan

iborat amallarning har biri bajarilganda «value»ga shu qiymatni berib borish yo'li bilan yoki har ma'lum millisoniyada bajarilgan ish hajmiga qarab foiz hisobida son berish orqali ushbu obyektni akslantirib borish mumkin. Umuman olganda, dasturlarni o'rnatish yoki buyruq bajarilishini kutish jarayonida har bir o'quvchi foizni ko'rsatuvchi tasmani ko'rgan deb ishonamiz. Misol:

<progress value="79" max="100">



Ko'rinishidan shunga o'xshab ketadigan meter tegi ham mavjud, u doimiy kuzatilib borilishi lozim bo'lgan jarayonlar uchun qo'llaniladi. O'ziga xos jihat shundaki, unda yuqori va quyi chegaralarni ko'rsatib qiymat shu oraliqdan chiqadigan hollarda ogohlantirish tarzida boshqa rangga kiradi. Quyidagi alomatlarga ega:

- **value** — joriy qiymati;
- **min** — eng kichik qiymat, odatiy qiymati: 0;
- **max** — eng katta qiymat, odatiy qiymati: 1;
- **low** — quyi xavfli qiymat;
- **high** — yuqori xavfli qiymat;
- **optimum** — aslida, me'yoriy qiymati qanday bo'lishi lozimligi;

Mantiqan olib qaraganda, haqiqiy son qabul qiladigan ushbu qiymatlar «min» va «max»lar orasida bo'lishi talab etiladi. Masalan, yurgizgichdagi suyuqlik harorating o'zgarish jarayonini kuzatib borishda; agar 80° dan yuqori bo'lsa, bu vahimali holat, agar 30° dan past bo'lsa bu o'ta sust holat sifatida baholash lozimligi sozlansa, quyidagicha teg yozish mumkin:

```
<meter value="22" max="100" low="30" high="80">
```

Bunda xavfli oraliqdagi qiymatlar boshqa rangda akslantiriladi.

Aslida, input so'zining egizagi hisoblangan output nomli tegi faqatgina HTML5da kiritilgan. Bu kirituvchi darcha emas, aksincha, chiquvchi natijani ko'rsatish uchun mo'ljallangan obyekt. Ungacha o'z vaqtida hodisaning natijasini chiqarib borish uchun turli teglardan foydalananish biroz noqulaylik tug'dirar edi. Ushbu teg qay darajada samarali ekanligini ko'rish uchun forma elementlaridagi qiymatlar o'zgarganda «output»ga chiquvchi natijani kuzatib borish imkonini beruvchi misol keltiramiz:

```
<form oninput="result.value = a.value * b.value">
    <input type="range" name="b" value="50"> *
    <input type="number" name="a" value="10"> =
    <output name="result"> 500 </output>
</form>
```

Ta'kidlab o'tish lozimki, formaning «oninput» hodisasiga JavaScript kodi yozilgan satr berilgan. Unda formadagi obyektlarning qiymati, ularning nomlari bilan qanday olinishi va ular ustida matematik amal bajariishi ko'rsatilgan, xolos.

7.5. Formaga obyektlarni joylash

HTML tilidagi formani ham o'zimiz odatlangan operatsion tizimdagи oynaga qiyoslashimiz to'g'ri bo'ladi. Oyna ichidagi tugmalar, yozuv va kiritish darchalari foydalanuvchiga qulay tartibda joylashtirilgan bo'ladi. Biz ham o'z veb-sahifamizda bunga katta e'tibor qarashimiz lozim. Maxsus jilolar berish orqali veb-formani operatsion tizimdagи oyna ko'rinishiga keltirish mushkul emas.

Yuqorida tanishgan darchalarimiz maxsus teg yordamida brauzerda obyekt yaratilishini ta'minlaydi, xolos. Lekin foydalanuvchi ma'lumot kiritishi yoki tanlashi uchun darchaga qanday qiymat berilishi lozimligini bilshe kerak. Buning uchun har bir darchaga izoh sifatida nom berishni ta'minlovchi **label** tegi qo'llaniladi. Uning «for» nomli alomati mavjud bo'lib, qaysi darcha uchun ekanligini ko'rsatadi. Barcha formadagi darchalar uchun label yozish tavsiya etiladi. Undagi yozuvga bosilganda tegishli darchaga bosilganday samara beradi. Bu belgilovchi input obyektlar uchun juda qulay, chunki obyektning hajmi kichik, yozuv esa ancha uzun. Misol:

```
<label> Belgilash: <input type="checkbox"></label>
```

Formadagi obyektlarni guruhlash uchun **fieldset** tegidan foydalaniladi. Uning ichidagi elementlar to'g'ri to'rtburchak ichida akslantiriladi. Chegaraviy chiziqning tepasiga guruh nomini qo'yish uchun **legend** tegi ishlataladi. Misol:

```
<form>
    <label for="yoshi">
        Yoshi:
        <input name="yoshi" type="number" value="34">
    </label> <br>
    <fieldset>
        <legend> Jinsi </legend>
        <label>
            <input name="jinsi" type="radio"> Erkak
        </label> <br>
        <label>
            <input name="jinsi" type="radio"> Ayol
        </label> <br>
    </fieldset>
    <p><input type="submit"></p>
</form>
```

CSS bilan tanishish jarayonida obyektlarni bir tekisda bejirim qo'yish uchun nimalar qilish kerakligini batafsil ko'rib chiqamiz.

Umuman olganda, qiymatlarni saqlash uchun doim forma yaratish shart emas. Odatda, object va applet teglarida ma'lum qiymatlarni maxsus nomlar bilan **param** tegi yordamida saqlash mumkin. Lekin ushbu o'zgaruvchilarni serverga yuborish yoki foydalanuvchi tomonidan o'zgartirish maxsus skriptlar yordamida boshqacha tarzda amalga oshiriladi. Mukammalroq tasavvur hosil qilish uchun **param** tegini qo'llashga misol keltiramiz:

```
<object classid="my.class">
  <param name="manzil" value="http://yurt.uz/">
  <param name="rang"    value="magenta">
  <param name="yoshi"   value="35">
</object>
```

8-BO'LIM

CSS TARKIBI

8.1. CSSni bog'lash

CSS (Cascading Style Sheets — sahifaning pog'onali shamoyili) obyektlarga bezak va jilo berishga mo'ljallangan. Zamonaviy dasturlashda HTML tili faqat sahifada nima chiqishini va qanday tartibda joylashishini ko'rsatadi, xolos. CSS esa akslantirilayotgan obyektlarning qay tarzda ko'rinishini va qanaqa namoyon bo'lishi lozimligini belgilaydi. Bunday imkoniyat HTMLda yozilgan kodning hajmini kichraytirishga imkon yaratadi, har bir obyektning qanday tusda akslanishini uning tegida yozish shart emas.

Turli xil qurilmalar uchun sahifaning o'ziga mos ko'rinishda namoyon bo'lishini ta'minlaydi. Kompyuter, smartfon, planshet, televizor kabilarning o'lchamiga mos ravishda sahifadagi obyektlarni joylashtirish mumkin. Bunda nafaqat masshtabni o'zgartirish, balki har bir qurilmada ularning foydalanuvchilari ko'nikkan ko'rinishda obyektlarni ko'rsatsa bo'ladi.

CSSdan foydalanish tezkorlikni oshirishda ham katta rol o'ynaydi. Birinchidan, HTML sahifaning hajmi kichikligi uchun u tez yuklansa, ikkinchidan, bir qancha veb-sahifalar uchun yagona CSS faylini qo'llab, kod

yozilishini qisqartirish mumkin. Uchinchidan, CSS alohida fayl sifatida yuklansa, brauzerlar uni o'zida keshlab (xotirasiga saqlab olib), har safar uni yuklayotganda ortiqcha vaqt ham, internet trafigi ham sarflab yotmaydi. Veb-sahifaning hajmi kamligi nafaqat vaqt va trafikdan tejashga yordam beradi, balki yozilgan qonuniyatlar kamligi uchun ularni brauzer eslab qolib, akslantirishida ham tezkorlikka erishishi ta'minlanadi.

CSS uch xil ko'rinishda HTMLga bog'lanadi:

1. Sahifaning bosh (head) qismida **style** tegi ichida e'lon qilish orqali.
2. Alohida «css» kengaytmali faylda e'lon qilib, sahifaning bosh qismida unga murojaat etish orqali:

```
<link rel="stylesheet" type="text/css"
      href="mystyle.css">
```

3. Har bir tegda mavjud bo'lgan universal alomat **stylega** CSS qonuniyatlari bo'yicha yozilgan satr sifatida qiymat berish orqali.

Birinchi va uchinchi usullardan foydalanish tavsiya etilmaydi. Sababi, ularda HTML va CSSda yozilgan kodlar aralashib ketadi. Ikkinci usulda esa biz birgina faylni almashtirish orqali saytimizning to'laligicha shamoyilini o'zgartirishimiz mumkin. Shuni ta'kidlab o'tish lozimki, ayni bir obyektning bitta alomatiga ikki xil qiymat berilsa, brauzer oxirgi qiymat bo'yicha akslantiradi. Shuning uchun uchinchi usulda yozilgan buyruq avvalgilarini e'tiborsiz qoltirishni taqozo etadi va qaysidir hollarda bu imkoniyat ham asqotishi mumkin.

Bir-biriga ziddiyatli shartlar berilganda, odatda, «style» alomati ichidagisi inobatga olinsa ham, alohida fayl sifatida yaratilgan qonuniyatlarni qat'iy tatbiq etish usuli mavjud. Buning uchun «!important» (muhim) jumlasini berilgan qiymatdan so'ng yozish lozim. Masa-lan, qayerda ko'rsatilgan rang qat'iy tatbiq etilishi lozim bo'lsa, u quyidagicha yoziladi:

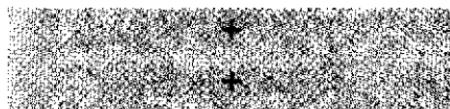
```
color: red !important;
```

Agar «!important» CSS faylda ham, teg «style»ida ham yozilgan bo'lsa, albatta, eng yaqini, ya'ni «style» ichidagisi inobatga olinadi.

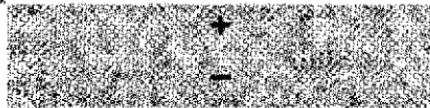
Shu o'rinda yana bir muhim narsani qayd etib o'tish lozim. Har bir brauzer uchun foydalanuvchining o'zi maxsus qoidalar asosida yaratib olgan CSS faylini yuklab, uni barcha veb-sahifalarga birday tatbiq etish mumkin. Bunda sahifa ort rangini, husnixatlar nomi, rangi va o'lchamini, ba'zi teglarning akslantirilish jilosini va boshqa CSS standartlari bo'yicha qonuniyatlarni belgilash mumkin. Kelgusida brauzerga qo'shimcha yukanib, barcha sahifalar uchun tatbiq etiladigan tashqi CSS faylni va brauzerning o'zidagi sozlashlar orqali akslantirish uchun kiritilgan maxsus o'zgartirishlarni «foydalanuvchi jilosi» deb yuritamiz. Veb-sahifa yaratuvchi HTML va CSS fayllar ichida yozish orqali berilgan qonuniyatlarni esa «muallif jilosi» deb ataymiz. Har ikkisida ham «!important»ni o'z qonuniyatlari ustuvor hisoblanishi uchun qo'llash mumkin. Agar ayni bir element uchun berilgan shartlarning qaysi birida «!important» bo'lmasa, o'z-o'zidan u e'tiborsiz qoldiri-

ladi. Quyida har ikkisida ham «!important» qo'yilgan hollarda aynan qaysi biri inobatga olinishini jadval tarzda ko'rib chiqamiz, chunki foydalanuvchi o'ziga xos shamoyil yarataman deb e'tibor bermagan holda dasturlovchi tomonidan o'rnatilgan maxsus shakllarni buzib yuborishi mumkin:

Foydalanuvchi jilosi



Muallif jilosi



Bunda kim tomonidan «!important» qo'llanilganligi «+» va «-» orqali, kataknинг to'qligi bilan qaysi biri inobatga olinishi ko'rsatilgan.

8.2. Umumiy sintaksis

CSSda ham e'lon qilish bo'limi mavjud bo'lib, odatda, uni yozish shart emas. Ushbu bo'limda qonuniyatlar qanday yozilganligi va qay tarzda tatbiq etish lozimligi ko'rsatiladi. Sintaksisi quyidagicha:

@buyrug qonuniyatlar

Misol:

```
@charset "utf-8";
```

Izoh sifatida yozilgan yoki brauzer e'tibor qaratmasligi lozim bo'lgan yozuvlar maxsus belgilar – «/*» va «*/» larning orasiga olib qo'yiladi.

Umuman olganda, CSS quyidagi sintaksis asosida bayon etiladi:

nomlanishlar { qoidalar }

Bunda nomlanishlar:

nomlanish [, nomlanish]

qoidalar:

qoida [, qoida]

qoida:

xususiyat : qiymatlar

qiymatlar:

qiymat [qiymat]

E'lon qilish bo'limidagi «qonuniyatlar» shaklan «qoidalar» ko'rinishida ham berilishi mumkin. Masalan, maxsus husnixat qo'shish:

```
@font-face {
    font-family: Consolas; /* Husnixat nomi */
    src: url(/font.ttf); /* Husnixat fayli */
}
```

Nomlanish, aslida, kelishilgan identifikator [2.6.] bo'lib, unda ikkinchi belgisidan boshlab chiziqcha (-)

qatnashishiga ruxsat etiladi. Bu nomlanishlarni guruhlashda foydalilanadi. Nomlanish quyidagi ma'nolardan birini anglatishi mumkin:

◆ **Universal nomlanish**. Sahifadagi barcha elementlarga taalluqli bo'lishi uchun nomlanish sifatida yulduzchadan foydalilanadi, u barcha nomlanish ma'nosini beradi, ba'zi hollarda uni ishlatmasa ham bo'ladi. Misol:

```
* { font-family: Times New Roman, sans-serif; }
```

◆ **Tegning nomi**. Bunda sahifadagi barcha shu nomdagi teglar hosil qilgan elementlar ko'rsatilgan qoidalar bo'yicha jilolanadi. Misol:

```
div { color: grey; } <div> GREY </div>
```

◆ **Tegning «id»si**. Bunda faqat «id»si shu identifikatorga teng bo'lgan obyekt tuslanadi, xolos. CSSda bunaqa nomlanish identifikator oldiga «#» belgisi qo'yish bilan boshlanadi. (Agar ma'lum tegning shu nomdagi identifikatori uchungina ushbu qoidalar tatbiq qilinishi lozim bo'lsa, teg nomidan so'ng panjara va identifikator hech qanday bo'sh joylarsiz nomlanish sifatida yozilishi kerak. Lekin bitta sahifada identifikator takrorlanmas bo'lishi talab etilgani bois bunday sintaksis deyarli ishlatilmaydi.) Misol:

```
#mydiv { color: grey; } <div id="mydiv">GREY</div>
```

◆ **Sinflash**. Ma'lum nom bilan HTML teglarini sinflarga ajratish mumkin. Buning uchun kerakli teglardiagi universal hisoblanuvchi class alomatiga e'lon qilingan nom beriladi. Nomlanish sinfni anglatuvchi identifikator bo'lsa, uning oldiga nuqta qo'yiladi.

Misol:

```
.sinf { color: grey; } <div class="sinf">GREY</div>
```

♦ Teqlarni sinflash. Agar ko'rsatilgan qoidalar faqat ma'lum teglar uchun tatbiq etilishi talab qilinsa, tegning nomini sinf identifikatoridan oldingi nuqtadan ham avval bo'sh joylarsiz yozish kerak: **p.myclass**. Agar obyektga bir paytda bir necha sinflarning qoidalari umumlashmasini tatbiq etish lozim bo'lsa, «class» alomatida ular bo'sh joy bilan ajratilib qo'shtirnoq ichida yoziladi. Aynan faqat bir necha sinf ko'rsatilgan teqlargagina taalluqli bo'lishi uchun nomlanish quyidagicha ko'rsatiladi:

```
.class1.class2 { color: grey; }
```

```
<div class="class1 class2"> GREY </div>
```

♦ Ichidagi teglar. Ba'zan ma'lum teqlarni boshqa-cha qoidalar asosida tuslashga to'g'ri keladi, agar ular boshqa bir tegning ichida yotsa. Ya'ni bir tegning ichida joylashgan boshqa teqlarni tuslash uchun ularning nomlari ketma-ket bo'sh joy bilan ajratib yoziladi. Misol:

```
div p { color: grey; }
```

```
<div><p> GREY </p></div> <p>TEXT</p>
```

♦ Qo'shni teg. Aynan bir xil darajadagi «chuqurlik» da joylangan bitta umumiyl tashqi teg ichidagilar qo'shni teglar hisoblanadi. Masalan, «head» tegi «body»ga qo'shni, lekin «body» tegi «title»ga qo'shni emas. Sababi, «html» asosiy birinchi darajali teg

hisoblansa, uning ichidagi ikkinchi darajadagilar bir-biriga qo'shni, uchinchi darajali chuqurlikdagi «title»ga ikkinchi darajadagi chuqurlikdagi «body» ham, uchinchi darajali «footer» ham qo'shni emas. Qo'shni teglar CSSda plyus (+) bilan ifodalanadi. Misol:

```
div + p { color: grey; }  
<div> BLACK </div> <p> GREY </p>
```

Bunda qoida faqat plyusdan keyin ko'rsatilgan nomdagi teg («p») uchungina tatbiq etiladi, agar «div» va «p» qo'shni teglar bo'lib, orasida boshqa element qatnashmagan bo'lsa.

♦ **Qardosh teglar**. Ular aynan «qo'shni teg» singari aniqlanadi, faqat yagona emas, bиринчи nomlanishdagi tegga nisbatan bir xil chuqurlikda joylashgan va undan keyin kelgan ikkinchi nomlanishga mos teglarning barchasiga tatbiq etiladi. Qardosh teglar «~» («tilde» — to'lqin) bilan ko'rsatiladi:

```
div ~ b { color: grey; }  
<div>BLACK</div>      <b>GREY</b>  
<p><b>BLACK</b></p> <b>GREY</b>
```

♦ **Ichki teglar**. «Ichidagi teglar»dan farqli o'laroq, «ichki teglar» nazarda tutilayotgan tegning ichida joylashadi, xolos. Ya'ni tarkibidagi boshqa teglarning ichki qismida bo'lsa, bu inobatga olinmaydi. Ko'zda tutilayotgan teg **n**-darajali chuqurlikda joylashgan bo'lsa, unga ichki teg **n+1** darajada bo'lishi shart. Masalan, «meta»lar «head»ga ichki teglar bo'lishi shart, «div»

esa «body»ga nisbatan bunday ma'suliyatni olmagan. Ichki teglar CSSda «>» belgisi bilan ifodalanadi. Misol:

```
div > p { color: grey; }

<div> <p>GREY</p> <b><p>BLACK</p></b> </div>
```

♦ **Alomatli**. Teglarning alomati va u qabul qilgan qiymatiga qarab jilolash uchun to'g'ri qavsdan foydalanildi. Bunday usulda jilo berishning bir qancha ko'ri-nishlari mavjud:

<u>nomlanish[alomat]</u>	{...}	«alomat»i (qiymati ahamiyatsiz) ko'rsatilganlar uchun
<u>nomlanish[alomat=qiymat]</u>	{...}	«alomat»i «qiymat»ga teng bo'lganlar uchun
<u>nomlanish[alomat^=qiymat]</u>	{...}	«alomat»i «qiymat» jumlasidan boshlanadiganlar uchun
<u>nomlanish[alomat\$=qiymat]</u>	{...}	«alomat»i «qiymat» jumlesi bilan yakunlanadiganlarga
<u>nomlanish[alomat*=qiymat]</u>	{...}	«alomat»i «qiymat» jumlasiga ega matndan iboratlarga
<u>nomlanish[alomat~=qiymat]</u>	{...}	«alomat» tarkibida (<i>bo'sh joy bilan</i>) «qiymat» mavjudlarga
<u>nomlanish[alomat =qiymat]</u>	{...}	«alomat»ning «qiymat»ida chiziqcha (-) bo'lganlar uchun

Misollar:

```
p[align] {
    font-weight: bold;
}
p[align="center"] {
    font-weight: bold;
}
p[title^="c"] {
    font-weight: bold;
}
p[align$="t"] {
    font-weight: bold;
}
p[align*="cent"] {
    font-weight: bold;
}
p[class~="bir"] {
    font-weight: bold;
}
p[align|= "bir"] {
    font-weight: bold;
}
```

<p align="right">**BOLD**</p>

<p>REGULAR</p>

<p align="center">**BOLD**</p>

<p align="left">REGULAR</p>

<p title="center">**BOLD**</p>

<p title="left">REGULAR</p>

<p align="right">**BOLD**</p>

<p align="left">**BOLD**</p>

<p align="center">**BOLD**</p>

<p align="left">REGULAR</p>

<p class="bir nom">**BOLD**</p>

<p class="birNom">REGULAR</p>

<p class="bir-nom">**BOLD**</p>

<p class="birNom">REGULAR</p>

Bir nechta teng kuchli nomlarni yozish lozim bo'lsa, ular yuqoridagi sintaksisda keltirilgani kabi vergul bilan ajratiladi. HTMLda esa shunchaki bo'sh joy tashlab yozilar edi. Masalan:

```
.my-div, .my-p {    <div class="my-div my-p">
    color: grey;        GREY
}

```

Shu o'rinda nozik bir nuqtaga e'tibor qaratish lozim. Agar bir necha sinf ko'rsatilgan bo'lsa, obyekt ulardagi qoidalarning umumlashmasi sifatida akslanadi. Aytaylik, ayni bir alomatga keltirilgan ikki sinfda turlicha qiymat berilgan bo'lsa, oxirgi ko'rsatilgan nomdag'i sinfdagi qoidalari emas, balki CSSda oxirgi e'lon qilingan sinf qoidalari tatbiq etiladi. Yuqoridaq misol bo'yicha, CSSda avval «my-p» e'lon qilingan bo'lsa, «class»da qanday ketma-ketlikda berilishidan qat'i nazar, avval «my-p» asosida akslantirilib, so'ng «my-div» qoidalari tatbiq etiladi.

8.3. Qiymatlarning turlari

CSSda ham qiymatlar HTMLda qo'llaniladigan aksariyat turlarni qabul qiladi. [2.6.] Faqat ayrim o'ziga xos qo'shimchalar kiritilgan.

- **inherit** — qiymatni tashqi tegning ayni xususiyatidan oladi. Agar tashqi tegga mos qiymat o'rnatilmagan bo'lsa, odatiy qiymatni qabul qiladi. Ta'kidlab o'tish lozimki, teg uchun ko'rsatilgan xususiyatlarning qiymatlari, odatda, uning ichidagi teglarga ham ta'sir qiladi, shunday ekan, ushbu qiymatni berish ko'pincha sama-

rasiz. Ammo ba'zan qiymatlar qo'shni teglar xususiyatlariiga qarab o'zgaradi, shunday hollarda ayrim xususiyatlar tashqi tegniki singari bo'lishini aniq ko'rsatib o'tishga to'g'ri kelishi mumkin.

- **initial** — asl qiymatini berish. Yuqorida ta'kidlanganidek, ichkaridagi teglar, odatda, tashqisidagidan xususiyatlarni o'zlashtirib oladi. Lekin ba'zi teg uchun, uning ko'rsatilgan xususiyatiga brauzerda belgilangan asl qiymat berish lozim bo'lsa, ushbu so'z ishlatiladi.

- **unset** — agar tashqi tegning mos xususiyatiga qiymat berilgan bo'lsa, uni «inherit» singari qabul qiladi. Aks holda, «initial» sifatida ishlaydi.

Ushbu uchala qiymatlardan birini qabul qiladigan **all** xususiyati avvaldan mavjud bo'lgan katta CSS fayllarni o'zgartirmasdan, undagi ba'zi qoidalarni bekor qilish uchun keng qo'llaniladi. Ayrim hollarda mavjud yirik loyihalarga kichik o'zgartirishlar qo'shish ham talab etiladi. Shunday vaziyatlarda oldindan kiritilgan jilolarni o'r ganib chiqib o'zgartirishga vaqt ko'p sarflanishi mumkin, «all» xususiti esa buni osongina hal eta oladi:

```
h3 { all: unset; color: blue; }
```

Rangning qabul qiluvchi qiymatlari HTMLnikidan ancha mukammalroq. Umuman olganda, rangni faqat CSS orqali berish kerak, bu obyektga jilo berishga kiradi va faqat alohida faylda ko'rsatilishi maqsadga muvofiq. Aslida, shakl-shamoyilni aniqlovchi kattaliklarning barcha CSS yordamida aniqlanishi tavsiya etiladi, lekin bu har doim ham barcha dasturlovchilarga ma'qul kelavermaydi. Shunday bo'lsa-da, tuslovchi omillar,

ayniqsa, rang faqat CSSda ko'rsatilishi qat'iy talab etiladi.

Rang nomi va RGB [2.6.] kodlanishdan tashqari qiymatni **rgb** va **rgba** funksiyalari [8.4.] bilan ham berish mumkin. Bunda ranglarning o'nlik sanoq tizimidagi mos qiymatlari hamda qay darajada shaffofligi ko'rsatiladi. Sintaksisi:

rgb(x,x,x) $x = 0..255$ oraliqdagi butun son yoki uning foizdagi ifodasi

rgba(x,x,x,a) $a = 0..1$ oraliqdagi shaffoflikni bildiruvchi haqiqiy son

HSL (Hue — soya, Saturate — qoramtil, Lightness — yorqin) funksiyasi yordamida ranglarning kodlarini aylana sifatida joylashtirib «rangli doira» hosil qilinadi va mos burchakdagi qiymatni RGB tartibida ko'rsatish bilan tanlanadi. Bunda har bir son 0..359 orasida yoki shunga nisbatan foiz hisobida beriladi. Sintaksisi:

hsl(s,n,l) $h = 0..359$ oraliqdagi butun son; **s** va **l** — 0% dan 100% gacha qiymat.

Bunga shaffoflik omili qo'shilib **hsla** funksiyasi hosil qilingan. Sintaksisi:

hsla(h,s,l,a) $h = 0..359$; **s,l** — 0% .. 100%; **a** — 0..1;

Bundan tashqari, **rang** foydalanuvchi sozlashlaridan olinishi mumkin. Agar keltirilgan nomga mos rang topilmasa, vazifasiga ko'ra eng yaqin element rangi olinadi. Bunday usul yaratilayotgan obyektlar rangini operatsion tizimda mavjudlariga mos holda olish bilan

sahifa tusini tez va mutanosib ravishda o'zgartirishga katta yordam beradi. Ushbu usul ayni veb-sahifaning turli brauzer va kompyuterlarda turlichay xilda namoyon bo'lishini ta'minlaydi. Har kimga har xil rang yoqadi, foydalanuvchining qurilmasida ko'zi o'rgangan ranglarda saytning namoyon bo'lishi unga nisbatan e'tiborni jalb qilishning bir ijobiy omiliidir. Shu o'rinda mavjud kalit so'zlar ro'yxatini keltirib o'tamiz:

- **activeborder** – faol oyna chegarasining rangi;
- **activecaption** – faol oyna sarlavhasining rangi;
- **appworkspace** – ko'p hujjatli oyna orttasviri;
- **background** – ishchi maydon orttasviri;
- **buttonface** – hajmlli obyekt yuzasining rangi;
- **buttonhighlight** – hajmlli obyektning to'yin-gan soyasi rangi;
- **buttonshadow** – hajmlli obyekt soyasi rangi;
- **butontext** – tugmadagi matn rangi;
- **captiontext** – sarlavhadagi matn rangi;
- **graytext** – o'chirilgan (nofaol) matn rangi;
- **highlight** – belgilangan obyekt rangi;
- **highlighttext** – belgilangan obyekt matnining rangi;
- **inactiveborder** – nofaol oyna chegarasining rangi;
- **inactivecaption** – nofaol oyna sarlavhasining rangi;
- **inactivecaptiontext** – nofaol sarlavha matnining rangi;
- **infotext** – suzuvchi izoh matnining rangi;
- **infobackground** – suzuvchi izoh orttasvirining rangi;

- **menu** – menyu orttasviri rangi;
- **menutext** – menyu matnining rangi;
- **scrollbar** – surgich rangi;
- **threeddarkshadow** – hajmli elementning qoramtil sojasini rangi;
- **threedface** – hajmli elementning old tomoni rangi;
- **threedhighlight** – hajmli elementning belgilash rangi;
- **threedlightshadow** – hajmli elementning och soyasining rangi;
- **threedshadow** – hajmli elementning to‘yin-gan sojasini rangi;
- **window** – oyna (orttasviri) rangi;
- **windowframe** – oyna hoshiyasining rangi;
- **windowtext** – oynadagi matn rangi.

8.4. Funksiyalar

CSSda qiymatlarning nafaqat kelishilgan o‘lchov birliklari yordamida, balki maxsus funksiyalar orqali ham berish mumkin. Shu o‘rinda **funksiya** nimaligi haqida tushunchasi bo‘lmagan mushtariylar uchun eslatib o‘tishni joiz topdik.

Ma‘lum amallar, hisob-kitoblar bajarib, qayta ishlash jarayonidan so‘ng maxsus natija qaytaradigan dasturning kichik bo‘lagi **funksiya** deyiladi. Unga qo‘srimcha ravishda kirish qiymatlari ham berilishi mumkin.

Yuqorida biz berilgan sonlarga mos rangning qiymatini qaytaradigan «**rgb**», «**rbga**», «**hsl**» va «**hsla**» kabi funksiyalar bilan tanishdik. Endi tasvirning ko'rinishini o'zgartiradigan maxsus funksiyalarni ko'rib o'tamiz.

8.4.1. Xiralashtirish

Obyektni akslantirishda ma'lum funksiyalar yordamida ko'rinishini o'zgartirish mumkin. Ular **filter** xususiyatiga maxsus nom va qiymatlar bilan ko'rsatiladi. Obyektning barcha elementlarini (Gauss usulida) xiralashtirib ko'rsatish uchun **blur** funksiyasi qo'llaniladi. Sintaksis:

```
filter: blur(olcham);
```

Ushbu funksiyani «body»dan tashqari ixtiyoriy ko'rindigan teg uchun qo'llash mumkin. Misol:

```
<div class="xira">
  
  <span>Raqami: xxxxxxx </span>
</div>
<p> Noma'lum shaxs surati </p>
```

```
.xira {
  filter: blur(8px);
}
```

8.4.2. Yorqinlashtirish

Tasvirlarning yorqinligini oshirish yoki kamaytirish uchun «brightness» funksiyasidan foydalaniadi. Oddiy yozuvlarga bu usulni qo'llash kutilgan samara bermaydi. Rasmning oddiy holatida yorqinlik 1 yoki 100% deb hisoblanadi va shunga nisbatan foizda yoki haqiqiy son bilan qism sifatida qiymat beriladi. Qiymat manfiy bo'lishi mumkin emas, agar ko'rsatilmasa, 1 deb hisoblanadi va hech narsa o'zgarmaydi. Sintaksis:

filter: **brightness(foiz)**;

8.4.3. Xijillashtirish

Tasvirni yorqinlashtirishga qarama-qarshi amal xijillashtirishdir, shuning uchun u ziddiyatlash ma'nosidagi «contrast» so'zi bilan nomlangan. Sintaksisi va unga beriladigan qiymat aynan yorqinlashtirishniki singari.

8.4.4. To'yintirish

Rasmdagi ranglarning bir-biriga yaqindan birlashtirish uchun **saturate** funksiyasi ishlataladi. Unga 0 qiymat berilsa, surat oq-qora ko'rinishga kelib qoladi. Bir qiymat berilganda esa o'zgarishsiz qoladi. Unga qancha katta qiymat berilsa, shuncha rasmdagi ranglar soni kamayib boradi. Sintaksis:

filter: **saturate(haqiqiy son)**;

8.4.5. Shaffoflik

Tasvirlarning shaffofligini ta'minlash uchun **opacity** funksiyasi ishlataladi. Shaffof obyekt o'z rangini yo'qotish bilan birga orqasidagi tasvirlarni ko'rishga imkon yaratadi, ya'ni tiniq oyna (ko'zgu emas) tasavvurini beradi. Sintaksis:

```
filter: opacity(foiz);
```

8.4.6. Soya qo'shish

Tasvirlarni jilolashda ularning ko'rinishidagi ta'sirini oshirish uchun soya qo'shish ham muhim ahamiyatga ega bo'ladi. Bunda obyekt, xususan, rasm to'g'ri to'rtburchak shaklida qiyofaga ega bo'lishi shart emas, soya uning chegarasi bo'ylab joylashaveradi. Sintaksis:

```
filter: drop-shadow(x, y, r, c);
```

Bunda **x** — soyaning yotiqligi surilishi, kiritilishi majburiy qiymat. Musbat son bo'lsa, o'ngga, manfiy bo'lsa, chapga ko'chadi; **y** — soyaning tikkasiga surilishi, majburiy qiymat. Musbat bo'lsa, pastga, manfiy bo'lganda tepaga ko'chadi; **r** — soyaning suvashtirish radiusi, qiymat berilmasa, 0 qabul qilinadi va bunda suvashtirilmagan tiniq holda ko'rindadi; **c** — soya rangi. Misol:

```
.soya {  
    filter: drop-shadow(3px 2px 6px red);  
}
```

```
<div class="soya">  
    
  <span> Olma </span>  
</div>
```

8.4.7. Oq-qora tasvir

Tasvirni oq-qora (monoxrom) tusga kiritish uchun **grayscale** funksiyasi qo'llaniladi. Unga foiz hisobida qiymat beriladi. Agar 1 yoki 100% berilsa, rasm to'laligicha oq-qora ko'rishiga o'tadi, qiymat berilmasa, 0 deb qabul qilinadi va tasvir rangi o'zgarmaydi. Sintaksisi:

```
filter: grayscale(foiz);
```

8.4.8. Rasm rangini o'zgartirish

Yozuvlarning rangini o'zgartirish juda sodda, lekin ba'zan rasmning rangini ham boshqacha tusda chiqarishga to'g'ri keladi. Bu bilan ayni bir suratni qayta-qayta saqlab xotirani behuda sarflashning oldi olinadi. Rasmning rangini butunlay o'zgartirish uchun «rangli doira»ga [8.3.] asoslangan **hue-rotate** funksiyasi qo'llaniladi. Sintaksisi:

```
filter: hue-rotate(burchak);
```

Obyektdagi akslanayotgan har bir nuqta o'z rangiga ega. O'sha rang kodining o'rniga maksimal qiymatga

nisbatan teskarisi berilsa, tasvir fotokamera plynokasida ko'rindigandek tus oladi. Shunday usul rangni ag'darish deb ham yuritiladi, bunday imkoniyatni invert funksiyasi taqdim etadi. Sintaksis:

```
filter: invert(foiz);
```

Qadimiy suratlar oq-qoradan tashqari jigarrahangsimon tusga ham ega bo'lgani uchun aynan shunday eski tasvir taassurotini berishda **sepia** funksiyasidan foydalaniadi. Sintaksis:

```
filter: sepia(foiz);
```

8.4.9. Orttasvir ranglari

CSS yordamida obyekt orqasiga rasm qo'yish uchun **url** funksiyasidan foydalangan holda **background-image** xususiyati ishlataladi. Sintaksis:

```
background-image: url(ishorat);
```

Ishoratda rasmning manzili ko'rsatiladi. Orttasvirni qay tarzda joylashtirish haqida quyida batafsil to'xtalamiz, hozir ushbu xususiyatga yana qanday funksiyalar yordamida qiymat berish mumkinligi bilan tanishib o'tamiz.

- **Chiziqli tuslanish.**

```
background-image: linear-gradient(yo'nalish, rang1, rang2);
```

Bunda obyektning orqa qismining tusi «rang1»dan «rang2»ga qarab o‘zgarib boradi, agar «yo‘nalish» ko‘rsatilgan bo‘lsa, aynan u bo‘yicha, aks holda, tepadan pastga tik ravishda ranglar almashiniladi. Yo‘nalish «burchak» hisobida [2.6.] yoki quyidagi inglizcha nomlanishlar bilan berilishi mumkin:

- **to top** – pastdan tepaga, tik (0deg).
- **to left** – o‘ngdan chapga, yotiq (270deg yoki 90deg).
- **to bottom** – tepadan pastga, tik (180deg).
- **to right** – chapdan o‘ngga, yotiq (-270deg yoki 90deg).
- **to top left** – pastki o‘ng burchakdan tepe chapga, yonbosh.
- **to top right** – pastki chap burchakdan tepe o‘ngga, yonbosh.
- **to bottom left** – tepe o‘ng burchakdan pastki chapga, yonbosh.
- **to bottom right** – tepe chap burchakdan pastki o‘ngga, yonbosh.

▪ **Yoysimon tuslanish.**

background-image: **radial-gradient(shakl uslub at joyi ranglar);**

Bunda «shakl»ning odatiy qiymati ellips bo‘lib, unga aniq «circle» yoki «ellipse» qiymati berilishi mumkin, «uslub»da shaklning radiusi beriladi yoki qanday o‘lchamni egallashini ko‘rsatadigan quyidagi so‘zlardan foydalilaniladi:

- **closest-side** – «shakl»ning chegarasi o‘zi joylashgan maydonning eng yaqin tomoniga tegadigan holatda chiziladi. Ellips bo‘lsa, ikkita yaqin tomonlar inobatga olinadi. Bunda shakl to‘laligicha akslanib, tashqi obyekt ichiga joylashadi.
- **closest-corner** – «shakl» chegarasi maydonning eng yaqin burchagiga tegadigan bo‘ladi.
- **farthest-side** – chegara maydonning uzoq tomonlariga nisbatan olinadi.
- **farthest-corner** – chegara maydonning uzoq burchagiga tegadigan tarzda chiziladi.

Shaklning markazi qayerda bo‘lishi «joyi»ga yotiq va tik yo‘nalishlar bo‘yicha piksel yoki foiz hisobida qiymat berish bilan ko‘rsatiladi. Albatta, «joyi» standart o‘rnlardan birini egallashi lozim bo‘lsa, qiymatni «top», «bottom», «right», «left», «center» kabi inglizcha so‘zlar yordamida bersa ham bo‘ladi.

Nechta rang o‘zgarib borishi orqali akslantirish lozim bo‘lsa, vergul bilan ajratilgan holda shuncha «ranglar» berish mumkin.

Misol:

```
background-image: radial-gradient(
  circle farthest-corner at top left, cyan, aqua, blue
);
```

▪ **Takroriy tuslanish.**

Maydon yuzasi katta bo‘lganda tuslanishda uning ko‘p qismi ayni bitta rang bilan tasvirlanib qoladi.

Bunday paytlarda qonuniyatni takrorlash yaxshi samara beradi. Buning uchun chiziqli va yoysimon tuslanishlarni takrorlab akslantiradigan **repeating-linear-gradient** va **repeating-radial-gradient** maxsus funksiyalari mavjud. Ularga beriladigan qiymatlar mos funksiyasini bilan bir xil. Faqat takrorlanishni ta'minlash uchun o'zgarib boradigan, ya'ni ikkinchi rangdan boshlab keyingilariga qancha qalinlikni egallashi ko'rsatib borilishi lozim.

Misol:

```
background-image: repeating-radial-gradient(  
    circle at top left, cyan, aqua 10px, blue 80px  
) ;
```

8.4.10. Hisoblangan qiymat

Juda ko'p hollarda obyektlarning o'lchamini va joylashish o'rmini ko'rsatish uchun tashqi tegning aksangan maydoni hajmini inobatga olishga to'g'ri keladi. Foizlarda qiymat berish doim ham kutilgan samara bermaydi. Bunday vaziyatlarda maxsus calc funksiyasi yordamida hisoblanadigan qiymatni xususiyatga berishga to'g'ri keladi. Unda to'rt arifmetik amal: qo'shish, ayirish, ko'paytirish va bo'lishlarni amalga oshirish mumkin. Unutmaslik lozimki, «+» va «-» ishoralarini har ikki tomonidan bo'sh joy bilan ajratilishi shart! Aks holda, brauzer uni tushunmaydi. Umuman olganda, barcha amallarni bo'sh joy bilan ajratib yozish tavsiya etiladi.

Misol:

```
width: calc(100% - 20px);  
height: calc(100% / 2);
```

8.5. Tanlov asosida biriktirish

Avval veb-sahifa namoyon bo'ladigan qurilma (media) turlarini eslab o'tsak [2.4.]:

- **all** – barcha qurilmalar
- **aural** – tovushli brauzerlar
- **braille** – Brayl tizimli qurilmalar, ko'rilar uchun
- **handheld** – uyali (kichik ekranli) qurilmalar
- **print** – chop etish qurilmasi
- **projection** – proyektor
- **screen** – monitor
- **tv** – televizor

Yuqorida kelishib olganimizdek, CSS qonuniyatlarini alohida faylga yozamiz. Zarur bo'lganda kerakli sahifaga aynan kerakli «css»ni yuklab foydalanaaveriladi. Bundan tashqari, bitta «css» fayl ichida boshqasini chaqirib birlashtirib olish imkoniyati ham mavjud. Buning uchun quyidagicha sintaksis qo'llaniladi:

```
@import "«css» fayl manzili" media turi;
```

Misol:

```
@import "/style/main.css" screen;  
/* kompyuterlar uchun */  
@import "/style/smart.css" print, handheld;  
/* chop etish yoki smartfonlar uchun */
```

Nafaqat fayllarni saralash asosida biriktirish, balki qonuniyatlarning o'zini ham tanlov bo'yicha yozish mumkin. Misol:

```
@media screen { /* kompyuterlar uchun */  
    BODY {  
        color: darkblue; /* matn rangi to'q ko'k */  
    }  
}  
@media print { /* chop etish uchun */  
    BODY {  
        color: black; /* matn rangi qora */  
    }  
}
```

8.6. Tabaqalash

Odatda, CSS qoidalari HTML teglarini ma'lum sinflarga ajratib, shular bo'yicha tatbiq etish asosida ishlaydi. Lekin bunday taqsimlash [8.2.] doim ham yetarli emas. Masalan, o'nlab ishoratlar ro'yxatiga ega

veb-sahifani qaraydigan bo'lsak, yana qayta bosib vaqt sarflamaslik uchun ko'p hollarda qaysilariga tashrif buyurilganligini ajratib ko'rsatish talab etiladi. Bunga yorqin misolni global izlash xizmatlarini taqdim etuvchi saytlarda ko'rganmiz (google.com, yandex.ru va bosh-qalar).

Ayni bir xil sinfga oid obyektlarni joriy holatiga qarab bir necha tabaqaga ajratib, har biriga o'ziga mos CSS qoidalar berish mumkin. Buning uchun «nomlanish»dan so'ng ikki nuqta (:) va holat nomi bo'sh joylarsiz yoziladi:

♦ **:active** — obyektning faollashgan holati. Ko'rsatkich (kursor) bilan tanlanib, sichqoncha bosib turilgandagi (hali qo'yib yubormasdan avvalgi) holat. Misol:

```
div:active { color: maroon };
```

♦ **:checked** — «radio» va «checkbox» darchalarining belgilangan holati.

♦ **:default** — aynan bir xil omillar bilan berilgan teglar ichidan odatiy qiymat sifatida qabul qilingan elementning holati. Masalan, bitta forma ichida bir necha yuboruvchi «submit» tugmalari qo'yilgan bo'lsa, kiritish («Enter») bosilganda, odatda, ularning birinchisi ishga tushadi, ya'ni u «:default» holatida bo'ladi.

♦ **:disabled** — nofaol holat. Foydalanuvchi bunday elementlar qiymatini o'zgartira olmaydi, tugma bo'lsa, bosish imkonsiz, ya'ni o'z faoliyatini bajarmaydi. Bunday obyektlar, odatda, xiralaشتirib ko'rsatiladi, yana qandaydir qo'shimcha jilo berish lozim bo'lsa, ushbu tabaqadan foydalaniladi.

♦ **:empty** — teglarning ichida hech qanday element, hattoki, bo'sh joy yoki ko'rilmaydigan belgilari ham bo'limgan holati.

♦ **:enabled** — nofaollashtirilmagan elementlarga.

♦ **:first-child** — tashqi tegning ichida birinchi joylashgan element uchun. Bunday holat bir nechta qo'shni obyektlarning birinchisi uchun boshqacharoq jilo berish lozim bo'lganda asqatadi.

♦ **:last-child** — tashqi tegdagi eng oxirgi elementgagina tatbiq etiladi.

♦ **:first-of-type** — tashqi teg sinfiga oid barcha teglarning birinchi elementi uchun tatbiq etiladi. Masalan, jadval yaratilayotganda har bir kataknini anglatuvchi «td» tegi «tr»ning ichida yotadi. Agar-da nomlanishni «td:first-of-type» ko'rinishida berilsa, har bir qatordagi ilk katak uchun qo'llanilgani bois jadvalning birinchi ustuni ko'rsatilgan o'zgacha jilo bo'yicha akslanadi.

♦ **:last-of-type** — tashqi teg sinfidagi barcha teglar ichidagi oxirgi elementga tatbiq etiladi.

♦ **:focus** — fokus qabul qilgan element holati, ya'ni «tab» tugmasini bosish yoki ko'rsatkichni ichiga qo'yish orqali tanlangan element uchun tatbiq etiladi.

♦ **:fullscreen** — ushbu holat brauzer to'liq ekranga yoyilgan holda akslantiriladi.

♦ **:hover** — «sichqoncha» ko'rsatkichi obyekt ustiga borgan, lekin hali bosilmagan holat. Shuni unutmaslik kerakki, ushbu holatning ta'siri «:active» bo'lganda ham yo'qolmaydi. Ya'ni obyekt sichqoncha bilan bosilar ekan, albatta, ko'rsatkich (kursor) uning ustida bo'ladi.

Shunday xususiyat mavjudki, u faqat ko'rsatkich obyektning ustiga kelganda akslantiriladi. Uni «hover»da berish shart emas, masalan: **cursor**. Qiymati aniqlangan tur bo'lib, quydagilardan birini qabul qiladi:

- **url** – ko'rsatkich rasmining manzili beriladi;
- **auto** – ko'rinishini obyekt turiga qarab o'zi aniqlaydi;
- **none** – ko'rsatkichni yashiradi;
- **default** va context-menu, help, pointer, wait, progress, cell, all-scroll, crosshair, text, alias, vertical-text, not-allowed, no-drop, copy, move, col-resize, row-resize, n-resize, ne-resize, grab, e-resize, se-resize, s-resize, sw-resize, w-resize, nw-resize, nesw-resize, nwse-resize, zoom-out, zoom-in, grabbing – operatsion tizimda aniqlangan mos rasmlarni akslantiradi.

♦ **:in-range** – obyektga berilgan qiymat uning «min» va «max» alomatlarida ko'rsatilgan oraliqda bo'lganda.

♦ **:out-of-range** – obyektga foydalanuvchi bergen qiymat uning «min» va «max» alomatlarida ko'rsatilgan oraliqdan tashqarida bo'lgan holatda. Misol:

```
<input type="number" value="30" max="359" min="0"
       placeholder="Burchakni kiritting">
input:in-range      { background: #e2edc1; }
input:out-of-range { background: #f9c3d2; }
```

♦ **:indeterminate** – belgilanuvchi darchaning noaniq holati, ya'ni «checkbox»ning qiymati rost ham, yolg'on ham emas.

♦ **:invalid** — kiritilgan qiymati darchaning turiga mos bo'lмаган holat. Masalan, pochta manzili оrniga oddiy ishorat kiritilsa.

♦ **:lang** — tegning «lang» alomatida ko'rsatilgan til mos kelgan holat. Misol:

```
<q lang="ru"> Надежный источник </q>
<q lang="uz"> Mukammal qo'llanma </q>
q:lang(ru) { color: darkred; }
q:lang(uz) { color: maroon; }
```

♦ **:link** — hali tashrif buyurilmagan ishorat holati.

♦ **:not** — alomatning qiymati ko'rsatilgandagidek bo'lмаган elementlar uchun tatbiq etiladi. Misol:

```
<input type="button" value="Bosish">
input:not([type="submit"]) { color: maroon; }
```

♦ **:nth-child(qiymat)** — Ko'rsatilgan tegning ichidagi elementlar tartib bo'yicha sanalganda **n**-lariga tatbiq etiladi, unda «qiymat» quyidagilardan birini qabul qiladi:

- **odd** — barcha toq o'rindagi elementlari uchun;
- **even** — barcha juft o'rindagi elementlarga;
- **butun son** berilganda o'sha o'rinda joylashgan elementga;
- «**n**» qatnashgan arifmetik ifoda berilganda **n** natural son sifatida sanalib, o'sha ifoda qiymati bo'yicha o'rinni egallagan elementga. Misol:

```
tr:nth-child(3n) { background-color: #f0f0f0; }
```

♦ **:nth-last-child** — ushbu holat «:nth-child» singari bo'lib, faqat sanoq oxirgi elementdan teskari tartibda

amalga oshiriladi. Jadvaldag'i qatorlarni bir-biridan farq-lab turish uchun bu usulni qo'llash yaxshi natija beradi.

♦ **:nth-of-type** — tashqi teg sinfidagilarning **n**-elementlari uchun qo'llanadi. Bunda sanoq tartibini «**order**» xususiyati bilan o'zgartirish mumkin.

♦ **:nth-last-of-type** — «**:nth-of-type**»ni teskari sanoq bo'yicha tatbiq etadi.

♦ **:only-child** — ko'rsatilgan sinfda tashqisining ichida yagona bo'lgan teg holati.

♦ **:only-of-type** — tashqi sinfga oid teqlarning ichida ko'rsatilgandek yagona bo'lgan holatlarda tatbiq etiladi.

♦ **:optional** — «**required**» alomati ko'rsatilmagan darchalar uchun qo'llaniladi.

♦ **:read-only** — «**readonly**» alomati berilgan, o'zgartirish mumkin bo'lмаган holat.

♦ **:read-write** — qiymatini o'zgartirish mumkin bo'lgan darchalar uchun.

♦ **:required** — «**required**» alomati ko'rsatilgan darchalarga tatbiq etiladi.

♦ **:target** — «**id**»si veb-sahifaning manzilini ko'rsatuvchi ishoratda «#»dan so'ng berilgan identifikorga teng bo'lgan teglar uchun qo'llaniladi. Misol:

URL <http://yurt.uz/?a=4#hikoyalar>

CSS **h2:target { font-weight: bold; }**

<h2 id="sherlar"> She'rlar </h2>

HTML **<h2 id="hikoyalar"> Hikoyalar </h2>**

<h2 id="hikmatlar"> Hikmatlar </h2>

Natija She'rlar **Hikoyalar** Hikmatlar

♦ **:valid** — foydalanuvchi tomonidan kiritilgan qiymati aniqlangan turga va cheklovga mos tushgan darchanning holati.

♦ **:visited** — foydalanuvchi tomonidan tashrif buyurilgan ishoratlar holati.

8.7. Harakatlantirish

Bir paytlar «Internet Explorer» brauzeri yuguruvchi qatorni akslantirish uchun **marquee** tegini o'ziga xos alomatlar bilan kiritgan edi. Keyinchalik boshqa brauzerlar ham uni o'zlariga tatbiq eta boshladи. E'tibor qaratish kerakki, ushbu teg HTMLda mavjud bo'lma-gan, HTML5da esa uni qo'llash mutlaqo tavsiya etilmaydi. Ammo bundan ham ancha mukammal tarzdagi harakatlarni ifodalash uchun CSSning o'z imkoniyatlari yetarli.

8.7.1. Harakatlantiruvchi xususiyatlar

Harakatni boshqarish uchun «**animation**» xususiyati dan foydalaniladi. Uning kengaytirilgan imkoniyatlari bilan batafsил tanishamiz:

- **Nom berish – animation-name: *nom***

Bunda «*nom*» identifikator bo'lib, unga murojaat qilishni ta'minlaydi. Agar nom o'rnida «none» qiyamat berilgan bo'lsa, harakatlanishni o'chiradi. Ayni bir qoi-

dalar uchun turli xilda harakatlanish paydo qilish kerak bo'lsa, «nom»ga bir necha identifikator vergul bilan ajratilgan holda beriladi va u harakatlanishga bog'langanda mos nomga murojaat qilinadi.

▪ **Bajarilish vaqtı – animation-duration:** vaqt

Boshlang'ich holatdan oxirgisiga qancha vaqtda o'tish lozimligini ko'rsatadi, «nom» bir nechta bo'l-ganda har biri uchun alohida vergul bilan ajratib vaqt berish mumkin. Odatiy qiymati: «0s», nol soniya, harakat sezilmaydi.

▪ **Harakatlanish tezligi xususiyati sintaksisi:**

animation-timing-function: qiymat

Harakatlanishning bir maromdag'i tezlikda bajarilishi doim ham yaxshi samara bermaydi. E'tibor jalg etiladigan vaziyatda sekinlashib, keyin obyektning tezroq ko'rinish sohasidan chetlatilgani ma'qul. Buning uchun fizika fanidan bizga tanish bo'lgan tezlanish tushunchasi ishlataladi, ammo uning ham o'zgarmas bo'lishi kutilgan natijani bermaydi. Shuning uchun Bezier algoritmi bo'yicha u o'zgartirib boriladi, **cubic-bezier(x₁, y₁, x₂, y₂)** funksiyasi [6.4.1.] ko'rsatilgan nuqta va koordinatalar boshi bo'yicha hosil qilingan to'g'ri to'rtburchak ichiga chiziladigan yoy o'zgarishi bo'yicha qiymat oladi. Ba'zi standart tezlik o'zgarishlarini maxsus nomlar bilan ham berish mumkin:

- **ease** – odatiy qiymat, harakat sekin boshlanib, keyin tezlashadi va oxirida yana sekinlashadi, cubic-bezier(0.25, 0.1, 0.25, 1) qiymatiga teng.
- **ease-in** – sekin boshlanib, oxirida tezlashadi, cubic-bezier(0.42, 0, 1, 1) singari.

- **ease-out** – tez boshlanib, oxirida sekinlashadi, cubic-bezier(0, 0, 0.58, 1) singari.
- **ease-in-out** – boshlanish va yakunida sekinlashadi, cubic-bezier(0.42, 0, 0.58, 1) singari.
- **linear** – harakat boshidan oxirigacha bir xil tezlikda bo'ladi.
- **step-start** – harakatlanish bo'lmaydi, darhol yakuniy joylashuvga o'tadi.
- **step-end** – harakat bo'lmaydi, ko'rsatilgan vaqt tugagach yakuniy nuqtaga ko'chadi.
- **steps(son, start | end)** – uzluksiz harakat bo'lmaydi, ko'rsatilgan «son» miqdorida sakrashlar amalga oshiriladi. Bunda boshlang'ich va oxirgi joylashuv nuqtalarini tutashtiruvchi kesma teng miqdorda «son»ta qismga ajratiladi va hosil bo'lgan nuqtalarda obyekt ketma-ket ko'chirib o'tiladi. Agar bo'linganda butun son chiqmasa, «start» so'zi yozilgan holda, hosil bo'lgan haqiqiy sonning butun qismi olinadi, «end» deb ko'rsatilganda esa, undan bir birlik kattasi olinadi.

▪ **Kutish vaqtি – animation-delay: vaqt**

Ko'rsatilgan vaqtdan keyin harakat boshlanadi. Agar «nom» bir nechta bo'lsa, ularning har biriga mos boshlang'ich kutish vaqtini vergul bilan ajratilgan holda berish mumkin. Vaqt soniya yoki millisoniya hisobida beriladi. [2.6.]

▪ **Takrorlanish soni** quyidagi sintaksisda aniqlanadi:
animation-iteration-count: infinite | son

- **infinite** – harakatlanish to'xtovsiz davom etadi.

- «**son**» necha marta harakatlanish bajarilishini ko'rsatadi. Manfiy qiymat berish mumkin emas. Agar 0.5 ko'rsatilsa, harakatlanishning yarmi bajariladi.

- **Harakat yo'nalishi** sintaksisi:

animation-direction: normal | alternate | reverse | alternate-reverse

- **normal** — harakatlanish yakunlangach, darhol boshlang'ich holatga o'tadi.
- **alternate** — harakat yakunlangach, teskari yo'nalishda ortga qaytadi.
- **reverse** — harakatlanish teskari yo'nalishda bajariladi.
- **alternate-reverse** — harakat teskari yo'nalishda boshlanadi va yakunlangach, ortga qaytadi, ya'ni ko'rsatilgan yo'nalishda amalga oshiriladi.

- **Harakat yakunlanishi** sintaksisi:

animation-fill-mode: none | forwards | backwards | both

- **none** — harakatlanish yakunlangach, darhol boshlang'ich holatga o'tadi.
- **forwards** — harakat yakunlangach, oxirgi joylashuvda qoladi.
- **backwards** — harakatlanish yakunlangach, boshlang'ich holatga qaytadi.
- **both** — harakat yakunlangan holatda qotadi. forwards va backwards qiymatlaridagi holatlar nisbiy bo'lib, ular «animation-iteration-count» va «animation-direction» xususiyatlarida ko'rsatilgan qiymatlarga bog'liq. Masalan, yo'nalishda teskari qiymat berilgan bo'lsa, oxirgi joylashuv teskari olinadi.

- **To'xtatib turish** qiyidagicha belgilanadi:
animation-play-state: paused | running

- **paused** – harakatlanishni to'xtatish;
- **running** – harakatlanishni davom ettirish;

Sichqoncha yordamida obyekt tanlanganda yoki bosilganda hakaratlanishni to'xtatib, keyin kerak bo'lganda o'sha joyidan davom ettirish uchun bu usul keng qo'llaniladi.

8.7.2. Harakatlanishni ta'minlash

Yuqorida ko'rib o'tgan xususiyatlarimiz harakatlanish qanday tarzda amalga oshishini belgilaydi, ammo harakatlanishning o'zini ta'minlamaydi. Buning uchun uning «**@keyframes**» buyrug'idan foydalaniлади.

```
@keyframes nom {  
    from { qidalar }  
    to   { qidalar }  
}
```

Bunda «nom» «animation-name»da ko'rsatilgan identifikator, «from»dan keyin boshlang'ich holatni aniqlovchi CSS qoidalar, «to»dan keyin oxirgi holat ko'rsatiladi. Boshlang'ich va oxirgi holatni ifodalash uchun «from» va «to» so'zlarining o'rniga mos ravishda «0%» va «100%» sifatida ko'rsatish mumkin. Shunday ekan, «animation-duration»da ko'rsatilgan vaqtga nisbatan ixtiyoriy foiz hisobida o'sha paytdagi vaziyatning qoi-

dalarini ko'rsatish mumkin. Agar «0%» ko'rsatilmasa, unda harakat darhol boshlanmaydi.

Misol tariqasida 100px o'lchamdagি kvadratning avval enini to'rt marta oshirib, keyin bo'yini ikki marta uzaytiradigan harakat cheksiz takrorlanishini ko'ramiz.

```
<div class="quti"></div>
.quti {
    background-color: green;
    animation-name: kengay;
    animation-duration: 2s;
    animation-direction: alternate;
    animation-iteration-count: infinite;
}
@keyframes kengay {
    0% { width: 100px; height: 100px; }
    50% { width: 400px; height: 100px; }
    100% { width: 400px; height: 200px; }
}
```

Har bir «from», «to» yoki foizda ko'rsatilgan holat yoziladigan qoidalar ichida bir xilda xususiyatlar ko'rsatish tavsiya etiladi, aks holda, uni harakatlantirish vaqtida boshlang'ich qiymatlariga qarab yaratiladi va kutilgan natija akslantirilmasligi mumkin.

8.7.3. Xususiyatlarlarni umumlashtirish

CSSda kengaytirilgan xususiyat qiymatlarining har birini alohida berish o'rniغا uni umumiy nomlanishda

birlashtirish imkoniyati mavjud. Ya'ni quyidagilar bir xil ma'noni anglatadi:

```
animation-name: harakat;  
animation-delay: 800ms;  
animation-direction: alternate;  
animation-iteration-count: infinite;  
animation-timing-function: ease-in;
```

/ xato umumlashtirish */*

animation: harakat 800ms alternate infinite ease-in;

/ to'g'ri umumlashtirish */*

animation: harakat ease-in 800ms infinite alternate;

Qiymatlar bo'shliq bilan ajratib umumlashtirib berila-yotganda ularning ketma-ketligi ushbu mavzuda tavsiflangani bo'yicha kelishi talab etiladi. Agar biror xususiyat uchun odatiy qiymat qanoatlantirsa, uni ko'rsatmasdan tushirib qoldirish mumkin. Agar umumlashtirilgan qiymatlarda ketma-ketlik o'zgarsa, brauzer berilgan miqdorning yozuviga qarab mos xususiyatga biriktirishga urinadi, ammo e'tibor berish lozimki, ushbu holatda «animation-delay» va «animation-duration»lar bir xil turdag'i vaqt qiymatini qabul qiladi. Demak, misolda keltirilgan «800ms» vaqt «animation-delay»ga emas, «animation-duration»ga beriladi.

Albatta, umumlashtirilgan qiymat berish usuli yozuvni qisqartirish bilan birga kod yozilgan fayl hajmini kamaytiradi. Bu internet trafik hajmi kam bo'lishiga olib keladi. Ma'lumot tez yuklanishi bilan birga uni qayta ishlash jarayoni ham ancha tezlashadi. Ammo u kam-

chilikdan xoli emas. Masalan, «animation-name»da bir nechta nom ko'rsatish lozim bo'lsa, unga mos boshqa xususiyatlarga ham, shart bo'lmasa-da bir necha qiymat vergul bilan ajratilgan holda berilishi kerak bo'ladi. Umumlashtirilgan qiymatlardagi yozuvlar ko'paygan sari uni mos xususiyatlarga bo'lib taqsimlashda chalkashliklar yuzaga keladi. Shuning uchun bitta xususiyatda bir necha qiymat beriladigan holda qiymatlarni umumlashtirish tavsiya etilmaydi.

Quyida xususiyatlar yoritilayotganda, ularni umumlashtirishda beriladigan qiymatlar ketma-ketligi muhim bo'lsa, qay tarzda keltirish lozimligi alohida bayon etilgan.

8.7.4. Sodda harakatlantirishlar

Aslida, tabaqlash [8.6.] qo'llanilgan obyektlar, ma'lum ko'rsatilgan holat uchun boshqa qoidalar berilgan bo'lsa, o'sha hodisa bajarilgan paytda darhol o'zgarishlar amalga oshirilishi orqali keskin harakatlanish bajariladi. Misol uchun, obyektning ustiga ko'rsatkich keltirilganda (:hover) yoki uzluksiz bosib turish orqali faollashtirilganda (:active) yangi qoidalarga darhol o'tish o'rniga harakatlanish (animation) joylash ham mumkin. Bunday hollarda doim «@keyframes»ni qo'llash qulay emas. Xususiyatlar qiymatini o'zgartirish orqali paydo qilinadigan bunday harakatlanishni ifodalashda «animation» o'rniga soddalashtirilgan talqini — «transition»dan foydalaniadi. U quyidagi kengaytirilgan xususiyatlarni o'z ichiga oladi.

- **Xususiyat qiymatini o'zgartirish:**

- transition-property:** none | all | xususiyat
- **none** — hech bir xususiyat qiymati bo'yicha o'zgartirilmaydi.
 - **all** — barcha yozilgan xususiyatlar o'zgaradi.
 - «**xususiyat**» nomli qiymatigina o'zgaradi. Agar bir necha xususiyat bo'yicha o'zgarishi talab etilsa, ular vergul bilan ajratib yoziladi.

- **Samarali harakatlanish vaqtி:**

transition-duration: vaqt

Samarali qo'shimcha harakat qancha vaqt davomida bo'lishini ko'rsatadi, odatiy qiymati 0, ya'ni harakatlanish ko'rinxmaydi va faqat ko'chish sodir bo'ladi.

- **O'tish vaqtி tezligi:**

transition-timing-function: qiymat

Bu samarali harakatlanishning sodir bo'lish vaqtı o'zgarishi «animation-timing-function» singari aniqlanadi. [8.7.1.]

- **Kutish vaqtி. transition-delay: vaqt**

Samarali harakat boshlanishidan avval qancha kutish lozimligini ko'rsatadi. Vergul bilan ajratib bir necha qiymat berish mumkin, ularning har biri mos ravishda «transition-property»da ko'rsatilgan xususiyatlar uchun tatbiq etiladi. Misol:

```
#tepa {  
    background-color: green;  
    height: 20px;  
    transition: height 2s ease-out 200ms;  
}
```

```
#tepa:hover { height: 120px; }  
<div id="tepa"></div>
```

8.7.5. Harakatlantirish samaralari

Harakatlantirishni shunchaki boshlang'ich holatdan oxirisiga ma'lum vaqtda tekis siljishlar orqali amalga oshirmsandan, orada qo'shimcha o'zgarishlar, ya'ni samaralar qo'shish ham mumkin. Buning uchun ko'rinishini o'zgartiruvchi funksiyalardan foydalilanladi. Avval qaysi nuqtaga nisbatan o'zgarish amalga oshirilishini aniqlab olish lozim, buning uchun quyidagi xususiyat qo'llaniladi:

transform-origin: x y z

Bunda Dekart koordinatalar tizimiga mutanosib mos o'qlar bo'yicha qiymat kiritiladi. Odatda, «50% 50% 0» singari qiymat qabul qiladi. Joylashishni anglatuvchi «top», «right», «bottom», «left», «center» so'zlarini «x» va «y» qiymatlarining o'rniga qo'llash mumkin.

Ichki obyektlar uch o'lchamli fazoda qanday akslanrilishi «transform-style» xususiyatida ko'rsatiladi. Odatiyligi qiymati: **flat**, ya'ni tashqi element bilan bitta tekislikda. Bundan tashqari, «preserve-3d» qiymatini qabul qilishi mumkin. Uch o'lchamli shaklni fazoviy aylantirishda uning orqa tarafi «backface-visibility» xususiyati yordamida ko'rsatiladi. U «visible» (orqa tarafi ham ko'rinarli, odatiy qiymat) va «hidden» (orqasi ko'rinsandan, oldi bilan bir xil akslanadi) qiymatlaridan birini qabul

qiladi. Albatta, bu xususiyatning ta'siri ikki o'lchamli elementlar uchun sezilarli bo'lmaydi.

Ko'rinishni o'zgartiruvchi funksiyalar «**transform**» xususiyatiga qiymat sifatida beriladi. Bir necha funksiya ko'rsatilishi kerak bo'lsa, bo'sh joy bilan ajratiladi.

Obyektni turgan joyidan siljитish uchun quyidagi funksiyalardan mosini qo'llash mumkin: **translate3d(x,y,z); translate(x,y); translateX(x); translateY(y); translateZ(z);**

```
<div class="suril"></div>
.suril { transition-duration: 2s; }
.suril:hover { transform:translate(120px,20px); }
```

Obyektning masshtabini mos yo'naliшlar bo'yicha o'zgartirish uchun esa quyidagi funksiyalar ishlataladi: **scale(x,y); scaleX(x); scaleY(y); scaleZ(z); scale3d(x,y,z);** Misol:

```
<div class="kattalash"></div>
.kattalash { transition-duration: 2.3s; }
.kattalash:hover { transform: scale(4,3); }
```

Masshtabni o'zgartirish uchun oldin **zoom** xususiyati ishlatalgan, uning kamchiligi shundaki, faqat bitta o'lcham asosida (eni va bo'yi alohida berilmaydi) o'zgartiradi va namoyish etilayotganda joylashgan boshlang'ich nuqtasining koordinatasi saqlanib qoladi. Bu esa yonidagi obyektlarning surilishiga olib keladi. Uning vazifasini to'laqonli bajara oladigan «**transform**» qabul qiladigan «**scale**» funksiyasi esa hajmi o'zgarayotgan obyektning markaziy nuqtasiga nisbatan oladi va boshqa elementlar joylashishiga ta'sir o'tkazmaydi. Shu bilan

birga, «zoom»ni qo'llamaydigan «FireFox»da ham ishlaydi.

Obyektni ko'rsatilgan o'qqa nisbatan «a» burchakka burish uchun **rotate(a);** **rotateX(a);** **rotateY(a);** **rotateZ(a);** **rotate3d(x,y,z,a);** funksiyalari, tik yoki yotiq ravishda ma'lum burchakka og'dirish uchun **skewX(a);** **skewY(a);** va **skew(ax, ay);** funksiyalari ishlataladi.

Bundan tashqari, tasvir nuqtalarini matritsa yordamida qayta aniqlab akslantiruvchi funksiyalar ham qo'llash mumkin. [6.4.3.]

Yuqorida o'rgangan bilimlarimizni tatbiq etgan holda 200px o'lchamli kvadrat maydonni egallagan olma suratini kursor ustiga borgan vaqtida bir dumalatib bo'yoqli tasvir holatiga keltirib qo'yadigan CSS yozib ko'ramiz:

```

.dumala { transition-duration: 2.7s; }
.dumala:hover {
    filter: invert(80%);
    transform: translate(200px, 10px) rotate(1turn);
}
```

8.8. Qatorli va blokli teglar

HTML tilida teglar uch turga bo'linadi. Birinchisi — **tavsif teglari**, ma'lumot uchun qo'llaniladi. Ular brauzer-

da namoyon bo'lmaydi, akslantiriladigan teglar qanday qonuniyatlar asosida paydo bo'lishi lozimligini aniqlaydi. Ularga «head» va uning ichidagi teglardan tashqari quyidagilar ham kiradi:

**
** — Yangi qatorga o'tishni ta'minlash;

<wbr> — Matn obyekt ichiga sig'magan holda keyingi qatorga tushishini ta'minlash.

Ikkinchisi — **blokli teglar**, ular bilan quyida tanishamiz. Uchinchisi — **qatorli teglar**, matnning ayrim qismalarini alohida jilo bilan ko'rsatish uchun mo'ljallangan. Agar matn ichidagi ikki so'zni ajratib ko'rsatish lozim bolsa, ular mos qatorli tegga olinganda, blokli tegdan farqli o'laroq, so'zlarning joylashishiga qarab bittasi yuqori, ikkinchisi keyingi qatorga tushib qolishi mumkin. Qatorli tegdagagi birinchi satrning quyi qismi «**asos chizig'i**» deb yuritiladi, u husnixat o'lchami va nomiga qarab turlicha balandlikda bo'lishi mumkin. Qatorli teglarning eni ichidagi elementning hajmiga mos kenglikdagi joyni egallaydi, ular ichma-ich qo'llanilishi mumkin. Qatorli teglarga quyidagilar kiradi:

<a> — Ishorat joylashtirish.

<abbr> — Qisqartma ekanligini anglatish.

**** — Harflarni qalinlashtirish.

<bdi> — Yozuv yo'nalishini o'zgartirish.

<bdo> — Matn yo'nalishini ko'rsatish.

<cite> — Ko'chirma gap (boshqa manbadan olingan) ekanligini anglatish.

<code> — Dasturiy kod sifatida yozilganligini anglatish.

<label> — Forma elementiga nomlanish qo'yadi.

- ** — O'chirilgan yozuvni anglatish, ustidan chizilgan harflar bilan yoziladi.
- <dfn>** — Atama ekanligini ko'rsatish, egri tarzda yoziladi.
- ** — Urg'u berilgan matn bo'lagi, egri tarzda akslantiriladi.
- ** — Husnixatni o'zgartirish (ishlatish tavsiya etilmaydi).
- <i>** — Egri yozuv.
- ** — Rasm qo'yish.
- <input>** — Forma darchasini qo'yish.
- <ins>** — Qo'shilgan matn qismini anglatish, tagiga chizilgan holda yoziladi.
- <kbd>** — Kiritiladigan matnligini anglatish, keng harflar bilan yoziladi.
- <mark>** — Matnning muhim qismini anglatadi.
- <meter>** — Oraliqdagi qiymatni tasvirlash [7.4.6.]
- <q>** — Qisqa ko'chirmani anglatadi.
- <rp>** — <ruby> tegini qo'llamaydigan brauzerlarda nima chiqishini ko'rsatadi.
- <rt>** — <ruby> obyektining quyi yoki tepasida nima chiqishini ko'rsatadi.
- <s>** — Ustidan chizilgan yozuv.
- <samp>** — Dastur natijasi sifatida chiqqan yozuvni anglatadi.
- <small>** — Kichik o'lchamdagи yozuv.
- <select>** — Tayinlanuvchan darcha (ochiluvchi ro'yxat) qo'yish. [7.4.3.]
- ** — Oddiy qatorli teg yaratish.
- <sub>** — Quyi indeksdagi yozuv, masalan: H₂O

- <sup>** — Yuqoriroqda daraja sifatidagi yozuv, masalan: x^2
- ** — Muhim matnning qismiligini anglatish, qalin harflar.
- <textarea>** — Matn kiritish darchasi.
- <time>** — Sana va vaqtni anglatuvchi yozuv.
- <video>** — Lavha fayl qo'shish.
- <var>** — O'zgaruvchi ekanligini anglatish, egri yozuvda ifodalanadi.

Blokli teglar doim to'g'ri to'rtburchak shaklidagi maydonni egallaydi. Odatda, ular alohida qatordan o'rinn olib, butun kenglikni egallaydi. Qatorli teglar singari matn ichidagi so'zlar orasida joylashib ketavermaydi. *Qatorli teglar ichiga blokli teglar joylash mumkin emas!* Umuman olganda, qatorli va blokli teglarning boshqa ajralib turadigan xususiyatlari qolmagan. CSS yordamida qatorli tegni blokli tegga va aksincha aylantirish imkoniyati mavjud. Buning uchun **display** xususiyatidan foydalilanadi. Qiymati aniqlangan tur:

- **none** — yashirin, obyekt akslantirilmaydi.
- **inline** — qatorli teg obyekti sifatida akslanadi.
- **block** — blokli teg obyekti sifatida akslanadi.
- **inline-block** — o'zi qatorli teg ko'rinishida va ichki qismi blokli teg sifatida namoyon bo'ladi. Ya'ni oldingi element yakunlangan joyidan boshlangsasa ham, uning eni va bo'yini o'zgartirsa bo'ladi.
- **list-item** — ro'yxatdagi singari nishonchaga ega blokli teg bo'ladi. [3.4.2.]

- **inline-table** — blokli teg sifatida akslanadi, ammo atrofini boshqa elementlar, xususan, matn bilan o'rash mumkin bo'ladi.
- **flex** — blokli teg bo'ladi, faqat ichidagi qo'shni teglar bir nechta bo'lsa, yuzasi shu obyekt o'l-chamiga qarab bir xilda o'zgarib («moslashuvchan» [9.1.]) joylashadi.
- **inline-flex** — «flex» xususiyatli («moslashuvchan joylashuv» bo'yicha) qatorli teg.
- **run-in** — qatorli yoki blokli sifatida qat'iy aniqlanmagan teqlarning ichidagi tashkil etuvchilariga qarab o'zi aniqlash imkonini beradi. Buni qo'llash tavsiya etilmaydi, ba'zi brauzerlar to'g'ri akslantirmasligi mumkin.
- **table** — jadval obyekti singari namoyon bo'ladi.
- **table-caption** — jadval sarlavhasi xususiyatlari bilan akslantiriladi.
- **table-cell** — jadval katagi shaklida, «td» yoki «th» xususiyatlarini oladi.
- **table-column** — jadvaldagи «col» tegi xususiyatlari bilan akslanadi.
- **table-column-group** — «colgroup» obyekti xususiyatlari bilan namoyon bo'ladi.
- **table-footer-group** — «tfoot» singari oxirgi bir necha satrni akslantiradi.
- **table-header-group** — «thead» singari bir necha boshlang'ich qatorni saqlaydi.
- **table-row** — jadval qatori («tr») sifatida namoyon bo'ladi.
- **table-row-group** — «tbody» singari bir necha qatorni ko'rsatuvchi blokli teg.

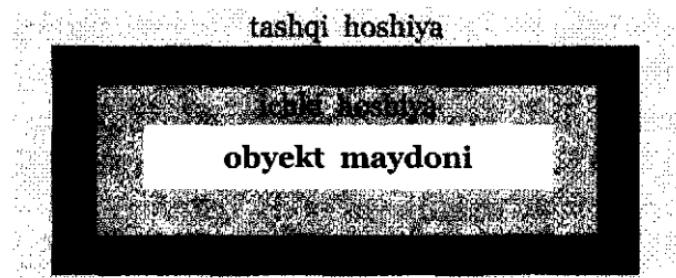
Quyidagi teglar blokli hisoblanadi:

- <address> — Muallif yoki tashkilot manzillarini ko'rsatadi.
- <article> — Matnning maqola, yangilik yoki maxsus xabarligini anglatadi.
- <aside> — Sahifaning yon qismidagi bo'lagini bildiradi.
- <audio> — Tovush faylini joylashtirish.
- <button> — Tugma yaratish.
- <blockquote> — Boshqa manbadagi ma'lumotni ajratib ko'rsatadi.
- <canvas> — Tasvir chizishni ta'minlash.
- <div> — Sahifaning alohida bo'limini ifodalaydi.
- <dl>, <dd>, <dt> — Izohli ro'yxat tuzadi [3.4.3.]
- <figure> — Sahifadagi elementlarni guruhlaydi.
- <footer> — Sahifaning quyi qismi.
- <form> — Forma yaratish.
- <h1> — Birinchi darajali sarlavha.
- <h2> — Ikkinchi darajali sarlavha.
- <h3> — Uchinchi darajali sarlavha.
- <h4> — To'rtinchi darajali sarlavha.
- <h5> — Beshinchi darajali sarlavha.
- <h6> — Oltinchi darajali sarlavha.
- <header> — Sahifa yoki bo'limning bosh qismini aniqlaydi.
- <hr> — Yotiq chiziq qo'yadi.
- <iframe> — Freym qo'yadi.

- <nav> — Ishoratlar guruhini aniqlaydi.
- , — Sanoqli ro'yxat tuzadi [3.4.1.]
- , — Belgilangan ro'yxat tuzadi [3.4.2.]
- <pre> — Tarkibini yozuvdag'i belgilanishlar saqlangan holda ko'rsatadi.
- <p> — Xatboshini aniqlaydi.
- <ruby> — Katta bo'limgan tavsifnomalar, kirish so'zlarini belgilaydi.
- <section> — Sahifa bo'limini anglatadi.
- <table> — Jadval yaratadi. [4.4.]

8.9. Obyekt shakli

Blokli teg to'g'ri to'rtburchak shaklida namoyon bo'ladi. Uning chegaraviy chizig'i, ya'ni perimerti hamda ichki va tashqi hoshiyasi mavjud. Tashqi hoshiya — chegaraviy chiziqdandan qo'shni obyektlargacha yoki o'z tashqi tegining maydoni qirrasigacha bo'lgan oraliq. Ichki hoshiya — bu chegaraviy chiziqdandan o'z elementlari qanchalik ichkarida joylashishini aniqlovchi masofa.



Quyidagi xususiyatlar bilan obyekt o'lchami aniqlanadi:

▪ **padding** — ichki hoshiya qalinligi, qiymati «o'lcham» yoki foiz [2.6.] hisobida «soat mili» qoidasi asosida beriladi. Ya'ni qiymat bittadan to'rttagacha son bilan aniqlanadi. Agar bitta son berilgan bo'lsa, to'rt tarafdag'i ichki hoshiya bir xilda o'sha songa teng bo'ladи. Ikkita son bo'lsa, yuqori va pastki hoshiyalar birinchи songa, yon hoshiyalar ikkinchi songa teng bo'ladи. Uchta son bilan berilsa, birinchisi yuqorini, ikkinchisi yon tomonlarini, uchinchisi pastki hoshiyani anglatadi. To'rtta son bo'lsa, ular mos ravishda yuqori, o'ng, past va chap hoshiyalarni bildiradi. Odatiy qiymati: 0. Ularning har birini alohida «padding-top», «padding-right» «padding-bottom», «padding-left» va kengaytirilgan xususiyatlar yordamida ham ifodalash mumkin.

▪ **margin** — tashqi hoshiya qalinligini o'lchamlarda ko'rsatuvchi umumlashtirilgan xususiyat. U «margin-top», «margin-right», «margin-bottom» va «margin-left» kengaytirilgan xususiyatlarini «soat mili» qonuniyati asosida aniqlaydi.

▪ **visibility** — Obyektning qay tarzda ko'rinishini taminlaydi, qiymati:

- **visible** — odatiy qiymat, o'z holicha ko'rinishini taminlaydi;
- **hidden** — yashirin, bunda barcha egallab turgan joyi, ya'ni o'lchamlari saqlanib qoladi. Faqat obyekt ko'rinishini taminlaydi, ya'ni shaffof rangda namoyish etiladi.
- **collapse** — bu faqat jadval kataklari uchun qo'llaniladi, agar yagona katakka tatbiq etilsa,

shunchaki ko'rinmas holda joy egallaydi, agar qator yoki ustunning barcha kataklariga qo'llanilsa, u jadvaldan o'chiriladi. Matn uchun qo'laniqganda «hidden» sifatida ishlaydi.

▪ **height** — obyekt maydonining balandligi, unga chegara va hoshiyalarning qalinligi kirmaydi. Qiymatini o'lcham yoki foiz hisobida berish mumkin. Odatiy qiymati: «auto», ya'ni ichki tashkil etuvchilarining umumiy balandligidan kelib chiqib belgilanadi.

▪ **width** — obyekt maydonining eni, unda ham chegara va hoshiyalar qalinliklari hisobga olinmaydi. 100% qiymat berilganda u tashqi obyekt kengligini to'la egalaydi. Aytaylik, unda «padding: 10px; margin: 8px;» deb ko'rsatilgan bo'lsa, bu holda element tashqi obyektniga sig'may surgich paydo bo'ladi yoki bir qismi qirqilib ko'rinxay qoladi. Sababi, ikkita yon tomonlarini hisobga olganda obyektning eni 100% dan «36px»ga kattalashib qoldi. Bunday vaziyatda surgich paydo qilmasdan obyektni to'la ko'rsatish uchun uning eniga quyidagicha qiymat beriladi [8.4.10.]:

```
width: calc(100% - 36px);
```

▪ **box-sizing** — obyektning eni va bo'yini o'lchamini hisoblash algoritmini aniqlaydi. Qiymati aniqlangan tur:

- **content-box**: odatiy qiymat, bunda «width» va «height»da ko'rsatilgan qiymatga ichki va tashqi hoshiyalar hamda chegara qalinligi kirmaydi.
- **border-box**: ko'rsatilgan qiymat o'z ichiga chegara qalinligi (border-width) va ichki hoshiyani (padding) oladi, tashqi hoshiya (margin) bunga kirmaydi.

- **box-shadow** – Obyektga soya qo'shib, uning umumiy o'lchamini oshiradi. Odatiy qiymati: «none», lekin qiymat berish sintaksisi biroz murakkabroq:

box-shadow: [inset] x y taralish cho'zilish rang;

Bunda «inset» soyaning obyekt ichida bo'lishini ta'minlaydi, agar ko'rsatilmasa, tashqariga tushadi, «x» va «y» – mos o'q bo'yicha surilish o'lchovi. Soya uzoqroqqa yorqin rangda tarqashi uchun «taralishi»ga kattaroq musbat qiymat berish zarur, odatda, uning qiymati 0, u bitta rangda aniq chegarada bo'lishini ta'minlaydi. Soya qalinligini «cho'zilish»ga berilgan qiymat aniqlaydi, odatiy qiymati 0 va bu holda soya obyekting o'lchamiga teng hisoblanadi. Soya «rang»i ko'rsatilmasa, u qora sifatida olinadi. Agar soya bir necha qatlamdan iborat bo'lishi kerak bo'lsa, qiymatlarni vergul bilan ajratib, shu tartibda bir nechtasini yozish mumkin.

Turli kengaytmali monitorlarda va brauzer hajmi o'zgartirilganda aniq o'lchami berilmagan teglar uchun akslantirilayotgan obyektlarning yuzasi o'zgaradi. Buning natijasida muhim ma'lumot sig'may qolishi yoki surilib ketganligi bois kerakli joyda ko'rinxmasligi mumkin. O'lcham o'zgarar ekan, bejirimligini yo'qotmaslik uchun obyektlarning eng katta va eng kichik qabul qilishi mumkin bo'lgan qiymatlarini qat'iy belgilab qo'yish darkor bo'ladi. Bunday vazifani mos ravishda «**max-height**», «**max-width**», «**min-height**» va «**min-width**» xususiyatlari bajaradi.

Chegaraviy chiziqning qalinligi «**border-width**» xususiyati yordamida beriladi. Qiymatiga «o'lcham» yoki

thin (2px), **medium** (4px) va **thick** (6px) so'zlaridan birini berish mumkin. Odatiy qiymati: «medium».

Hoshiyalardan farqli tarzda, chegara maxsus chiziq yordamida akslantiriladi. Agar u ko'rinuvchan bo'lsa, rangini ham o'zgartirish imkonи bo'lishi kerak. Buning uchun «**border-color**» xususiyatidan foydalaniladi. Qiymati: «rang». Agar ko'rinmas bo'lishi lozim bo'lsa, shaffof rang beriladi, u «**transparent**» so'zi bilan ifoda laniladi.

Chegaraning qanday tarzda chizilishi «**border-style**» xususiyati bilan aniqlanadi. Qiymati chiziq shaklini ko'rsatuvchi aniqlangan tur:

- **none** — odatiy qiymat, chiziq ko'rinxmaydi, qalinligini 0 qilib qo'yadi.
- **hidden** — ko'rinxmas;
- **dotted** — nuqtali;
- **dashed** — (uzuq) chiziqchali;
- **solid** — uzluksiz chiziqli;
- **double** — ikki qavat chiziqli;
- **groove** — botiq chiziqli;
- **ridge** — bo'rtiq chiziqli;
- **inset** — ichkariga soya tushgansimon;
- **outset** — tashqariga soya tushgansimon.

Chiziqnинг qalinligini, rangini va shaklini bildiruvchi bu uchala xususiyat ham «soat mili» qoidasi asosida to'rtala tarafga turlichа tatbiq etilishi mumkin. Qiymatlari to'rtala tomon uchun bir xil bo'lganda, ya'ni bittadan ko'rsatiladigan holda ularning yagona «**border**» xususiyatiga birlashtirish mumkin. Bunda qaysi kengay-

tirilgan xususiyat qiymati qayerda kelishining ahamiyati yo'q. Rang, o'lcham va aniqlangan turni anglatuvchi turlicha qiymatlardan brauzerning o'zi mosini kerakli xususiyatga biriktirib olaveradi.

`border: 2px inset green;`

Bunda to'rtala tomonga ham bir xil qoida tatbiq etiladi. Aniq bir tomonni boshqacha tarzda chizish lozim bo'lsa, uning uchun alohida ushbu xususiyatlardan biri ishlatiladi: «**border-top**», «**border-right**», «**border-bottom**» yoki «**border-left**». Bular ham uchta kengaytirilgan xususiyatni umumlashtiradi. Agar ayni bitta tomonga yagona kengaytirilgan xususiyat qiymati berilishi shart bo'lsa, tanishib chiqilgan qonuniyatlar asosida quyidagi-cha yozish mumkin:

<code>border-top-color</code>	<code>border-top-style</code>	<code>border-top-width</code>
<code>border-right-color</code>	<code>border-right-style</code>	<code>border-right-width</code>
<code>border-bottom-color</code>	<code>border-bottom-style</code>	<code>border-bottom-width</code>
<code>border-left-color</code>	<code>border-left-style</code>	<code>border-left-width</code>

Misol tariqasida tagiga uzuq chiziqli chegara qo'yildigan obyektni ko'ramiz:

`border-bottom-style: dashed;`

Jadvaldagи kataklarni yoki yonma-yon turuvchi blokli teg obyektlari chegaralovchi chiziqlar alohida-alohida bo'lib, ikki qatlamlı chiziq sifatida ifodalanadi. Bu esa xunuk ko'rinish kasb etadi. Jadvalda chiziqlarni bittalik qilib chizish uchun «cellspacing» alomatiga 0 qiymat

beriladi. Buning CSSdagi muqobili «**border-collapse**» xususiyatidir. Qiymati aniqlangan tur:

- **collapse** — chegarani anglatuvchi chiziqlar yonma-yon kataklar uchun birlashtirilib, yagonaday ifodalananadi. Bunda jadvaldagi «cellspacing» alomatining qiymati inobatga olinmaydi.
- **separate** — chegaraviy chiziqlar har bir obyekt uchun alohida chiziladi. Yonma-yon obyektlar uchun ikki qatlamlili chiziq namoyon bo'ladi.

Ushbu xususiyat jadvallar yoki «display» xususiyatida jadvalga oid qiymat berilgan obyektlar uchun ishlaydi. Jadvalning yana bir alomati — «cellpadding» [4.1.] vazifasini CSSdagi «**border-spacing**» xususiyati bajaradi.

To'g'ri to'rtburchak shaklidagi chegara qirralarini to'g'ri burchak shaklida bo'lishi zamonaviy shamoyil tablarga u qadar ham mos emas. Bejirim ko'rinishi uchun qirralarni yoysimon qilib qo'yish odat tusiga kirib ulgurdi. Avvallari buning uchun to'rtta chetga maxsus rasm qo'yib, uni chegaraviy chiziqli birlashtirib yuborish mashaqqati mavjud edi. Lekin chiroyli ko'rinishni ta'minlash uchun dasturlovchi dizaynerlar anchagina ortiqcha mehnatni amalga oshirishga ham majbur bo'lishgan. Hozirda qirrani o'tmaslashtiruvchi xususiyat «**border-radius**» mavjud bo'lib, uni barcha zamonaviy brauzerlar qo'llaydi. Unga soat mili qoidasi bo'yicha berilgan o'lchamdagisi radiusli aylana yoyi chegaraviy chiziqlarga birlashtirilib, qirra o'rniga yoy

chiziladi. Agar yoy aylana emas, ellipsning qismi sifatida ko‘rinishi lozim bo‘lsa, qiymatda «/» (taqsim)dan so‘ng ikkinchi radius beriladi.

Aynan faqat bitta burchakni yoysimon qilish uchun «**border-bottom-right-radius**», «**border-bottom-left-radius**», «**border-top-left-radius**» va «**border-top-right-radius**» maxsus xususiyatlaridan birini qo’llash kifoya. Ta’kidlab o’tish lozimki, ushbu xususiyat «border-collapse: collapse» qiymatini olgan jadvallar uchun ishlamaydi.

Chegarani oddiy chiziqlar bilan chizish har doim ham talabni to‘la qondira olmaydi. Ba’zan maxsus rasm asosida shakl berishga to‘g’ri keladi. Lekin rasmning o’lchamini oldindan kattaligi aniq bo’lmagan obyektga moslash biroz murakkablik tug‘diradi. Bunday muammoni hal etish uchun «**border-image**» xususiyati qo’llaniladi. Uning tatbiq etish algoritmi quyidagicha: avval to‘g’ri to‘rtburchak shaklida bezakli chegaraga ega rasm (ramka) tanlanadi. Uning qaysi tomonidan qancha nuqta birligida kesib olinishi lozimligi aniqlanadi. Kesilgan tasmalarni obyektning chetiga ma’lum masshtab asosida o’zgartirib, uchlari taqab qo‘yiladi, ortib yoki yetmay qolgan tomonlar asosiga tasma bo‘laklari cho‘zib, takrorlanib yoki kesilib to‘ldiriladi. Shu tariqa rasmdagi hoshiyaviy bezakning obyektga mos masshtablangan ko‘rinishi paydo bo‘ladi. Ushbu xususiyatning umumiy sintaksisi quyidagi-cha:

border-image: rasm eni [/qalinligi] [joylashishi];

Bunda url funksiyasi yordamida «rasm»ning manzili ko‘satiladi, rasmning qaysi tomonlaridan qancha nuq-

tadan qirqib olish lozimligini «rasm eni»da, uni qanchaga ko'paytirib qo'yish lozimligi esa «qalinligi»da ko'rsatiladi. Bu ikki qiymat «soat mili» qoidasi asosida bittadan to'rttagacha son qabul qilishi mumkin. Faqat ularda butun son ko'rsatiladi, «o'lcham» emas (10px sifatida qiymat berish xato). Mos tomoni uchun rasmidan kesib olingen tasmani «joylashishi»da qanday qo'yish lozimligini ko'rsatiladi. Uning odatiy qiymati «stretch» bo'lib, unda tasma yuzasini cho'zib yoki siqib ko'rsatilgan o'lchamga moslanadi; «repeat» qiymati berilsa, qirg'oqda (tasmani o'zgartirmasdan) rasm o'z o'lchamida qaytadan qo'yilib takroriy davom ettiriladi; «round»ni qabul qilganda tasma shunday masshtabda o'zgartirib joylanadiki, natijada takrorlanadigan rasm butunligini saqlaydi.

border-image: url(hoshiya.png) 30 20 / 1 1.5 round round;

Ta'kidlab o'tish lozimki, «border-collapse: collapse» qiymatini olgan obyektlar uchun ushbu xususiyat ishlamaydi.

Shaklni ifodalovchi yuqoridagi xususiyatlarni o'lchamni aniqlovchilarini istisno qilgan holda qatorli teglar uchun ham tatbiq etish mumkin.

9-BO'LIM

OBYEKTLARNI JOYLASH VA JILOLASH

9.1. Obyektni joylashtirish

Obyektlarni joylashtirish va jilolash uchun faqat CSS imkoniyatlardan foydalanish tavsiya etiladi. Yuqorida ta'kidlaganimizdek, har xil ekranlarda veb-sahifa o'ziga xos tarzda namoyon bo'lishi kerak.

Obyekt HTMLda qanday e'lон qilinishidan qat'i nazar, CSS yordamida nafaqat uning ko'rinishini, balki joylashadigan o'rnini ham o'zgartirish mumkin. Joylashishni aniqlovchi xususiyatlar, odatda, blokli teglarga tatbiq etiladi, lekin juda zarur bo'lsa, qatorli tegni «display» yordamida turini o'zgartirib, unga ham qo'llash mumkin. Quyida joylashishni belgilovchi muhim xususiyatlar bilan tanishamiz.

Elementning **tepadan**, **o'ngdan**, **pastdan**, **chapdan** o'z tashqi obyektiga nisbatan qancha masofada joylashishi mos ravishda **«top»**, **«right»**, **«bottom»**, **«left»** xususiyatlari bilan beriladi. Ularni kelgusida **«atrof xususiyatlari»** deb yuritamiz. Qiymati o'lcham yoki foizda beriladi, odatiy qiymati: **«auto»** va u obyekt joyini o'zgartirmaydi.

Brauzer oynasi yoki boshqa obyektlarga nisbatan qanday joylashishi quyidagi xususiyat bilan aniqlanadi:

position: absolute|fixed|relative|static|sticky;

Bunda:

- **absolute** — obyektning hajmi boshqa elementlarning joylashishiga ta'sir qilmaydi, go'yo u o'zini boshqa parallel tekislikda (qatlamda) turganday tutadi. Atrof xususiyatlari (top, right, bottom, left) brauzer oynasi bo'yicha, aniqrog'i, «body» akslangan maydonga nisbatan olinadi.
- **relative** — nisbiy joylashuv, bunda atrof xususiyatlari o'zi turgan o'ringa nisbatan qancha masofaga ko'chishini ko'rsatadi. Shu joyda muhim eslatma: agar «relative» qiymat berilgan obyekt ichiga boshqa «absolute» qiymatli element joylansa, uning atrof xususiyatlari «body»ga nisbatan emas, tashqi obyekt joylashuviga nisbatan olinadi.
- **fixed** — joylashishi bo'yicha «absolute» qiymatiga o'xshaydi, ammo surgich yordamida sahifa siljitelganda ham u o'z o'rnnini o'zgartirmaydi. Bu qiymatdan ko'pincha reklama oynalarini qo'yishda foydalilaniladi.
- **static** — obyekt odatdagidek akslanadi, bunda atrof xususiyatlarining qiymati ta'sir qilmaydi.
- **sticky** — o'zgaruvchan joylashuv. Ma'lum chegara doirasida o'zini ko'chuvchan tutadi, chegara tashqarisida esa o'zgarmas holatda qotadi. Bu, odatda, suriladigan matnga ega bo'lgan sahifada sarlavha uchun ishlataladi. Surgich

bilan siljitelganda matn ko'rinadigan vaziyatda ko'chib boradi, surilayotgan obyekt ekranda ko'rinmay qolganda u ham joyini o'zgartirmasdan qotadi.

Joylashuvi va atrof xususiyatlarining ba'zi qiymatlarida obyektning o'lchami mos ravishda o'zgaradi, bunda «width» va «height» qiymatlari inobatga olinmasligi mumkin. Masalan:

```
position: relative;
```

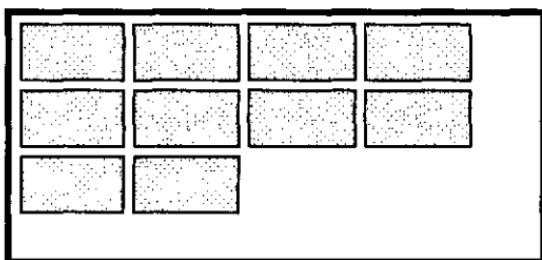
```
top: 3px;
```

```
right: 2px;
```

```
bottom: 3px;
```

```
left: 2px;
```

Shu o'rinda «**moslashuvchan joylashuv**» tushunchasi ga to'xtalib o'tsak, bir nechta obyektlarni boshqa bittasining ichida qanday ko'rinishda namoyon bo'lishini aniqlashda bu usul keng qo'llaniladi. Aslida, u katta quti ichida kichik qutilarni tartib bilan taxlashga o'xshaydi. Bitta qator to'lgandan keyin (ortib qolgan joyga bitta quti sig'magan taqdirda) ikkinchisiga o'tiladi. Faqat oxirgi qatordagi qutilar soni, umumiy miqdorga qarab, oldingilaridan kam bo'lishi mumkin. Bunda qutichalar yotiq va tikkasiga tekis kataklar sifatida joylashadi. Misol:



Ushbu rasm «moslashuvchan joylashuv»ni to‘g‘ri tasavvur qilish uchun imkon beradi. Tashqi qutini «moslashuvchan maydon» deb yuritamiz. Bunda har bir ichki obyektning atrof xususiyatlari bir xil qiymat qabul qiladi. Bunday usulda joylashishni boshqarish uchun quyidagi xususiyatlar ishlatiladi:

▪ **flex-direction** — elementlarning joylashish yo‘nalishini aniqlaydi. Qiymati:

- **row**: joylashish tartibi matn yozilish yo‘nalishida bo‘ladi (odatiy qiymat);
- **row-reverse**: joylashish tartibi matn yozilish yo‘nalishiga teskari bo‘ladi;
- **column**: ustun shaklida tepadan pastga yo‘naladi;
- **column-reverse**: ustun shaklida pastdan te-paga qarab joylashadi;

▪ **flex-wrap** — elementlarning qatorda joylashishini aniqlaydi. Qiymati:

- **nowrap**: ichki elementlar faqat bitta qatorda joylashadi (odatiy qiymat);
- **wrap**: ichki elementlar «flex-direction»da ko‘rsatilgan yo‘nalishda bir necha qatorda joylashishi ta‘minlaydi.
- **wrap-reverse**: «flex-direction»da ko‘rsatilgan-dagiga teskari yo‘nalishda ichki elementlarning bir necha qatorda joylashishini ta‘minlaydi.

▪ **flex-flow** — umumlashgan xususiyat, «flex-wrap» va «flex-direction» qiymatlarini birdaniga berish imkoniyatini yaratadi.

- **flex-grow** — ichki elementlar o'lchamining proporsionalligini ta'minlaydi. Agar barcha elementlar 1 qiymat qabul qilgan bo'lsa, ularning barchasi teng hajmda bo'ladi, qaysidir biriga 2 qiymat berilgan bo'lsa, u 1 qiymatlardan ikki barobar katta akslantiriladi. Qiymati: son.

- **flex-shrink** — siqish koeffitsiyenti, barcha ichki elementlarni bitta qatorda joylashtirish uchun obyektning boshqalariga nisbatan necha barobar qisqartirish mumkinligini aniqlaydi. Qiymati: butun son.

- **flex-basis** — ichki elementlarning boshlang'ich kengligini aniqlaydi. Odatiy qiymati: «auto», aslida, o'lcham yoki foiz hisobida eni ko'rsatiladi.

- **flex** — «flex-grow», «flex-shrink», «flex-basis»larni umumlashtiruvchi xususiyat.

- **order** — «moslashuvchan» elementlarning yozilish tartibida emas, ko'rsatilgan raqamlar bo'yicha ketma-ketlikda namoyon bo'lishini ta'minlaydi. Misol:

```
.quти:nth-of-type(1) { order: 3; }
```

9.2. Orqa tasvir

Yuqorida obyektning chegarasini tasvirlash imkoniyatlari bilan tanishgan edik. [8.9.] Shunga o'xshash xususiyatlarni orqa tasvir (orttasvir) uchun ham qo'llash mumkin. Rangi quyidagi xususiyat bilan qo'yiladi:

background-color: *rang* | transparent;

Odatiy qiymati: «transparent», shaffof rangni anglatadi. Bunda ko'rsatilgan rang obyektni to'liq qoplaydi va undagi elementlar orqasida tasvirlanadi. Agar yagona rangning o'rниga rasm yoki gradiyent qo'yilishi lozim bo'lsa, «background-image» [8.4.9.] xususiyatidan foydalilaniladi. Unda bir nechta rasmi vergul bilan ajratib ko'rsatish mumkin. Ko'rsatilgan rasm yuklanguncha «background-color»da berilgan rang akslanib turadi. Ortrasm obyekt tashkil etuvchilari bilan birga surilishini ta'minlash uchun quyidagi xususiyatdan foydalilaniladi:

background-attachment: fixed | scroll | local;

- **fixed** – ortrasm harakatlanmaydi;
- **scroll** – tashkil etuvchilar bilan birga suriladi (odatiy qiymat);
- **local** – agar obyekt surgichga ega bo'lsa, suriladi. Faqat elementning tashqarisiga chiqib ketgan ortrasm qismi o'z joyida qoladi.

Chegara uzlusiz chiziq bilan berilmaydigan hollarda ortrasm chegaraning aynan qaysi qismidan boshlab ko'rinishi kerakligini belgilash muhim ahamiyatga ega. Chegarani ifodalovchi tasvirning shaffof qismlari mavjud bo'lsa, ortrasm o'sha joylarda ham o'z ifodasini topishi maqsadga muvofiq bo'ladi. Bunday imkoniyatni aniq belgilash uchun quyidagi xususiyat qo'llaniladi:

background-clip: text|padding-box|border-box|content-box;

- **text** – ortrasm faqat matn ortida akslanadi.

- **border-box** — odatiy qiymat, ortrasm chegara tagida ham akslanadi;
- **padding-box** — ortrasm chegara ichida (ichki hoshiyadan boshlab) akslanadi;
- **content-box** — ortrasm faqat obyekt maydonining o'zida akslanadi.

Umuman olganda, ushbu xususiyat chegara va ichki hoshiya qalinligiga qarab ortrasmni qayerdan boshlab akslantirish lozimligini aniqlaydi. Aslida, aynan shuning uchun «**background-origin**» xususiyati ham kiritilgan, faqat bu «background-attachment: fixed;» qiymat berilganda inobatga olinmaydi. Qolgan hollarda qabul qiladigan qiymatlari va vazifalari aynan «background-clip» bilan bir xil. Faqat «background-clip: text;» qiymati bundan mustasno, ushbu qiymat o'ziga xos bo'lib, eski brauzerlar uchun «-webkit-background-clip: text;» singari yozilishi talab qilinadi.

Rasmning o'lchami har doim ham obyektning akslanyotgan hajmiga mos kelavermaydi. Bunday vaziyatlar da «**background-size**» xususiyati yordamida masshtablashga to'g'ri keladi. Qiymat sifatida ikkita son: eni va bo'yini, o'lcham yoki foiz hisobida berish mumkin. Agar bitta qiymat berilsa, u eni hisoblanib, bo'yi unga proporsional ravishda hisoblab topiladi. Aniq o'lchamlar berilganda ortrasm ko'rsatilganday to'g'ri to'rtburchak shaklida namoyon bo'ladi, foiz ko'rsatilganda obyektning hajmiga nisbatan hisoblanadi. Odatiy qiymati «**auto**» berilsa, hajmi o'zgarmasdan akslantiriladi. Agar bitta qiymat berilib, ikkinchisi sifatida «**auto**» yozilsa, ikkinchisi proporsionallik saqlangan holda birinchisiga

nisbatan hisoblanib topiladi. Qiymat sifatida «cover» yozilsa, ortrasmni to'laligicha namoyon etadi, bunda masshtab shunday o'zgaradiki, eni yoki bo'yи obyekt o'lchamiga to'g'ri keladi. Qiymat o'rniga «contain» berilganda ortrasmni masshtablab, obyektning ichida bo'sh joy qolmaydigan tarzda joylashtiriladi. Bunda ortrasmning eni yoki bo'yidan bir bo'lagi ko'rinnay qolishi mumkin.

Ko'rsatilayotgan rasm o'lchami o'zgartirilar ekan, katta hajmdagi faylni qayta ishlashda biroz vaqt talab qilinishi mumkin. Bunday vaziyatlarda masshtablash uslubini o'zgartirish orqali kerakli natijaga erishish mumkin, «image-rendering» xususiyati qanday usulda masshtablash kerakligini aniqlaydi. Qiymati aniqlangan tur:

- **auto** – brauzerning o'zidagi algoritm bo'yicha o'zgartiriladi, odatda, u «bikubik» uslub hisoblanadi;
- **crisp-edges** – bunda tasvirdagi ranglarning chegaralovchi va ularga yaqin nuqtalarni akslantirish bilan tez namoyish bo'lishi ta'minlanadi;
- **pixelated** – kattalashtirilayotganda tiniqlik saqlanishi ta'minlanadi, kichiklashayotganda «auto» singari ishlaydi. Bunda rasm asta paydo bo'ladi.

Har doim ham ortrasm obyektni to'laligicha qoplashi talab etilmaydi. Masalan, tugmalarga nishoncha qo'yish lozim bo'lsa, undagi nom oldida kichik rasm akslantirilishi kerak. Rasmning o'lchamiga qarab uning joylashadigan holatini ham o'zgartirish talab etiladi. Buning uchun quyidagi xususiyatidan foydalilaniladi:

background-position: joyi;

Bunda «joyi» obyektning chap yuqori qirrasiga nisbatan tik va yotiqlar qanday masofada joylashishi bo'sh joy bilan ajratib ko'rsatiladi. [8.4.9.] Bundan tashqari, faqat yotiqlar joylashish «background-position-x» yoki faqat tik joylashish «background-position-y» xususiyatlari bilan ko'rsatilishi mumkin.

Aksariyat hollarda ortrasmni takrorlashga to'g'ri keladi. Katta-katta obyektlarning ortini tasvirga to'ldirish uchun hajmi kichik rasmdan foydalanib takroriy namoyish etiladi. Takrorlanishni «background-repeat» xususiyati yordamida boshqariladi. U quyidagi qiymatlardan birini qabul qiladi:

- **repeat** — odatiy qiymat; ortrasm ham yotiqlar, ham tik takrorlanadi.
- **no-repeat** — takrorlanmaydi, ortrasm ko'rsatilgan nuqtada chiqadi.
- **repeat-x** — faqat yotiqlar takrorlanadi.
- **repeat-y** — faqat tikkasiga takrorlanadi.
- **space** — butun sondagi rasmlar ko'rindan digan holda takrorlanadi, bunda oraliq masofalar o'zgartirilishi mumkin.
- **round** — butun sondagi rasmlar ko'rindan digan holda takrorlanadi, bunda ortrasm masshtabi o'zgarishi mumkin.

Qiymatlarni yotiqlar va tik ravishda alohida-alohida «repeat-x» va «repeat-y» bilan berish o'rniiga bo'sh joy bilan ajratib ketma-ket berish mumkin. Masalan:

background-repeat: space round;

Beshtagacha orttasvirni aniqlovchi kengaytirilgan xususiyatlarni yagona tarzda «background»ga umumlash-tirish mumkin. Bunda ketma-ketlik muhim ahamiyatga ega emas, faqatgina, «background-position» qiymatidan «background-size»ni ajratish uchun taqsim («/») bilan bo'lib yoziladi. Odatiy qiymati:

```
background: transparent none repeat scroll 0% 0%;
```

Aslida, u «background-color», «background-image», «background-repeat», «background-attachment» hamda «background-position»larning odatiy qiymatlarini umum-lashtirgan.

9.3. Nisbiy joylashtirish

Obyektning tashqi va qo'shni elementlarga nisbatan qanday joylashishi quyidagi xususiyatlar yordamida ko'r-satiladi:

▪ **clear** — obyektni qaysi tomondan qo'shni elementlar o'rab olishini taqiqlashni ko'rsatadi. U «float»ning qiymatini bekor qiladi, aniqlangan tur qabul qiladi:

- **left** – chapdagи elementlarni pastga tushiradi.
- **right** – o'ngda elementlar chiqishini taqiqlaydi.
- **both** – ikki tomondan ham o'rashni man etib, qatorda yolg'iz o'zi akslanadi.
- **none** – odatiy qiymat, «float» yoki boshqa sozlashlarga qarab akslanadi.

▪ **float** — obyektning qaysi tomonga tekislanishi lozimligini ko'rsatadi, bunda qolgan tomonlardan boshqa elementlar o'rab oladi. Qiymati — aniqlangan tur:

- **left** — chapga tekislanadi, o'ngdan boshqalar o'rab olishi mumkin.
- **right** — o'ngga tekislanadi, boshqa elementlar chapdan o'rab olishi mumkin.
- **none** — oddiy holat (odatiy qiymat).

▪ **outline-color** — tashqi chegara rangi. Odatiy qiymati: «invert», ortrangiga qarab o'zi aniqlashga imkon beradi. Barcha brauzer buni birdek qo'llamagani uchun aniq rang berish tavsiya etiladi.

▪ **outline-offset** — elementning tashqi (outline) chegarasi va o'z chegarasi (border) orasidagi masofani o'rnatadi. Odatiy qiymati: 0. Manfiy o'lchamdagiga qiyomat berish orqali tashqi chegaraning obyekt ichida akslani shini ta'minlash mumkin.

▪ **outline-style** — tashqi chegara shakli. Qiymati chiziq shaklinikiday. [8.9.]

▪ **outline-width** — tashqi chegara qalinligi. Qiymati «border-width»niki singari.

▪ **outline** — tashqi chegara shakli. Bir paytning o'zida «outline-color», «outline-style», «outline-width»larning qiymatlarini ko'rsatishga mo'ljallangan. Tashqi chegara bilan oddiy chegara (border)ning farqi shundaki, tashqi chegara obyektning o'lchamiga va joylashishiga ta'sir o'tkazmaydi hamda to'rtala tomon uchun ham bir xilda bo'ladi.

▪ **z-index** — elementning boshqalarga nisbatan ustida yoki ostida bo'lishini anglatadi. Umuman olganda, veb-sahifadagi barcha elementlar bitta tekislikda joylashadi, lekin ba'zan bitta obyektni boshqalarining ustida ularning joylashuvini o'zgartirmagan holda namoyon etishga zaruriyat tug'iladi. Bunday hollarda kerakli obyektlarni boshqa qatlamda akslantirishga to'g'ri kela-di. Ushbu xususiyat butun sondagi qiymat qabul qilib, elementning qaysi qatlamdaligini ko'rsatadi. Qancha qiymat katta bo'lsa, demak, eng ustida joylashgan degan tasavvurni beradi. Obyektning boshqa qatlamda ekanligi «position» xususiyati orqali ko'rsatiladi.

9.4. Ichki elementlarini akslantirish

Obyektning (blokli tegning) ichidagi tashkil etuvchi elementlarini joylashtirish uchun quyidagi xususiyatlaridan foydalaniladi:

▪ **align-content** — oraliq masofalarni o'zgartirish orqali «moslashuvchan joylashuv»ni ta'minlaydi. [9.1.] Qiymati — aniqlangan tur:

- **flex-start** — «taxlash»ni bir boshdan boshlaydi;
- **center** — elementlarni moslashuvchan maydon markaziga joylaydi;
- **flex-end** — ortiqcha bo'shliq moslashuvchan maydonning yuqori qismida bo'ladi;

- **space-between** – ortib qolgan bo'shliq, qatorlar o'rtasida teng taqsimlanadi;
 - **space-around** – barcha ichki elementlarning yuqori va pastki tashqi hoshiyalari teng bo'ladi;
 - **stretch** – ichki elementlar ortiqcha bo'shliq to'lguncha birday kattalashadi;
- **justify-content** – elementlarning o'lchами va hoshiyalarini inobatga olgan holda «moslashuvchan joylashuv»ni ta'minlaydi. Qiymati «align-content»nikiday.
- **align-items** – ichki elementlarning ustun bo'yicha joylashuvini aniqlaydi. Qiymati – aniqlangan tur:
- **flex-start** – tepalari bir tekisda joylashadi;
 - **center** – markazlari tashqi maydon o'rtasidagi yotiq chiziqdagi joylashadi;
 - **flex-end** – pastki chegaralari tashqi maydonning quyi qismida joylashadi;
 - **stretch** – odatiy qiymat, elementlar cho'zilib, to'la balandlikni egallaydi;
 - **baseline** – «asos chizig'i» bo'yicha tekislanadi.
[8.8.]
- **align-self** – faqat joriy qatordagi elementlarni «align-items»da ko'rsatilgandagidek joylashtiradi. Odatiy qiymati: «auto», ya'ni tashqi elementi mavjud bo'lmasa, hammasini teng bo'yli qiladi, aks holda, «align-items» qiymatini oladi.
- **clip** – elementning qanday shakldagi qismi ko'rinishi lozimligini bildiruvchi funksiya ko'rsatiladi. Hozircha faqat to'g'ri to'rtburchak shakli qo'llaniladi. Odatiy qiymati «auto» bo'lib, element maydoni to'liq ko'rindi. Misol:

clip: rect(Y1, X2, Y2, X1);

▪ **column-count** — ichki elementlarning nechta ustun sifatida joylashishini ko'rsatadi. Bu usul ro'znomadagidek matnlarni bir necha ustun ko'rinishida yozish lozim bo'lganda keng qo'llaniladi. Odatiy qiymati: «auto». O'zi aniqlaydi, lekin aniq son tariqasida ustunlar nechtligini ko'rsatish tavsiya etiladi.

▪ **column-fill** — obyektning tashkil etuvchilari ustunlarga qanday taqsimlanishini ko'rsatadi. (Hozircha brauzerlar buni qo'llamaydi.) Qiymati aniqlangan tur:

- **auto** — ustunlar ketma-ket to'ldirib boriladi.
- **balance** — barcha ustun teng to'ldiriladi, faqat chop etilayotganda ushbu qoida oxirgi sahifaga tatbiq etiladi.
- **balance-all** — xuddi «balance»dagidek chop etilayotganda ham barcha sahifaga tatbiq etiladi.

▪ **column-gap** — ustunlar orasidagi masofa. Odatiy qiymati: «normal», bunda brauzerning o'zi aniqlaydi. Aslida, o'lcham sifatida masofa berish mumkin.

▪ **column-rule** — ushbu umumlashtirilgan xususiyat ustunlar orasiga chiziq chizishga mo'ljallangan. Odatda, chiziq yashirin, matn rangi bilan bir xil tusda bo'ladi. Bunda chiziqning rangini «**column-rule-color**», qalinligini «**column-rule-width**» va shaklini «**column-rule-style**» xususiyatlari bilan [8.9.] alohida-alohida berilishi ham mumkin.

▪ **column-span** — ichki element qancha ustunlarni egallashi lozimligini ko'rsatadi. Odatda, bu xususiyat sarlavha yoki rasm qo'yishda keng qo'llaniladi. Afsuski,

hozircha bu xususiyatni «Firefox» qo'llamaydi, ustunlarga oid yuqoridagi xususiyatlarning ba'zilariga «-moz» old qo'shimchasi bilan yozilgandagina uni to'g'ri akslanadiradi. Umid qilamizki, kelgusi talqinlarida buni boshqa brauzerlar singari inobatga oladi. Bu xususiyatning qiymati aniqlangan tur:

- **none** – odatiy qiymat, bitta ustunda joylashishini ta'minlaydi.
- **all** – element barcha ustunlarni egallaydi.

▪ **columns** – ustunlar sonini va enini birdaniga berish mumkin bo'lgan umumlashtirilgan xususiyat. Ta'kidlab o'tish lozimki, ustunlarga ajratish jadvalning ichida ishlamaydi, u boshqa blokli teglar uchun mo'ljalangan. Odatiy qiymati: «auto», aslida, «column-width» va «column-count» qiymatlari bo'sh joy bilan ajratilgan holda beriladi.

▪ **direction** – matn yozilish yo'nalishini ko'rsatadi yoki elementlarning joylashish tartibini aniqlaydi. Ushbu xususiyat har bir harfning o'rnini teskari tartibda qaytadan joylashtirib chiqmaydi, faqat ketma-ketlik qanday yo'nalishda bo'lishini bildiradi. Uni jadvalning kataklari yoki elementlarning akslanish tartibini o'zgartirish uchun ham qo'llash mumkin. Qiymati aniqlangan tur:

- **ltr** – chapdan o'ngga, odatiy qiymat.
- **rtl** – o'ngdan chapga.

▪ **object-fit** – qat'iy eni va bo'yisi ko'rsatilgan blokli teg ichidagi rasm yoki lavha o'lchamini o'ziga moslaydi. Qiymati aniqlangan tur:

- **contain** – proporsionallik saqlangan holda eni yoki bo'yisi tashqi tegnikiga teng bo'ladigan qilib

masshtablanadi va tashqi maydon o'rtasiga joylashadi.

- **cover** – proporsionallik saqlangan holda eni ham, bo'yи ham tashqi maydonni to'ldiradigan tarzda masshtablanadi va sig'may qolgan qismi kesib tashlanadi.
- **fill** – odatiy qiymat, obyektning eni ham, bo'yи ham tashqi tegniki singari o'zgartirilib, to'la maydonni qoplaydigan tarzda akslantiriladi.
- **none** – o'lcham o'zgartirilmaydi.

▪ **overflow** – o'lchami ko'rsatilgan blokli elementning ichidagilari sig'magan holda qanday yo'l tutishni ko'rsatadi, qiymati aniqlangan tur:

- **visible** – odatiy qiymat, tashqarisiga chiqib ketgan qismlari ham ko'rindi.
- **hidden** – sig'magan qismlari ko'rindmaydi.
- **scroll** – doim surgich ko'rsatiladi.
- **auto** – ichidagilar sig'magan hollarda surgich ko'rindi.

▪ **overflow-x** – faqat yotiқ yo'nalishda «overflow»ni tatbiq etadi.

▪ **overflow-y** – faqat tik yo'nalishda «overflow»ni tatbiq etadi.

▪ **vertical-align** – element balandligi yoki undagi matn qatorlariga nisbatan obyektni tekislab joylashiradi. Qiymati aniqlangan tur:

- **baseline** – asos chizig'iga tekislaydi. Asos chizig'i sifatida matnning birinchi qatori quyi qismi qaraladi, agar unday tasavvuriy chiziq

mavjud bo'lmasa, pastki chegara qabul qilinadi (odatiy qiymat).

- **bottom** – eng quyi qatorning pastki chegarasiga tekislaydi;
- **middle** – asos chizig'iga elementning o'rtasi va unga «x» harfining yarmichalik masofa qo'shib, shu nuqtaga nisbatan (tik yarmidan) o'tkazilgan chiziqqa tekislaydi;
- **sub** – asos chizig'inining pastida indeks sifatida o'lchami o'zgarmasdan yoziladi;
- **super** – asos chizig'i tepasida daraja sifatida o'lchami saqlanib yoziladi;
- **text-bottom** – so'nggi qatorning quyi chegarasiga tekislanadi;
- **text-top** – ilk qatorning yuqori chegarasiga tekislanadi;
- **top** – elementning yuqori chegarasiga tekislanadi;

9.5. Xos xususiyatlar

CSSda shunday xususiyatlar mavjudki, ular faqatgina ma'lum teglargagina taalluqli bo'ladi. Faqat o'ziga xos tegning xususiyatini o'zgartirish uchungina qo'llaniladi va boshqa elementlar uchun ishlamaydi. Masalan:

- **caption-side** – jadval yoki jadval tarzida akslanrilishi ko'rsatilgan blokli tegning sarlavhasi qayerda

ko'inishini aniqlaydi. U «table» ichidagi «caption» tegi uchungina qo'llaniladi. Qiymati aniqlangan tur:

- **top** – odatiy qiymat, sarlavha jadvalning tepe qismida ko'rindi.
- **bottom** – sarlavha jadvalning ostida ko'rindi.
- **empty-cells** – jadval katagi yoki jadval tarzida aks-lantirilishi ko'rsatilgan blokli teg chegarasi va ortrangini ko'rsatishni ta'minlaydi. Agar katak ichida hech qanday element bo'lmasa yoki akslanmaydigan bo'sh joylar mavjud bo'lsa, u bo'sh katak hisoblanadi. Ushbu xususiyatning qiymati «**border-collapse: collapse;**» berilganda e'tiborsiz qoldiriladi. [8.9.] Qiymati:

- **show** – odatiy qiymati, ortrang va chegaralar ko'rindi. Bu HTML5 talabi bo'yicha, oldin bo'sh katak ham ko'rinishi uchun akslanuvchi bo'shliq – « » qo'yib yozishga to'g'ri kelardi.
- **hide** – katak ko'rinxaydi, «**visibility: hidden;**» qiymati berilgani kabi. [8.9.]
- **table-layout** – jadval katagi yoki jadval tarzda aks-lantirilishi ko'rsatilgan blokli teg obyektining o'lchami qanday aniqlanishini belgilaydi. Qiymati aniqlangan tur:
 - **auto** – brauzer jadvalni to'laligicha yuklaydi, tarkibini tahlil qilib kataklar o'lchamlari qanday bo'lishini aniqlaydi va shundan so'nggina aks-lantiradi.
 - **fixed** – kataklar eni birinchi qatordagilar uchun ko'rsatilgan o'lchamda yoki <col> tegida ko'rsatilmagan bo'lsa, kataklarning eni teng bo'ladi.

Yozuvlar katak ichiga to‘g‘ri joylashishi uchun bunday jadvalning o‘lchamini oldindan berishga to‘g‘ri keladi.

▪ **quotes** — qo‘shtirnoq qo‘yishni ta’minlaydi. U faqat <q> tegi ichida yozilgan matnni qo‘shtirnoqqa oladi. Ma’lumki, qo‘shtirnoqning ochilish va yopilish belgilari bir-biridan farq qiladi, ular bo‘s sh joy bilan ajratib ko‘rsatiladi. Maxsus kod yoki belgilanishi yordamida ham yozish mumkin. Misol:

quotes: « »

▪ **resize** — foydalanuvchiga <textarea> elementining o‘lchamini o‘zgartirish imkoniyatini yaratadi. Qiymati aniqlangan tur:

- **none** — o‘lcham o‘zgartirish mumkin emas;
- **both** — har ikki yo‘nalishda o‘lchamini o‘zgartirish mumkin;
- **horizontal** — faqat yotiq yo‘nalishda o‘lchamni o‘zgartirish mumkin;
- **vertical** — faqat tik yo‘nalishda o‘lchamni o‘zgartirish mumkin;

▪ **tab-size** — tabulyator nechta belgi enicha o‘lchamni egallashini ko‘rsatadi. Qiymati butun son, odatda, 8 qabul qiladi. Lekin oddiy matnda u faqat bitta bo‘shliq bilan ko‘rsatiladi. Ushbu xususiyat qanday tatbiq etilishini ko‘rish uchun <pre> elementi ichidagi matnga qarash lozim.

9.6. Ichki matnni jilolash

Obyekt ichidagi matnning qanday ko'rinishda namoyon bo'lishini quyidagi xususiyatlar yordamida boshqarish mumkin:

- **color** — Matn rangi, qiymati: rang.

- **hyphens** — belgilangan kenglikka sig'magan matndagi so'zlarning keyingi qatorga qanday tarzda bo'g'in lab tushirilishini aniqlaydi. Qiymati aniqlangan tur:

- **none** — maxsus «­» belgilari qo'yilgan bo'lsa ham, bo'g'in ko'chirilmaydi;
- **manual** — odatiy qiymat, faqat «­» belgisi yoki «**wbr**» tegi qo'yilgan joylarda bo'g'in ko'chiriladi.
- **auto** — brauzerning o'zi aniqlaydi. Ya'ni maxsus qo'yilgan «yumshoq ko'chirish»lardan tashqari ko'rsatilgan til uchun brauzerning o'zida mavjud bo'g'in ko'chirish shartlari ham inobatga olinadi.

- **letter-spacing** — matndagi harflar orasidagi masofani o'zgartiradi, odatiy qiymati: normal. O'lcham sifatida oraliq masofani ko'rsatiladi. Harfning kattaligiga mos ravishda «em» yo «ex»da qiymat berish tavsiya etiladi.

- **line-height** — satrning eng kichik balandligini aniqlaydi. Agar satrda tegi qatnashgan bo'lsa, ushbu xususiyat ahamiyatini yo'qotadi. Qiymatini o'lcham yoki foizda ko'rsatish mumkin. Odatiy qiymati **normal** bo'lib, brauzerning o'zi aniqlaganday namoyon bo'lishini ta'minlaydi.

▪ **font-family** — husnixat nomi. Bir nechta nom vergul bilan ajratib ko'rsatiladi. Agar nom bir nechta so'zdan iborat bo'lib, bo'shliq bilan ajratilsa, u qo'shtirnoq ichida yozilish kerak. Ro'yxatdagi nomlardan birinchi mavjud bo'lgan husnixat tatbiq etiladi. Odatiy qiymati brauzerda belgilanadi, ko'p hollarda u «Calibri» yoki «Times New Roman» bo'ladi.

▪ **font-kerning** — harflar orasidagi maxsus masofa. Agar hamma harf orasida bir xil masofa bo'lsa, ikki tomonga egilib yoziladigan harflar oralig'i boshqalariga nisbatan kattaday tuyuladi. Masalan: «УЛ» yozilganda «Ү»ning o'ng tepasi va «Л»ning chap pastki qismi bitta tik chiziqda joylashgan, lotin yozuvidagi «IL» so'zida esa bunday masofa qo'yib bo'lmaydi, sababi, harflarning bir-biriga yaqin har bir nuqtasi yuqorida pastgacha o'zaro bir xil masofada joylashgan. Ushbu maxsus oraliq qanday qo'yilishi husnixatning o'zida aniqlangan bo'ladi. Qiymati aniqlangan tur:

- **auto** — agar husnixatda maxsus oraliqni qo'yish imkoniyati mavjud bo'lsa, bu holda o'lcham ki-chik bo'lgan hollarda o'qish qulay bo'lish uchun uni o'chiradi. Boshqa hollarda o'zi sozlab me'yordagi oraliqni o'rnatadi.
- **normal** — doim maxsus oraliq tatbiq etiladi.
- **none** — maxsus oraliq qo'llanilmaydi.

▪ **font-size** — husnixat o'lchami. O'lchamni maxsus so'zlar bilan ham ifodalash mumkin: xx-small, x-small, small, medium, large, x-large, xx-large. Bular HTMLdagi 1 dan 7 gacha ko'rsatiladigan qiyoziy qiymatlarning o'rniga qo'llaniladi.

▪ **font-stretch** – harflarning eni qanday joy egallashini aniqlaydi. Qiymati aniqlangan tur bo'lib, quyidagi qiyosiy so'zlar bilan beriladi: normal, ultra-condensed, extra-condensed, condensed, semi-condensed, expanded, semi-expanded, extra-expanded, ultra-expanded.

▪ **font-style** – husnixat shakli, aniqlangan tur:

- **normal** – odatiy ko'rinishda;
- **italic** – yozma harflar bilan ifodalangan;
- **oblique** – harflar o'ngga og'dirilgan shaklda akslantirilgan;

▪ **font-variant** – yozuvni bosh harflarda ifodalashni ta'minlaydi, qiymati:

- **normal** – odatdagidek ko'rinishni saqlaydi;
- **small-caps** – barcha kichik hafrlarni bosh harflar yordamida ifodalaydi, ammo o'lchami o'z holicha (kichikligicha) qoladi;

▪ **font-weight** – harflarni ifodalovchi chiziqlarning qalnligi va rangining to'qlik darajasini aniqlaydi. Qiymati «bold», «bolder», «lighter», «normal» kabi qiyosiy so'zlar yordamida yoki 100 dan 900 gacha yuzlik sonlar bilan berilishi mumkin.

▪ **font** – kengaytirilgan xususiyati yuqorida keltirilgan husnixatga doir xususiyatlarni bordaniga ko'rsatishga imkon beradi. O'lchamini ko'rsatayotganda satrning balandligini ham berish mumkin. Umumiy sintaksisi quyidagicha:

font: [font-style font-variant font-weight font-stretch]
font-size [/line-height] font-family

- **list-style-image** – ro'yxatga nishoncha sifatida rasm biriktiradi. Qiymatiga fayl manzili **url** funksiyasi orqali beriladi.

- **list-style-position** – ro'yxat nishonchasingin joylashtish o'rnnini belgilaydi. Qiymati aniqlangan tur:

- **inside** – nishoncha matn boshlanadigan nuqtaga qo'yiladi, ya'ni u ikkinchi qatordan boshlab keyingi satrlar bilan chapdan bir tekis joylashadi.
- **outside** – odatiy qiymat, matnning satrlari tekislangan chap tomondan nishoncha oldinga ajralib chiqib turgan holda joylashadi.

- **list-style-type** – nishonchaning ko'rinishi. Qiymati aniqlangan tur:

- **circle** – aylana sifatida;
- **square** – kvadrat shaklida.
- **disc** – yumaloq nuqta sifatida (tegi uchun odatiy qiymat);
- **armenian** – an'anaviy arman raqamlari bo'yicha;
- **decimal** – arab raqamlari bilan (1, 2, 3, 4, ...);
- **decimal-leading-zero** – arab raqamlari yordamida, faqat o'ndan kichik sonlarning oldi nol bilan to'diriladi (01, 02, 03, 04, ...);
- **georgian** – an'anaviy gruzin raqamlari bilan;
- **lower-alpha** – kichik lotin harflari orqali (a, b, c, d, ...);
- **lower-greek** – kichik grek harflari yordamida (α, β, γ, δ, ...);

- **lower-latin** – «lower-alpha» singari;
 - **lower-roman** – kichik rim raqamlari orqali (i, ii, iii, iv, v, ...);
 - **upper-alpha** – bosh lotin harflari bilan (A, B, C, D, ...);
 - **upper-latin** – «upper-alpha» singari;
 - **upper-roman** – bosh rim harflari orqali (I, II, III, IV, V, ...);
 - **none** – nishoncha qo'yilmaydi.
- **list-style** – ro'yxat shaklini aniqlaydi, bu kengaytirilgan xususiyatda «list-style-type», «list-style-position» va «list-style-image»larning qiymatlarini umumlashtirib ko'rsatish mumkin. Ro'yxat sifatida <dd>, <dt>, , va teglarining ichidagi matni yoki qiymati «display: list-item» bo'lgan obyekt qaraladi.
 - **opacity** – shaffoflik, odatiy qiymati: 1, unda hech o'zgarishsiz akslanadi. Qiymat sifatida 0 dan 1 gacha qism sifatida shaffoflik darajasini ko'rsatadigan haqiqiy son qabul qiladi. 0 butkul shaffoflikni anglatib, ko'rinnmaydigan darajada tiniq oyna misoli bo'ladi. Xiralikni shaffoflik bilan chalkashtirmaslik kerak. Xira element orqasida nima borligi ko'rinnmaydi, shaffof element qanchalik xira bo'lsa, orqasidagi obyektlar shunchalik yaqqol ko'rindi.
 - **text-align** – matnning yotiqlik tekislanishini aniqlaydi. Qiymati aniqlangan tur:
 - **center** – matn qatorning o'rtasiga nisbatan simmetrik tarzda joylashadi;
 - **justify** – so'zlar orasidagi bo'sh masofalar

kengaytirilib, matnning o'ng va chap tomoni bo'yicha tekis joylashtiriladi;

- **left** – faqat chap taraf bo'yicha tekis joylashtiriladi (odatiy qiymat);
- **right** – faqat o'ng taraf bo'yicha tekis joylashtiriladi;
- **start** – matn yo'nalishi chapdan o'ngga bo'lsa, chapga; aksincha bo'lsa, o'ngga tekislaydi;
- **end** – «start»ning aksi («dir»ning qiymatiga bog'liq).

▪ **text-align-last** – agar «text-align: justify» qiymati berilgan bo'lsa, oxirgi qatorni qanday tekislash kerakligini ko'rsatadi. Qiymati aniqlangan tur:

- **start** – satrning boshlang'ich tomoniga nisbatan tekislaydi (odatiy qiymat);
- **end** – yakuniy tomoniga nisbatan tekislaydi («dir»ning qiymatiga bog'liq);
- **left** – aynan chap tomonga nisbatan tekislaydi;
- **right** – aynan o'ng tomonga qarab tekislaydi;
- **center** – satrni o'rtaga tekislaydi;
- **justify** – kenglik bo'yicha har ikki tomonga nisbatan tekislaydi, agar faqat bitta so'zdan iborat bo'lsa, u chap tomonda bo'ladi;

▪ **text-decoration-line** – matnning qayerida chiziq chizilishini aniqlaydi. Qiymati:

- **line-through** – matn (har bir harf) ustidan chiziladi;
- **overline** – matn tepasidan chiziladi;
- **underline** – matn ostidan chiziladi;

- **none** – odatiy qiymat, chiziqni bekor qiladi (ishorat uchun ham taalluqli);
- **text-decoration-style** – matn ostiga chiziladigan («text-decoration-line» orqali berilgan) chiziqning shakli. Qiymati aniqlangan tur:

- **solid** – bitta to'g'ri chiziq;
- **double** – ikkita parallel to'g'ri chiziq;
- **dotted** – nuqtalardan tashkil topgan chiziq;
- **dashed** – uzuq chiziq;
- **wavy** – to'lqinli chiziq;

- **text-decoration-color** – «text-decoration-line» orqali berilgan chiziqning rangi. Odatiy qiymati matnning o'z rangi bilan bir xil.

- **text-decoration** – kengaytirilgan xususiyat bo'lib, «text-decoration-line», «text-decoration-style» hamda «text-decoration-color»lar qiymatlarini birdaniga berish imkoniyatini yaratadi.

- **text-overflow** – blokli element ichidagi matn ko'r-satilgan o'lchamga sig'magan taqdirda uni qanday kesib ko'rsatishni ifodalaydi. Ushbu xususiyat o'z-o'zidan «overflow: visible» berilgan holda ishlamaydi. Ikki xil qiymat qabul qiladi:

- **clip** – matnning maydonga sig'adigan qismi kesib ko'rsatiladi, xolos;
- **ellipsis** – matn maydonga sig'masa, kesib, oxiriga uch nuqta qo'shiladi;

- **text-indent** – xatboshining qolgan qatorlarga nisbatan qancha ichkarida bo'lishini ko'rsatadi. Qiymati o'lcham yoki foiz hisobida beriladi.

▪ **text-shadow** — matnning soyasi. Odatiy qiymati: «none», u soyasiz ko'rinishni ta'minlaydi. Umuman olganda, soya qiymati quyidagicha beriladi:

Sintaksis: **text-shadow: x y r c, x y r c**

Misol: **text-shadow: 1px 1px 2px black, 0 0 1em red;**

Bunda **x** — x o'qi bo'yicha siljish; **y** — y o'qi bo'yicha siljish; **r** — soya yoyilish radiusi; **c** — rangi; Bir necha qiymatlar vergul bilan ajratib beriladi.

▪ **text-transform** — matnni bosh yoki kichik harfli yozuvga aylantiradi. Qiymatlari:

- **capitalize** — har bir so'zdagi birinchi belgining bosh harfda yozilishini ta'minlaydi, qolgan harflar o'z holicha qoladi;
- **lowercase** — barcha yozuv kichik harflar bilan yoziladi;
- **uppercase** — barcha yozuv bosh harflar bilan yoziladi;
- **none** — odatiy qiymat, o'z holicha akslanadi;

▪ **unicode-bidi** — «direction»da ko'rsatilgan yo'nalish bo'yicha har bir harfning o'rnnini o'zgartirib akslantiradi. Masalan, arab tilida o'ngdan chapga qarab yoziladi. Buning uchun matnni qaytadan terib chiqish shart emas, ayniqsa, dasturiy ta'minot bilan tarjima qilinib yozilganda. Qiymati aniqlangan tur:

- **normal** — odatiy qiymat, brauzer yunikod belgilari bo'yicha aniqlaydi.
- **bidi-override** — «direction»da ko'rsatilgan yo'nalish bo'yicha har bir harf o'rni almashtiriladi.

- **embed** — xuddi «direction»da ko'rsatilganday akslanadi. (Xatboshi oxiridagi nuqta qator bo'shiga chiqib qoladi.)
- **white-space** — bo'sh joylarning qanday akslanishi ni ko'rsatadi. Qiymati:
 - **normal** — odatiy qiymat, barcha bo'sh joylar va qatorlarni ajratuvchilari yagona bo'shliq sifatida ifodalanadi. Qatorlarga bo'lishni o'zi aniqlaydi;
 - **nowrap** — qatorlarga bo'lish bo'sh joylar bo'yicha amalga oshirilmaydi, faqat
 tegi inobatga olinadi;
 - **pre** — qanday yozilgan bo'lsa, shundayligicha namoyon bo'ladi;
 - **pre-line** — eni sig'masa, qatorlarga bo'linadi va har bir qatorning boshi va oxiridagi bo'sh joylar o'chiriladi;
 - **pre-wrap** — yozilgani kabi akslanadi, faqat eni bo'yicha keyingi qatorga tushirilishi mumkin;
- **word-break** — so'z bo'lagini keyingi qatorga tushishga ruxsat berish. Qiymati:
 - **normal** — odatiy qiymat, so'z bo'lagining keyingi qatorga tushishiga ruxsat bermaydi, agar aniq
 tegi yoki ajratuvchi belgi qo'yilmagan bo'lsa;
 - **break-all** — matn ko'rsatilgan kenglikka si-g'ishi uchun so'zni bo'laklarga ajratishni o'zi aniqlaydi;
 - **keep-all** — keyingi qatorga so'zning bo'lagi tushishini xitoy, koreys va yapon tillari uchun

taqiqlaydi. Qolgan hollarda «normal» singari ishlaydi.

▪ **word-spacing** — so'zlar orasidagi masofani aniqlaydi. Qiymati: o'lcham. Mabodo «text-align: justify» berilgan bo'lsa ham, so'zlar orasidagi masofa ushbu xususiyatda ko'rsatilgandan kichik bo'lmaydi.

▪ **word-wrap** — ma'no jihatidan «word-break» bilan bir xil vazifa bajaradi. Farqi shundaki, «word-break»da to'laligicha joriy qatorga sig'magan so'zning o'zi keyingi qatorga olinmay, uning qismigina ko'chiriladi; ushbu «word-wrap»da esa avval so'zning o'zi tushiriladi, shunda ham kenglik yetmagan holda uning qismi keyingi qatorga yoziladi. Qiymati aniqlangan tur:

- **normal** – faqat
 tegi yoki unga o'xshagan ajratuvchilar qo'yilgandagina keyingi qatorga o'sha joydan ko'chiriladi;
- **break-word** – berilgan kenglikka qarab so'z yoki uning qismi tushiriladi;

▪ **writing-mode** — yozuvning tik yoki yotiq bo'lishini ta'minlaydi. Qiymati:

- **horizontal-tb** – odatiy qiymat, yotiq chapdan o'ngga va tepadan pastga yo'nalgan;
- **vertical-rl** – tik tepadan pastga va o'ngdan chapga;
- **vertical-lr** – tik tepadan pastga va chapdan o'ngga;
- **sideways-rl** – tik tepadan pastga, hamma harflar o'ngga «yonboshlaydi»;
- **sideways-lr** – tik tepadan pastga, hamma harflar chapga «yonboshlaydi»;

9.7. Chop etish xususiyatlari

Chop etish qurilmalari turlicha bo'lganligi bois ko'-pincha varaqning chetlaridan turlicha joy tashlashga yoki o'zgacha o'lchamda akslantirishga to'g'ri keladi. Chop etilayotgan sahifa uchun qonuniyatlar «@page» kalit so'zi yordamida beriladi. Undan keyin qonuniyatlar qaysi sahifalar uchun tatbiq etilishi ko'rsatilishi mumkin: «:first» – faqat birinchi sahifaga; «:left» – chap (juft o'rindagi) sahifalarga; «:right» – o'ng (toq o'rindagi) sahifalarga. Misol:

```
@page :first { top: 3.5sm; }
```

Ba'zi xususiyatlар faqat chop etilayotgandagina inobatga olinadi. Masalan:

- **marks** – varaqlarga bo'luvchi belgi o'rnatadi. (Ta'kidlab o'tish lozimki, hozircha bu ajoyib xususiyatni brauzerlar qo'llamaydi.) Qiymati aniqlangan tur:
 - **crop** – obyekt chop etilgandan so'ng navbatdagisini keyingi varaqqa tushiradi;
 - **cross** – keyingi varaqdagi obyekt ham joriysining bir bo'lagiligini anglatadi. Misol uchun, bir nechta varaqqa bitta katta rasmni chop etish.
 - **none** – odatiy qiymat, ajratuvchi qo'yilmaydi.
- **orphans** – bir nechta varaqqa joylashadigan matnlar uchun oldingi varaqda kamida qancha qator chop etilishi kerakligini ko'rsatadi. Qiymati: butun son.

- **page-break-after** — chop etish jarayonida ko'rsatilgan elementdan so'ng keyingi obyektlarning boshqa varaqqa chiqarilishini ko'rsatadi. Qiymati aniqlangan tur:
 - **always** — doim varaqlarga ajratilishini ta'minlaydi;
 - **auto** — ajratish faqat zarur bo'lgandagina bajariladi (odatiy qiymat);
 - **avoid** — elementdan so'ng ajratilmasdan chop etilishini ko'rsatadi;
 - **left** — keyingisi juft o'rinda bo'lishi uchun bir yoki ikki varaq bo'sh chiqaradi;
 - **right** — keyingisi toq o'rinda bo'lishi uchun bir yoki ikki varaq bo'sh chiqaradi;

Har bir yotiq chiziqdan keyin yangi varaqdan chop etishni ta'minlovchi misol:

```
hr { page-break-after: always; }
```

- **page-break-before** — ko'rsatilgan elementdan boshlab keyingi varaqda chop etilishini ta'minlaydi, qiymati «page-break-after»niki singari.

- **page-break-inside** — ko'rsatilgan element turli varaqlarda chop etilish holatini aniqlaydi. Qiymati «**avoid**» bo'lsa, element faqat bitta varaqda chiqishini ta'minlaydi. Odatiy qiymat «**auto**» berilsa, zarur bo'lgan holda, bo'lib chop etiladi.

- **widows** — bir necha varaqda chop etiladigan sahifa uchun navbatdagi varaqda eng kamida nechta qator bo'lishi lozimligini ko'rsatadi. Ushbu xususiyat qiymati butun son bo'lib, ustuvor hisoblanadi va u «orphans»nikiga zid bo'lsa ham, inobatga olinadi.

9.8. Hodisalarni boshqarish

- **pointer-events** — sichqoncha bosilgandagi hodisa ning bajarilishini boshqaradi. Qiymati aniqlangan tur:
 - **auto** — hodisa odatdagidek bajariladi;
 - **none** — hodisa bajarilishini to'xtatadi;
- **user-select** — foydalanuvchi tomonidan matnni belgilashni boshqaradi. Ba'zida sarlavhani anglatuvchi yoki alohida ahamiyat kasb etuvchi matnlarni (nusxa olish uchun) belgilash maqsadga muvofiq emas, shunday hol-larda ushbu xususiyatni qo'llash mumkin. Ammo ko'p brauzerlar hozircha buni to'g'ridan to'g'ri qo'llamaydi, «Internet Explorer»ga «-ms-user-select»; «Mozilla Firefox» uchun «-moz-user-select»; qolgan brauzerlar-da esa «-webkit-user-select» kabi yozish taqozo qilingadi. Qiymatlari:
 - **auto** — tashqi element xususiyatiga mos olinadi;
 - **none** — belgilash ta'qilanganadi;
 - **text** — faqat matnni belgilash mumkin bo'ladi;
 - **all** — butun obyektni ichki elementlari bilan to'laligicha belgilaydi;
 - **contain** — faqat element chegarasining ichida belgilashga ruxsat beradi;

Misol:

```
.sarlavha {  
    -moz-user-select: none;  
    -webkit-user-select: none;  
    -ms-user-select: none;  
    user-select: none; }
```

9.9. Vaziyatan sinflash

CSSda vaziyatan sinflash bir qarashda tabaqlashga o'xshab ketsa-da, mohiyatan undan keskin farq qiladi. Tabaqlashtirishda ayni paytdagi holatiga qarab shamoil beriladi, vaziyatan sinflashda esa faqat ma'lum vaziyatlardagina yoki elementning ajratilgan ma'lum bo'laklarigagina ta'sir qiluvchi shakl-siymo beriladi. Ularni farqlash uchun tabaqlashtirishda «::» ishlatsa, vaziyatan sinflashda «::» qo'llaniladi. Masalan, **content** xususiyati faqatgina elementning ikki tomondan chegarasini anglatuvchi «::after» yoki «::before» vaziyat sinflari bilangina qo'llaniladi. Bu ikkisi mos ravishda obyektning ortidan yoki oldidan qo'shimcha jilo berishda foydalaniladi. Faqat shu ikkilasi uchun ketma-ket ikkita ikki nuqta o'rniqa bitta qo'ysa ham bo'ladi.

Agar qandaydir ro'yxatga maxsus joziba berish bilan birga unga tartib raqami ham qo'shish lozim bo'lsa, uni CSS imkoniyatlari yordamida ham bajarish mumkin. Buning uchun ro'yxatni ifodalovchi tegini qo'llash ham talab etilmaydi. Ushbu imkoniyatni amalga oshirish uchun xizmat qiladigan xususiyatlar bilan quyida qisqacha tanishib o'tamiz:

- **counter-reset** – sanoq saqlanadigan o'zgaruvchini aniqlaydi. Qiymati:

- **none** – sanoq amalga oshirilishini taqiqlaydi;
- **inherit** – sanoq qiymati tashqi elementdagidan olinadi;
- **identifikator son** – sanoq saqlanadigan o'zgaruvchining nomi va boshlang'ich qiymati. Bir

nechta ko'rsatilishi lozim bo'lsa, bo'sh joy bilan ajratiladi.

▪ **counter-increment** — sanoqning keyingi qiymatini hisoblaydi. Qiymati:

- **none** — sanoqning oshirilishini taqiqlaydi;
- **identifikator son** — ko'rsatilgan o'zgaruvchi qiymatini berilgan songa oshiradi.

▪ **content** — teg ichidagi matnga qo'shimcha ravishda akslantiriladigan yozuv. Qiymat sifatida ixtiyoriy satr berilishi mumkin. Agar tegning biror alomati qiymatini ko'rsatish lozim bo'lsa, «**attr(alomat)**» funksiyasidan foydalaniladi. Qo'shtirnoq qo'shish lozim bo'lsa, uni aynan yozish o'rniga «**open-quote**» yoki «**close-quote**» so'zlari dan foydalanish mumkin. Satr qiymat qaytaradigan funksiyalar ishlatsa, uni qo'shtirnoqqa olingan oddiy satr bilan hech qanday matematik amalsiz biriktirish mumkin.

Maxsus funksiya **counter** yordamida ko'rsatilgan o'zgaruvchining joriy qiymati olinadi. Endi shu ma'lumotlar yordamida sanoqli ro'yxat tuzamiz:

```
<div>
    <h1> HTML 5 </h1>
    <h1> CSS 3 </h1>
    <h1> Veb-sahifa </h1>
</div>
div { counter-reset: sanoq 1; }
h1::before {
CSS      counter-increment: sanoq 1;
          content: counter(sanoq) "-bo'lim. ";
}
```

	2-bo'lim. HTML 5
Natija	3-bo'lim. CSS 3
	4-bo'lim. Veb-sahifa

Ushbu misolda har bir `<h1>` tegidagi yozuv oldidan nechanchi bo'limligini qo'shib chiqdik. CSS — HTMLda eng kam kod yozishni ta'minlab beruvchi vosita. Shunday ekan, bizga qulaylik yaratuvchi yana bir nechta vaziyat sinflari bilan tanishib chiqamiz:

- ◆ **::first-letter** — Matnning faqat birinchi harfiga tatbiq etiladigan sinf. Ayrim ro'znomalarda birinchi harflar alohida ajratib ko'rsatilishiga guvoh bo'lganmiz. Shunday talab qo'yiladigan bo'lsa, faqatgina CSS yordamida buni amalga oshirish mumkin. HTMLni tahrirlash har doim ham oson kechavermaydi, ayniqsa, u boshqa dasturiy ta'minotlar yordamida hosil qilingan bo'lsa.

- ◆ **::first-line** — birinchi qatorga tatbiq etiladi. Bu qonun hujjatlarida modda raqami va nomini ajratib ko'rsatishda qo'l keladi. Umuman olganda, keng tatbiq etish mumkin.

- ◆ **::selection** — matnning belgilangan qismigagina tatbiq etiladi. Ta'kidlab o'tish lozimki, ushbu sinf CSS3 dan chiqarib tashlangan bo'lib, uni rivojlantirish CSS4 uchun qoldirilgan. Hozircha «Mozilla Firefox» buni to'g'ridan to'g'ri qo'llamaydi, alohida «`::-moz-selection`» tarzida yozilsa, qo'llaydi. Umid qilamiz, bu noqulaylik yaqin vaqtlarda hal etiladi. Agar (nusxa olish uchun) belgilangan matnning bo'lagi boshqacha tarzda ko'rinish turishi lozim bo'lsa, quyidagicha qonuniyat yozishga to'g'ri keladi:

```
p::selection {  
    background-color: #62BA21;  
    color: white;  
}  
p::-moz-selection {  
    background-color: #62BA21;  
    color: white;  
}
```

«Internet Explorer» o'zi uchun bir qancha vaziyat sinflarini kiritgan, ammo boshqa brauzerlar uni qo'llamagani, hatto Microsoft ham u boshi berk ko'chaga kirib qolgani uchun boshqa «Edge» brauzeri ishlab chiqani bois biz uning sinflariga to'xtalib o'tishni lozim topmadik.

9-BO'LIM

MUKAMMAL SAYT YARATISH TALABLARI

Mukammal sayt yaratish talab etilar ekan, yozilgan kod barcha brauzerlarda birday namoyon bo'lishi shart. HTML4da bu juda katta muammo bo'lgani bois maxsus «DOCTYPE» kiritilgan edi. Ammo u ham samarali chora emasligi amalda ko'rindi. O'n yillik muzokalarlardan keyin hayotiy tajribalar asosida yaratilgan HTML5 chiqqaniga va ommaviy qo'llanilayotganiga qaramay, haligacha ba'zi qonuniyatlar ayrim brauzerlarda kutilganidek akslanmayapti. Shunday ekan, ularning o'rniga universal usulni topib qo'llashga to'g'ri keladi. Ushbu talabni inobatga olgan holda yuqoridagi tanishib o'tgan bilimlarimiz asosida veb-saytlar yaratishda keng qo'llaniladigan bir necha tajribalar bilan tanishib o'tamiz.

10.1. Bexato kod yozish

Yuqorida ta'kidlaganimizdek, HTML va CSS dasturlash tili bo'limganligi bois xato yoki nuqsonli yozilgan kodlarni brauzer imkon qadar tahlil qilib, munosib

qonuniyat bo'yicha namoyish etadi. Umuman iloji bo'limgan holda esa e'tiborsiz qoldiradi. Lekin bu mas'uliyatsizlik bilan kod yozishga yondashishga ruxsat bermaydi. Mukammal dasturlashni o'rganishga bel bog'lagan ekanmiz, koddagi har bir belgi nima uchun qo'yilganini anglashimiz va uning o'rini ekanligiga ishonchimiz komil bo'lishi kerak.

Agar oldindan yaratilgan sahifani takomillashtirish lozim bo'lsa, avval uning barcha qoidalar bilan to'g'ri yozilganligini tekshirib ko'rish, so'ngra esa qo'shimchalar kiritish tavsiya etiladi. Aks holda, topish juda mushkul bo'lgan xatoliklar, chalkashliklar kelib chiqishi mumkin. Masalan, bitta faylda qaysidir blokli elementning yopuvchi tegi ko'rsatilmagan, uning o'rniga yanglishib yana tegning ochish qismi yozilgan. U faylni akslantirayotganda foydalanuvchi bu kamchilikni sezmaydi, lekin bu kod to'g'ri yakunlangan deb hisoblagan holda davomidan boshqa fayldagi kodlarni unga birlashtirganda buzilish yuzaga keladi. Buni oddiy matematik amallarni bajarishda qavslarni to'g'ri qo'ymaslik oqibatida hisob natijasi turlicha chiqishi bilan qiyoslash mumkin.

Kod to'g'ri yozilganligini tekshirish uchun brauzerlarning o'z imkoniyatlaridan yoki ularga qo'shimcha yuklanadigan «Html Validator» kengaytmasidan foydalanish mumkin. Interfeysi sodda va turli brauzerlarda turlicha bo'lganligi bois qanday foydalanishni batatsil yoritishga to'xtalib o'tmaymiz. Agar veb-sahifa internetda joylashgan bo'lsa, «<http://validator.w3.org/>» sayti orqali ham tekshirish mumkin.

Kod yozuvchilarning bee'tiborligi, loqaydligi yoxud mukammal bilimga ega emasligi bois eng ko'p yo'l qo'yadigan kamchiliklarini sanab o'tishni lozim topdik:

♦ **DOCTYPE** ko'rsatilmagan. HTML faylning birinchi qatori «DOCTYPE» ko'rsatishdan boshlanishi kerak. [2.2.] [2.3.]

♦ **Imlo xato.** Tegning nomi yoki alomati noto'g'ri yozilgan. Odatta, bunday yozuvlar e'tiborsiz qoldiriladi, lekin AngularJS foydalanuvchilari biladiki, obyektga mu-rojaat qilish yoki uning ko'rinishini o'zgartirish uchun unga maxsus nomdagi HTMLda mavjud bo'lмаган alomatlar atayin yoziladi. Albatta, undan foydalanuvchilar yetarlicha, ammo o'z qoidalarini singdirish uchun bunday «buzg'unchilik» qilish maqsadga muvofiq emas.

♦ **Majburiy alomat.** Ba'zi teglarning ma'lum alomatlari doim ko'rsatilishi shart. Masalan, «link» tegi ishorat ko'rsatish uchungina qo'llaniladi, unda «href» bo'lishi kerak.

♦ **Notanish nom.** Ba'zi teglar yoki alomatlar, aslida, HTML qonuniyatlarida mavjud emas, lekin talabga qarab brauzerlarning o'zлари kiritishgan. Keyinchalik ommalashib, ko'p boshqa brauzerlar ham qo'llay boshlagan. HTML5da ularning bir qanchasi yangilik sifatida qo'shilgan bo'lsa-da, ayrimlari CSS orqali ifodalash mumkin bo'lганligi bois «tan olinmagan». Masalan, Microsoft kiritgan «marquee» [8.7.] tegi yuguruvchi satrni ifodalash uchun barcha brauzerlarda ishlaydi, ammo notanish nom sifatida qo'llash tavsiya etilmaydi.

♦ **Noo'rin teg.** Yuqorida takidlaganimizday, qatorli teg ichiga blokli teg yozish tavsiya etilmaydi. Bundan

tashqari, bir necha teglar ketma-ket yozilayotganda yopuvchi qismlari ochuvchisiga mutanosib bo'lishi kerak. Ya'ni oldin ochilgan – keyin yopiladi. Noo'rin yoziladigan teglarning eng dahshatlisi bitta obyektni yaratish uchun qatnashadigan teglar o'z tashqi tegi ichida kelmasligidir. Masalan, jadvalda kataknini ifodalovchi «td» qatorni ifodalovchi «tr» tegi ichida bo'lishi shart.

♦ **Yopilmagan teg**. Ko'p teglarning yopuvchi qismi ko'rsatilishi shart bo'lmasa ham, uni yozish tavsiya etiladi. Aks holda, qo'shni shu nomdagi teg bilan aralashib ketishi mumkin.

♦ **Qiymat berish**. Qiymatlar son yoki bo'shliq va ajratuvchi belgi ishlatilmagan satrdan iborat bo'lsa, alomatga tenglash paytida qo'shtirnoq ishlatmasa ham bo'lishini aytib o'tgan edik. [2.1.] Lekin barcha qiymatlarni qo'shtirnoq ichida yozish qat'iy talab etiladi. Sababi, faylning kodlamasi o'zgarganda ba'zi yunikod belgilari ajratuvchi sifatida qabul qilinishi mumkin. Bundan tashqari, bir joyda qo'shtirnoq qo'yib, boshqa joyda qo'ymaslik nafaqat chalkashlik keltirib chiqaradi, balki kodning chiroyli yozilishiga ham putur yetkazadi.

♦ **Maxsus belgi**. Maxsus belgilari «&» belgisi yordamida raqamli kodi yoki qisqa nomi orqali ifodalanadi. [2.1.] Oxirida esa «;» yozilmasa ham, kutilganday o'girib olish amalga oshsa-da, aslida, uni qo'yish talab etiladi.

♦ **Takrorlanish**. Ko'pincha texnik xatolik tufayli bitta nom qayta-qayta yozib yuboriladi. Masalan: «**b****b**» kabi yozuvlar ortiqcha takrorlanishdan boshqa narsa emas. Zero, qalin yozuvni ta'minlash uchun

bu kabi teglar ketma-ket yozilmaydi. Yoki tegning ichida ayni nomdag'i alomatga bir necha bor qiyamat berish ham o'rinsizdir! Odatda, bunday hollarda oxirgi ko'rsatilgan qiyamat inobatga olinadi, lekin bu ham aniq qoida emas.

♦ **Ortiqcha teg**. HTMLda bir-biriga o'xshash sahifalar yaratayotganimizda, albatta, nusxa ko'chirishdan foydalanamiz, ba'zan bir necha joydan nusxalangan kodlar ichida umumiy olib qaraganda ortiqcha hisoblanadigan teglar kelib qolishi tabiiy. Yozilgan kodni albatta nazoratdan o'tkazish kerak. Masalan, «body» va «head» kabi teglar bitta veb-sahifa uchun yagona «html» tegi ichida faqat bittadan qatnashishi shart. Yoki yagona «div» kutilganidek natija bera oladigan vaziyatda uni yana bir necha blokli tegga olish samarasiz va faqat brauzer ishini qiyinlashtiradi, xolos.

♦ **Noo'rin yozuv**. Keyingi kitoblarimizda HTML faylni boshqa dasturiy ta'minotlar orqali yaratish, uning kodlarini maxsus buyruqlar yordamida hosil qilishni ko'rib o'tamiz. Bunda arzimagan yanglishish tufayli teglar yozilishi va joylashishi o'zgarib ketaveradi. Aslida, barcha ko'rinadigan obyektlar «body» tegining ichida joylashishi shart [2.5.], uning tashqarisidagi yozuvlar ham ayrim brauzerlarda akslantirilganligi bois ba'zilar e'tibor qaratmasligi mumkin. Dastur tuzish jarayonida bu kabi noo'rin yozuvlarga yo'l qo'yilmasligi kerak.

10.2. Moslashuvchan CSS yaratish

Yuqorida veb-sahifaning ko'rinishini bitta CSS faylni almashtirish bilan butkul o'zgartirib yuborish mumkinligini ta'kidlab o'tdik. Umuman olganda, CSS yaratish dizaynerlarning ishi va ularning ko'p yillik tajribasi veb-sayt yaratishda ikkita, nari borsa, uchta rangni asos qilib barcha obyektlarni unga mutanosib tusda akslantirish kutilgan sarama berishini ko'rsatdi. Bunda birinchi rang asosiy ko'zga tashlanuvchi obyektlarga beriladi, ikkinchisi ortrang sifatida qo'llaniladi. Ko'pincha ortrang oqligicha qoladi, ammo oq ko'zni toliqti-ruvchi bo'lganligi bois uni ham o'zgartirishga talab bo'lmoqda. Uchinchi rang esa husnixatning ko'rinishi uchun beriladi. Qolgan tuslar, soyalar, chegaraviy chiziqlar shu ranglarga mutanosib ravishda sal to'q yoki sal och tarzda berilishi maqsadga muvofiq. Keyin husnixatlar balandligi, tugmalar o'lchami va maydonlar hajmi ekranda joylashishiga qarab ko'rsatiladi.

Shunday vaziyatlar bo'ladiki, sayt tayyor bo'lgandan keyin tanlangan asosiy rang yoqmay qolishi, uning o'rninga boshqasi maqbulday tuyulishi mumkin. Bunday vaziyatlarda CSS faylni boshqatdan yangidan yaratib chiqish malol kelishi aniq. Ushbu muammoni yechish uchun «SASS»dan foydalaniladi.

Mukammal dasturlashning asosiy talablaridan biri bitta kodni qaytarib yozmaslikdir. HTML dasturlash tili bo'limgaganligi uchun bunday muammoga doim duch kelinadi. Dasturlash tillari yordamida HTML fayl yaratib olish keng qo'llaniladi. Bunday uskunalar ichida

Xampton Katlin va Natan Veyzenbaum tomonidan yaratilgan **Haml** (*HTML abstraction markup language*) nomli HTML tuzib beradigan belgilanishlar tili keng ommalashgan. Keyingi kitoblarimizda veb-dasturlash tilлari bilan keng tanishib o'tamiz, shuning uchun hozir «Haml»ni bat afsil o'rganishga hojat yo'q. Lekin uning tub asosida yaratilgan «**sass**» (*Syntactically Awesome Stylesheets — sahifa shamoyilning beqiyos sintaksisi*) uskunasi CSS fayllarni moslashuvchan tarzda yaratishimizda juda asqotadi.

SASSning sintaksisi bo'yicha yangi qatorga o'tish va xatboshilar muhim ahamiyatga ega, masalan:

SASS

```
div
    color: #333
    align: center
```

CSS

```
div {
    color: #333;
    align: center;
}
```

Bunda qatorning oxiriga «;» qo'yish ham talab etilmaydi. Ammo bu biz ko'nikkan CSS sintaksisidan keskin farq qiladi, agar qatorlar o'z o'rnidagi ajratilmasa va qator boshiga bo'sh joylar to'g'ri qo'yilmasa, bunday faylni CSSga o'tkazishda xatolik yuzaga keladi. Uman olganda, «Ruby» yoki «Python»da dasturlashni bilmaydigan hamda bitta qatorga bir necha qoidalarni yozishni ma'qul ko'radiganlar uchun SASSning sintaksi noqulaylik keltirib chiqaradi. Shuning uchun o'zimiz o'rgangan CSS ko'rinishiga aynan o'xshash **SCSS** (Sassy CSS)dan foydalanamiz.

SCSS faylni CSSga o'tkazish juda oson, o'zingizning kompyuteriningizda operatsion tizimingizga mos o'giruvchini o'rnatasiz (*sass-lang.com* yoki *sass-scss.ru* saytlari ning biridan olishingiz mumkin) va bitta buyruq berish bilan kerakli CSS faylni hosil qilasiz. Agar bu ortiqcha amalday tuyulsa, tayyor onlayn o'giruvchilardan (masalan: *codepen.io*) foydalanish ham mumkin.

Quyida SCSS faylni yozish qoidalari hamda ularning CSSga qanday tarzda o'girilishi bilan batafsil tanishib o'tamiz.

10.2.1. O'zgaruvchilar

Yaratilishi lozim bo'lgan veb-sayt yoki har qanday veb-tizimni oldin tasavvurga keltirib; qanday ranglar ishlataladi, husnixatlar qanaqa bo'lishi kerak, maydon va tugmalarning shakli, undagi obyektlarning jilosi qay tarzda namoyon bo'lishi lozimligini aniqlab olgandan keyin ularni ifodalovchi o'zgaruvchilar kiritiladi. O'zgaruvchilarning nomlari «\$» belgisi bilan boshlanuvchi kelishilgan identifikator bo'lishi kerak, unda chiziqcha (-) qatnashishiga ruxsat beriladi. O'zgaruvchiga oldin qiymat beriladi, keyin shu qiymatni yozish o'rniqa o'zgaruvchining nomi keltiriladi. Masalan:

SCSS	CSS
\$main-color: darkblue;	p {
p { color: \$main-color; }	color: darkblue;
	}

Qonuniyatlarni yozish mobaynida o'zgaruvchining qiymatini almashtirish mumkin. Aytaylik, 200-qatordan keyin ichki kichik bo'limchalarni jilolaydigan qoidalar yozilgan va unda «asosiy rang» boshqacharoq bo'lishi lozim, bunday vaziyatda 200-qatorda «\$main-color: red;» deb yozib qo'yish kifoya.

10.2.2. Biriktirish

CSSda nomlanishlar ularning tatbiq etiladigan sohasiga qarab turlicha yozilishi mumkinligi bilan tanishib o'tganmiz. [8.2.] Bunday murakkab nomlanishlarni ham SCSSda ifodalashning usuli mavjud. Biriktiriladigan nomlanish tashqisining ichida joylashadi va u CSSga o'girilayotganda bo'sh joy bilan qo'shib umumlashtirib yoziladi. Ammo tabaqlash va vaziyatan sinflashni ifodalovchi so'zlar qatnashganda nomlanishda ikki nuqtadan avval bo'sh joy bo'lmasligi kerak. Bunday vaziyatda «&» belgisi ichki nomlanishning biriktiriluvchi bo'lagi oldidan qo'yiladi:

```
$main-color: magenta;  
nav {  
    color: $main-color;  
    li { display: inline-block; }  
    ~ a { text-decoration: none; }  
    &:hover { color: red; }  
}
```

```
CSS
nav { color: magenta; }
nav li { display: inline-block; }
nav ~ a { text-decoration: none; }
nav:hover { color: red; }
```

10.2.3. Fayldan yuklab olish

Ba'zan shunday jilolash qoidalari yoziladiki, uni ko'pgina CSS faylning ichiga qo'shish maqbul hisoblanadi. Universal shamoyillar, bejirim ko'rinishni ifodalovchi jozibalarni har doim birday qonuniyat asosida tatbiq etish maqsadga muvofiq keladi. Bunday hollarda qisqa qonuniyatlar to'plamini alohida faylga saqlab, uni kerak joyda yuklab olish lozim. Ixcham qonuniyatlar yozilgan ushbu faylni nomlashda birinchi harfi sifatida tagchiziq («_») qo'llash talab etiladi, albatta, kengaytmasi «.scss» tarzida bo'ladi. Bunaqa nomdag'i fayllar dasturiy ta'minot orqali o'girilayotganda e'tiborsiz qoldiriladi. Qachonki u boshqa «.scss» faylning ichida chaqirilsa, shundagina tarkibi inobatga olingan holda ko'rsatilgan joyga o'girilib «css» fayl hosil qilinadi.

Masalan, mavjud «_main.scss» nomli faylning ichida «**html, body { margin: 0; padding: 10px; }**» deb yozilgan bo'lsin. Uni quyidagicha chaqirib, tarkibidagi qoidalarni boshqa faylda tatbiq etamiz:

```
SCSS
@import 'main';
body { background-color: #efefef; }
```

```
html, body {  
    margin: 0;  
    padding: 10px;  
}  
body {  
    background-color: #efefef;  
}
```

E'tibor qaratish lozimki, ixcham faylni yuklash uchun «@import» kalit so'zi qo'llanilyapti va undan keyin qo'shtirnoq ichida faylning nomi kengaytma hamda tagchiziqsiz yozilyapti. O'giruvchining o'zi kerakli faylni nomi va kengaytmasi bo'yicha topib, tarkibidagi qonuniyatlarni hosil qilinayotgan fayl ichiga «@import» jumlesi yozilgan joyga munosib ravishdagi yerga qayta ishlab yozib beradi.

10.2.4. Funksiyadan foydalanish

Ayrim xususiyatlar turli brauzerlarda o'ziga xos nom bilan berilgandagina ishlaydi. Yuqorida «user-select» bilan tanishib o'tgan edik. [9.8.] Bunda ayni bir narsani har bir ommaviy brauzerning talabini inobatga olgan holda qayta-qayta yozib chiqishimizga to'g'ri keladi. Albatta bu, bir tomonidan, CSSning kamchiligi bo'lsa, ikkinchi tomonidan, brauzerlarning nuqsoni. Nima bo'lganda ham, hozircha bitta qoidani bir nechta qilib yozishga majburmiz. Bunday vaziyatlarda SCSSdan

foydalanilsa, kod yozish samaradorligi yanada takomillashadi.

Buning uchun ixtiyoriy nomda funksiya yaratamiz va unga o'zgaruvchi sifatida kiritiluvchi qiymatlarni ko'r-satishimiz mumkin. Funksiyani e'lon qilish uchun «@mixin» («mojaro» degan ma'noni anglatadi. *Aftidan, bunday noqulayliklar yaratuvchilarining orasida ham jiddiy kelishmovchiliklar keltirib chiqarayotgan bo'ssa kerak*) kalit so'zidan foydalaniladi. Misol:

```
SCSS @mixin userSelect($value) {  
    color: maroon;  
    user-select: $value;  
    -ms-user-select: $value;  
    -moz-user-select: $value;  
    -webkit-user-select:$value;  
}  
  
div { @include userSelect(none); }  
p h1 { @include userSelect(none); }  
  
div {  
    color: maroon;  
    user-select: none;  
    -ms-user-select: none;  
    -moz-user-select: none;  
    -webkit-user-select: none;  
}  
  
p h1 {  
    color: maroon;  
    user-select: none;  
    -ms-user-select: none;  
    -moz-user-select: none;  
    -webkit-user-select: none;  
}
```

10.2.5. Meros olish

Ko‘p hollarda bitta obyekt uchun tayinlangan qoidalaridan andoza olib, boshqa nomlanishli shartlar majmuasini yaratishga to‘g‘ri keladi. Bunda nomlanishning o‘zini chaqirib qo‘yish kifoya:

SCSS	CSS
.title { border: 1px solid #ccc; border-radius: 3px; padding: 2px 10px; color: #333; } h1 { @extend .title; font-size: 18px; } h2 { @extend .title; font-size: 16px; color: maroon; }	.title, h1, h2 { border: 1px solid #ccc; border-radius: 3px; padding: 2px 10px; color: #333; } h1 { font-size: 18px; } h2 { font-size: 16px; color: maroon; }

Bir qaraganda SCSSda yozilgan kod CSSga o‘girilgandagidan ko‘pday, meros olish samarali emasday tuyuladi. Ammo juda katta hajmli fayllarni tahrirlashda ularning ichidan andoza olinayotgan nomlanishni topib o‘zgartirish ancha mushkullik tug‘diradi. Ayniqsa, yuklanadigan ixcham faylni [10.1.3.] keyinchalik o‘zgartiravesh maqsadga muvofiq emas, qolaversa, undagi shartlar yangi «CSS» uchun boshqacha tatbiq etilishi mumkin. Umumiy qoidalar qanday nomlanishlarga taalluqli bo‘lish shartligini avvaldan to‘liq bilib bo‘lmaganligi

sababli «meroslash»dan foydalanish samara berishini amalda sinasa bo'ladi.

10.2.6. Matematik amallar

SCSSning eng foydali jihatni matematik amallarni qo'llay olishidir. O'lchamlarni oldindan kiritilgan o'zgaruvchiga nisbatan hisoblab topish tushunarli hol. Ammo hisob-kitob sodda to'rt matematik amal yordamida nafaqat sonli miqdorlar, balki ranglar uchun ham ttabiq etiladi. Bunda rang o'n otilik sanoq tizimidagi oltixonali son orqali ifodalanishi inobatga olingan. Quyida **h5** tegida yozilgan, obyektning markazida chiqishi nazarida tutilgan mavzularni tugmasifat ko'rinishda tasvirlovchi jilolashga misol keltiramiz:

```
$button-height:      24px;
$btnColor:          #eee;
$caption-color:     #333;

h5[align='center'] {
    height:           $button-height;
    width:            $button-height * 5;
    font-size:        $button-height * 12 / 10;
    margin:           6px;
    padding:          $button-height / 6;
    font-weight:      bolder;
    cursor:           pointer;
    background-color: $button-color;
    color:            $caption-color;
    border-radius:    6px;
    box-shadow:       2px 2px 3px $btnColor - #666,
                     inset 2px 2px 3px $btnColor - #333;
```

```

CSS
    &:active {
        color:           $caption-color + #222;
        box-shadow:     2px 2px 3px $btnColor - #666,
                        inset 2px 2px 3px $btnColor - #333;
    }

/* Faqat «align="center"» alomatli h5 teglar uchun */
h5[align='center'] {
    height: 24px; /* balandligi, o'zgaruvchan */
    width: 120px; /* eni, bo'yidan besh barobar uzun */
    font-size: 28.8px; /* kattaligi, h5 o'lchamiga mos */
    margin: 6px; /* tashqi hoshiya, qo'shnilaridan ajratadi */
    padding: 4px; /* ichki hoshiya, chetidan matngacha */
    font-weight: bolder; /* qalin yozuv */
    cursor: pointer; /* sichqoncha ko'rsatkichi */
    background-color: #eee; /* obyekt ortrangi */
    color: #333; /* matn rangi */
    border-radius: 6px; /* qirrasini yoysimon qilish */
    box-shadow: 2px 2px 3px #888888,
                inset 2px 2px 3px #bbbbbb; /* Soyalar */
}

/* Soyalar almashishi, tugma bosilganday ko'rinishi uchun */
h5[align='center']:active {
    /* Bosib turilgan holat, tugma tasavvurini beradi */
    color: #555555; /* Matn rangi, sal ocharadi */

    box-shadow: inset 2px 2px 3px #888888,
                1px 1px 2px #bbbbbb;
} /* Soyalar almashadi */

<!-- Tugmasifat sarlavha --&gt;
&lt;h5 align="center"&gt;HTML&lt;/h5&gt;
<!-- Tugmasifat sarlavha --&gt;
&lt;h5 align="center"&gt;CSS&lt;/h5&gt;
<!-- Oddiy sarlavha --&gt;
&lt;h5&gt; Mukammal qo'llanma &lt;/h5&gt;
</pre>

```

Natija



Mukammal qo'llanma

10.3. Modal oyna

Modal oyna shunday blokli tegning akslanishiki, namoyon bo'lувчи обьект ва унинг ichidagilardan tashqaridagi barcha elementlar nofaol tarzda bo'ladi. Foydalanuvchi yagona faol oynadagi talab etilgan amallarni bajarishini, boshqa elementlar «qotgan holda» kutib turishini taqozo etadi. Bu ba'zi qiyatlarni kiritishni talab etib, qolgan jarayonlarni ularga bog'liq ravishda bajarish lozim bo'lganda foydalanuvchi tomonidan kutilmagan xatolikka yo'l qo'yilishining oldini oladi. Demak, modal oyna akslanganda u o'z ichki elementlari bilan birga sahifadagi yagona faol obyekt bo'ladi.

Aslida, HTML5 va CSS3 imkoniyatlari bilan modal oynani yaratish juda oson. Buning uchun `<dialog>` tegi va `«::backdrop»` vaziyat sinfi mavjud. Ammo ularni «Xrom» va «Opera»dan boshqa brauzerlar qo'llamagani bois murakkabroq usuldan foydalanishga majburmiz.

Butun sahifani to'liq qoplaydigan blokli teg yaratamiz va uning eng usti qatlamda joylashishini ta'minlaymiz. Kerak bo'lganda ushbu obyektni ko'rsatamiz va undagi murojaat oynasini akslantiramiz, qolgan hollarda yashirin holatda bo'ladi. U boshqa qatlamda namoyon bo'layotgani uchun sahifadagi boshqa elementlar joylashishiga ta'sir ko'rsatmaydi. Avval modal oynani chaqiruvchi yorliq qo'yaylik, uni bosganda boshqa sahifaga o'tmaydi, faqatgina ishoratning oxirgi qismi o'zgaradi, yani «#»dan so'ng yorliq nomi qo'shiladi.

HTML kodini quyida keltirilgan kabi qisqacha ifodalash mumkin:

```

<a href="#showModal"> Modal oyna </a>
<div id="showModal" class="modalQatlam">
    <div>
        <a href="#yop" title="Yopish" class="yop">x</a>
        <h2> Foydalanuvchi bilan muloqot </h2>
        <p> Bu yerda tanlovlар va murojaatlar
            akslantirilishi mumkin</p>
    </div>
</div>

```

Endi har bir elementning qanday tarzda akslanishini CSS yordamida aniqlaymiz:

```

.modalQatlam {
    position: fixed; /* Boshqa qatlama koordinatasi ... */
    top: 0;           /* ... aniq ko'rsatilgan obyekt */
    right: 0;
    bottom: 0;        /* to'rt tarafga to'liq yoyilgan */
    left: 0;          /* sahifani to'la egallagan */
    z-index: 999;     /* Eng ustki (yuqori qiymatli) qatlama */
    background: rgba(222, 222, 222, 0.6); /* Orttasviri */
    opacity: 0;        /* Ko'rinnmaydigan darajada shaffof */
    transition: opacity 0.5s ease-in; /* Shaffoflik asta o'zgaradi */
    pointer-events: none; /* Sichqoncha bosishini bekor qiladi */
}
.modalQatlam:target { /* «Modal oyna» bosilgandagini ishlaydi */
    opacity: 1;        /* Ko'rindigan darajaga keltiradi */
    pointer-events: auto; /* Sichqoncha bosilishi hisobga olinadi */
}
.modalQatlam > div { /* Faqat modal qatlamdagи «div» uchun */
    width: 360px;      /* Murojaat oynasining eni */
    position: relative; /* «modalQatlam»ga nisbatan joylashgan */
    margin: calc(50% - 100px) auto; /* markazda joylashadi */
}

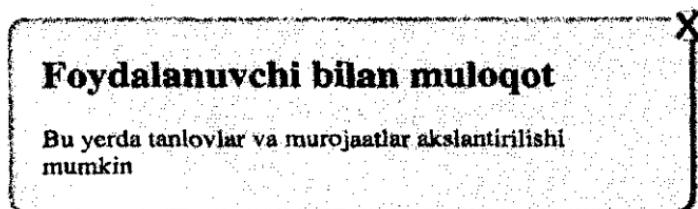
```

```

padding: 5px 20px; /* Ichki yozuvlarni chetidan uzoqlashtiradi */
border-radius: 10px; /* Oyna qirralarini yoysimon qiladi */
background: #dadada; /* Orttasviri kulrang */
box-shadow: 2px 2px 3px #000, inset 2px 2px 3px gray;
} /* Oyna bo'rtib chiqqanday ko'rinish uchun ikki xil soya qo'yilgan */
.yop { /* Oynani yopish uchun «x» shaklidagi nishoncha */
background: #dbdbdb; /* «x» nishonchasingin ortrangi */
font-family: Arial; /* «x» harfmas, yopish tasviriday chiqadi */
color: #666; /* «x»ning rangi */
line-height: 20px; /* balandligi, eniga mos bo'lishi kerak */
width: 20px; /* Nishoncha eni, yozuv hajmi inobatga olinadi */
font-size: 24px; /* «x» kattaroq ko'rinishi uchun */
position: absolute; /* joylashgan ornini aniq ko'rsatishga */
right: -8px; /* o'ngga chiqib turadi */
top: -4px; /* tepadan ham chiqib turadi */
text-align: center; /* «x» ortada joylashadi */
text-decoration: none; /* ostki chiziq akslantirilmaydi */
overflow: hidden; /* sig'magan «x»ning qismi ko'rinemaydi */
border-radius: 12px; /* doirasifat ko'rinishini ta'minlaydi */
box-shadow: 2px 0 3px #000; /* soyasi oynanikiga mutanosib */
} /* sichqonga borganda orttasviri o'zgarish uchun: */
.yop:hover { background: #eee; }

```

Ushbu kodlar o'z o'rniiga qo'yib bajarilsa, sahifada paydo bo'lgan «Modal oyna» yozushi bosilganda ekran markazida quyidagicha tasvir paydo bo'ladi:



Albatta, ta'bga qarab har kim o'zi istaganday shakl-shamoyil berishi mumkin. Oynaning balandligi oldindan aniq bo'limgani bois tik ravishda markazga chiqarish har doim ham muvaffaqiyatli amalga oshmaydi. Shuning uchun yuqoriga nisbatan aniq masofada joylash ma'qul topiladi, shunda **calc** funksiyasini ishlatischga zaruriyat qolmaydi.

10.4. Qisqa izoh

Ko'p hollarda berilgan matn akslantirilishi lozim bo'lgan maydonga sig'maydi. Qolaversa, maydonning o'zi ham turli monitorlar uchun turliche o'lchamda bo'ladi. Agar matn uchun bir qator joy ajratilgan bo'lsa, masalan, yangiliklar uchun mo'ljallangan saytlardagi mavzular ro'yxati singari, uning yo'li oson. Qisqa mavzular to'liqligicha ko'rinaraveradi, uzun sarlavhalar esa maydon eniga yetganda kesilib, to'liq akslanmaganini bildirish uchun uch nuqta qo'yiladi. Buning uchun CSS quyidagicha yoziladi:

```
.izoh {  
    width: 30%; /* eni qancha bo'lishi aniq chegaralanadi */  
    height: 20px; /* balandlik yozuv o'lchamiga mos bo'lishi lozim */  
    font-size: 16px; /* o'lcham balandlikka mos ko'rsatiladi */  
    color: darkblue; /* rangini ta'bga qarab ko'rsatish mumkin */  
    white-space: nowrap; /* yagona satrdan iborat bo'ladi */  
    overflow: hidden; /* sig'magan yozuv qismi ko'rinnmaydi */  
    text-overflow: ellipsis; /* yozuv kesilsa, 3 nuqta qo'shiladi */  
}
```

Bunda bir necha xususiyatlarning yagona mutanosib vazifa bajarishini inobatga olish lozim. Agar eni aniq ko'rsatilmasa, yozuv hajmiga qarab u o'zgargani bois, bejirimlik yo'qolib, kerakli joyda kesilmasligi mumkin. Odatda, sarlavha to'liq yozilib, unga beriladigan izohlar uchun bunday qisqartirishlar keng qo'llaniladi. Agar keyingi qatorga tushish taqiqlanmasa, qanday uzunlikdagi so'z boshqa satrga o'tgani aniq bo'limgaganligi bois, kesilgan qatorlar o'ng tarafdan tekis joylashmaydi. Yana bir muhim jihat - «overflow: hidden;» qiymati berilmasa, «text-overflow» o'z vazifasini bajarmaydi.

Tasavvur mukammalroq bo'lishi uchun aniq misol ko'rib o'taylik:

HTML

```
<h2> HTML </h2>
<p>Veb-sahifani jihozlovchi til</p>
```

CSS

```
<h2> CSS </h2>
<p>HTML elementlarini jilolaydigan qonuniyatlar majmuasi</p>
<h2> XML </h2>
<p>Ma'lumotlar qat'iy teglar bilan beriladi.</p>
<h2> SVG </h2>
<p>Maxsus teglar bilan tasvir chizishga yo'naltirilgan til</p>
```

```
h2 {
    width: 60px;
    height: 20px;
    padding: 2px 4px; /* ichki hoshiyalar, tepa-past va yonlari */
    text-align: center; /* matn satrning o'rtaida joylashadi */
    float: left; /* Chapga tekislanadi, o'ngda element chiqadi */
    background: #dedede;
    cursor: default; /* Ko'rsatkich shakli odatdagidek qoladi */
}
p {
    width: 380px;
    height: 20px;
    padding: 2px 4px;
```

CSS

```
font-size: 16px;      /* Husnixat o'lchami */
background: #eee;     /* Ortrangi */
color: darkblue;     /* Rangi */
white-space: nowrap; /* Faqat bitta qatorga yoziladi */
overflow: hidden;    /* Sig'magan qismi ko'rinxaydi */
text-overflow: ellipsis; /* Matn kesilsa, 3 nuqta qo'yiladi */
```

}

Natija

HTML Veb-sahifani jihozlovchi til.

CSS HTML elementlarini jilolaydigan qonuniya...

XML Ma'lumotlar qat'iy teglar bilan beriladi.

SVG Maxsus teglar bilan tasvir chizishga yo'

10.5. Harakatlanuvchi menyu

Veb-sayt yaratilar ekan, birinchi o'rinda undagi ma'lumotlar bo'lsa, ikkinchi o'rinda uning qanday jozibali namoyish etilishidir. Sahifadagi obyektlar vaqtga yoki foydalanuvchining tanloviga qarab o'zgarar ekan, ushbu jarayon o'ziga xos harakatlanish orqali amalga oshirilsa, mushtariyni o'ziga jalb etish uchun bu yaxshi omil sifatida xizmat qiladi. Harakatlantirish, albatta, maxsus hisob-kitoblar orqali bajariladi, buning uchun esa das-turiy ta'minot hamda qo'shimcha skriptlar talab etilishi turgan gap. Ammo CSSning standart harakatlantirish vositalari hamda HTMLdagi hodisalarini bir-biriga bog'lash orqali bir qancha animatsiyalarni yaratishimiz mumkin.

Yuqorida o'rgangan bilimlarimiz asosida ochilib-yopiluvchi izohli menu yaratamiz.

HTML kodini quyida keltirilganidek qisqacha ifodalash mumkin:

```
<aside id="chapTomon"><!--Sarlavhalar joylashgan maydon-->
  <details> <!-- Birinchi yashirin izohli sarlavha -->
    <summary>Mavzu 1</summary> <!-- 1-sarlavha nomi -->
    <p>
      Uch<br>qator<br>izoh <!-- Izoh aniq balandlikka ega -->
    </p>
  </details>
  <details> <!-- Ikkinchchi yashirin izohli sarlavha -->
    <summary>Mavzu 2</summary> <!-- 2-sarlavha nomi -->
    <p> Qandaydir izoh </p>
  </details>
</aside>
```

Endi har bir elementning qanday tarzda akslanishini CSS yordamida aniqlaymiz:

```
#chapTomon { /* izohli sarlavhalar chiqish maydoni */
  width: 185px; /*maydon eni, sarlavha enidan kattaroq */
  border: 1px solid #dedede; /*maydon chegarasi */
  border-radius: 6px; /*chevara qirralari yoysimon */
  float: left; /*chapda, o'ngidan obyektlar chiqishi mumkin */
}
details { /* izohli sarlavha qo'yish tegi */
  border-width: 1px; /*chevara qalinligi, aniq o'lchamda */
  background: #efefef; /*ortrangi, och kulrang */
  padding: 2px; /*atrofidagi ichki hoshiya */
  font-size: 18px; /*husnixat o'lchami */
  font-family: Arial; /*husnixat nomi */
```

```

line-height: 20px !important; /* qator balandligi, muhim!*/
min-height: 20px; /* eng kichik balandligi */
animation: yopil .3s ease-in; /* «yopil» 0.3 soniya bajariladi */
}
details[open] { /* izoh ochilgan holati */
max-height: 100px; /* ochilgandagi eng baland qiymati */
overflow: hidden; /* sig'magan matn ko'rinmay turadi */
animation: ochil .8s ease-out; /* «ochil» 0.8 soniya ishlaydi */
}
details summary { /* «details» ichidagi sarlavhalar */
background: #dedede; /* ortrang, ajrab turishi uchun to'qroq */
border-radius: 3px; /* tugma shakliga kelishi uchun */
box-shadow: inset 2px 2px 4px grey, 2px 2px 4px black;
padding: 2px 2px 2px 12px; /* sal o'ngroqdan boshlanadi */
font-weight: bold; /* sarlavha qalin yozuvda ko'rindi */
width: 160px; /* maydonning umumiyligi enidan kichik */
cursor: pointer; /* sichqoncha ko'rsatkichi (qo'l shaklida) */
}
details p { /* «details» ichidagi «p» teqlari, izoh yozuvi */
margin: 10px 24px; /* sarlavhaga nisbatan ichkarida */
color: #333; /* yozuv rangi (ajrab turishi uchun) */
}
@keyframes ochil { /* «ochil» harakatlanishini ta'minlash */
from { height: 20px; } /* boshlang'ich balandlik */
to { height: 100px; } /* yakuniy, eng yuqori balandligi */
}
@keyframes yopil { /* «yopil» harakatlanishini ta'minlash */
from { height: 100px; } /* boshlang'ich yuqori balandlik */
to { height: 20px; } /* yakuniy balandligi sarlavhanikicha */
}

```

Natijada quyidagicha harakatlanadigan mavzular ro'y-xati namoyon bo'ladi:

 **Mavzu 1**

Uch
qator
izoh

 **Mavzu 2**

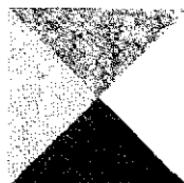
10.6. Qalqib chiquvchi oyna

HTMLda har bir obyekt yaratuvchi tegning «title» nomli alomati mavjud, unga berilgan matn sichqonga o'sha obyekt ustiga borganda izoh sifatida ko'rinishi. Ushbu foydali qisqa yozuvni istalgancha jilolash, ko'rinish vaqtini va joylashish o'rnini o'zgartirish, afsuski, mumkin emas. Uni brauzerning o'zi aniqlaydi. Ammo uning o'rnini bosa oladigan qalqib chiquvchi oynani o'zimiz yaratishimiz mumkin.

Avval bir qiziqarli va juda foydali CSSning imkoniyatini tahlil qilib o'taylik. Har bir obyekt yaratilishidan oldin va keyin uning tanasiga biriktirilgan holda paydo bo'lувчи, alohida-alohida ikkita qo'shimcha element shakllantirish mumkin. Yuqorida ushbu imkoniyatdan bo'lim nomlarining oldiga raqam qo'yishda foydalangan edik. [9.9.] Umuman olganda, blokli teglar uchun to'g'ri to'rtburchak shaklidagi alohida maydon hosil qilish imkoniyati mavjud. Har bir blokli teg to'g'ri to'rtburchak shaklini egallanishni inobatga olib uning perimetridan uchburchak yasash uchun foydalanamiz. Buning

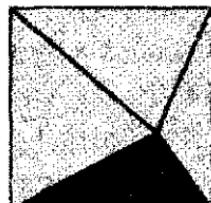
uchun chegaraviy chiziqni qalinqoq qilamiz va o'lchamini nolga tenglashtiramiz. Bunda chegara o'lchamiga ikki barobar kattalikdagi to'g'ri to'rtburchak hosil bo'ladi. Bilamizki, to'rt tomondagi chegaraviy chiziqlarni to'rt xil tusda chizish mumkin, bunda to'rtta uchlari bir nuqtada birlashgan uchburchak hosil bo'ladi:

```
div {  
    width: 0; height: 0; /* o'lchami nolga teng */  
    border: 50px solid; /* chegara qalinligi */  
        /* tomonlarining rangi turli xilda */  
    border-top-color: #A2A2A2;  
    border-right-color: #E0E0E0;  
    border-bottom-color: #6C6C6C;  
    border-left-color: #BFBFBF;  
}
```

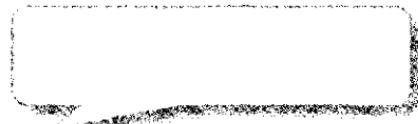


Albatta, har bir tomon uchun qalinlikni ham turli-cha berish mumkin, bu bilan istalgan shakldagi uchburchak hosil qilish mumkin. Qolgan uchtasining rangi shaffof qilib qo'yilsa, kerakli yagona uchburchak ko'rinarli bo'ladi. Uning «atrof xususiyatlari» [9.1.] yordamida istalgan nuqtaga ko'chirish mumkin. Avval kerakli uchburchak hosil qiluvchi kod yozib keltiraylik:

```
#uchburchak {  
    width: 0;  
    height: 0;  
    border-top: 80px solid #aaa;  
    border-left: 60px solid #ccc;  
    border-right: 30px solid #ddd;  
    border-bottom: 40px solid #333;  
}
```



Qalqib chiquvchi oynadagi matn qaysi obyekt uchunligini ko'rsatish maqsadida «mil» qo'yib qo'yish an'anaga kirgan. Blokli teglarimiz to'g'ri to'rtburchak shaklidagi obyekt hosil qilar ekan, uning kerakli joyiga uchburchakni shunday joylashtirish lozimki, obyektga ko'rsatuvchi «mil»li xabar beruvchi matnli obyekt tasavvurini bersin:



Albatta, uning soyasi va chegaraviy chizig'iga e'tibor berish zarur. Ammo biz hosil qilgan uchburchak, aslida, chegaraning o'zi, uning ichki rangi mavjud emas. O'lchami yo'q obyektning soyasi ham bo'lmaydi. Shuning uchun «mil»ni soya tushmaydigan tarafga qo'yish ma'qul. Ko'rsatuvchi «mil»ning va uning chegarasining rangini obyektnikiga o'xshatish uchun ikkita uchburchak chizamiz. Birinchisi sal kattaroq, chegara rangida bo'ladi. Ikkinchisi uning ustiga obyekt ortrangida chiziladi. Shunda tagida qolib ketgan katta uchburchak ustidagisining chegaraviy chizig'i sifatida ko'rindi. Faqat avval chegara rangidagi kattaroq uchburchakni «:before» tabaqasida, so'ng ichki uchburchakni «:after» tabaqasida qoidalarini yozamiz va joylashish nuqtasini mos ravishda ko'rsatamiz.

Aytaylik, bizning qalqib chiquvchi oynamiz «**izoh**» sinfiga taalluqli «**p**» tegining obyektlari uchun moljalangan bo'lsin. Hozircha faqat CSS imkoniyatlaridan kelib chiqib «oyna»ni qo'zg'almas joyda yashirish yoki ko'rinarli qilish bilan shug'ullanamiz. Keyingi JavaScriptga

bag'ishlangan kitobda «oyna»ni sichqoncha harakatiga qarab asta ko'chirish, mili va matnini o'zgartirish, qanday amalga oshirilishi bilan batafsil tanishamiz. Avvalo, CSS qoidalari asosida mili yuqoriga qaragan va soyasi pastga tushgan qalqib chiquvchi oynani ko'rinmas holda yaratamiz:

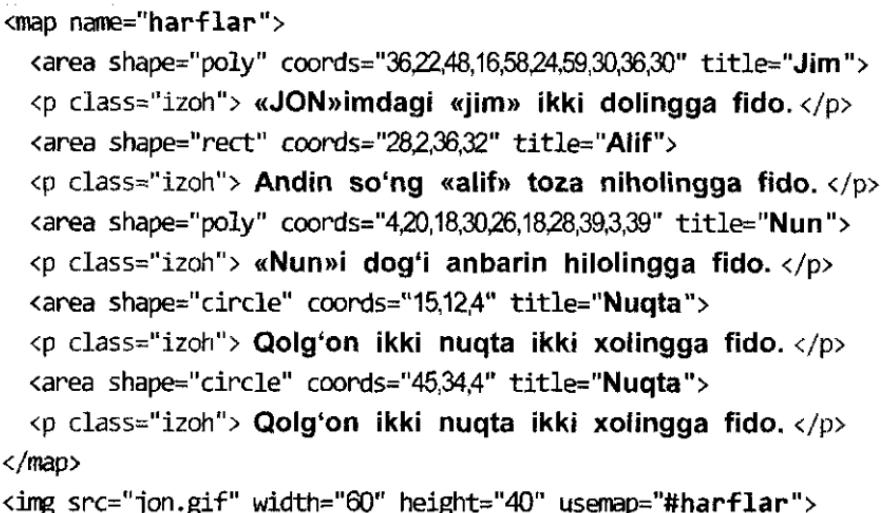
```
p.izoh { /* qalquvchi oyna faqat <p class="izoh">lar uchun */
    position: absolute; /* yuqori qatlama joylashadi */
    visibility: hidden; /* dastlab ko'rinxmaydigan bo'ladi */
    background-color: #eee; /* ortrangi */
    border: 3px solid gray; /* chegaraviy chiziqlar qalin va to'q */
    border-radius: 5px; /* qirralari yoysimon */
    padding: 4px 8px; /* matn chegaraviy chiziqdan ichkariroqda */
    font-weight: bold; /* qalin yozuv */
    top: 50px; /* yuqoriga nisbatan 50 piksel pastda */
    left: 18px; /* chapga nisbatan 18 piskel o'ngroqda */
    box-shadow: -3px 3px 5px #000;
} /* soya chap pastki tarafda, 5 pikselgacha enlaydi */
p.izoh::before { /* <p>dan oldin chiziladi */
    content: '';
    position: absolute; /* joylashuv o'rnini o'zgartirish uchun */
    top: -15px; /* <p>ga nisbatan 15 piksel yuqorida */
    left: 11px; /* <p>ning boshlang'ich nuqtasidan ichkariroqda */
    width: 0; height: 0; /* o'zining o'lchami yo'q */
    border-top: transparent; /* tepadagi uchburchak ko'rinxmas */
    border-right: 15px solid transparent; /* o'ngdagagi uchburchak */
    border-left: 15px solid transparent; /* chapdagagi uchburchak */
    border-bottom: 15px solid gray; /* pastki uchburchak ko'rinxadi */
}
p.izoh::after { /* keyingi uchburchak ham yuqoridagiday */
    content: '';
}
```

```
position: absolute;  
top: -10px;  
left: 14px;  
margin-left: 2px;  
width: 0;  
height: 0;  
background-color: transparent;  
border-top: transparent;  
border-right: 10px solid transparent; /* ko'rinnmas */  
border-left: 10px solid transparent; /* ko'rinnmas */  
border-bottom: 10px solid #eee;  
}
```

Qalqib chiquvchi oyna bilan brauzerning o'zidagi «title»ga berilgan izohni ko'rsatishini keskin farqlash uchun har ikkalasi ham ko'rinnadigan misol tuzaylik. Xaritali rasm qo'yish [5.1.] mavzusida rasm uchun maydon chizib, sichqoncha orqali rasm ustida ko'rsatkichni yurgizganda mos izoh chiqishiga doir misol ko'rgan edik. Endi kichik izohchaga mos ravishda u bilan birga kengaytirilgan matnni o'z ichiga olgan qalqib chiquvchi oyna ham ko'rinsin.

Alisher Navoiyning bir ajoyib to'rtligida «jon» so'zidagi harf va belgilarni baxshida etishi haqida yozilgan. Agar arab alifbosi asosidagi eski o'zbek yozuvni tasavvur qilinmasa, she'rda nima nazarda tutilganligini tushunish mushkul. Shuning uchun arab harflari yordamida «jon» degan so'zni yozamiz va uni rasm sifatida saqlab olamiz. O'lchami 60x40 piskel bo'lgan tasvir hosil bo'ldi. Endi «map» tegi yordamida undagi har bir harf va nuqtalar qayerda joylashganini belgilab chiqamiz. Har

bir maydonni ifodalovchi «area» tegining «title»iga mos belgi nomini beramiz va undan keyin to'rtlikning tegishli qatorini «p class="izoh"» tegi ichida keltirib o'tamiz. O'zimizga tushunarli bo'lgan quyidagicha kod yoziladi:

```
<map name="harflar">
  <area shape="poly" coords="36,22,48,16,58,24,59,30,36,30" title="Jim">
    <p class="izoh"> «JON»imdagi «jim» ikki dolingga fido. </p>
  <area shape="rect" coords="28,2,36,32" title="Alif">
    <p class="izoh"> Andin so'ng «alif» toza niholingga fido. </p>
  <area shape="poly" coords="4,20,18,30,26,18,28,39,3,39" title="Nun">
    <p class="izoh"> «Nun»i dog'i anbarin hilolingga fido. </p>
  <area shape="circle" coords="15,12,4" title="Nuqta">
    <p class="izoh"> Qolg'on ikki nuqta ikki xolingga fido. </p>
  <area shape="circle" coords="45,34,4" title="Nuqta">
    <p class="izoh"> Qolg'on ikki nuqta ikki xolingga fido. </p>
</map>

```

«Jon» degan so'zdan tashkil topgan rasmdagi «jim», «alif», «nun» harflari va yana ikkita nuqtani belgilab, ularga mos Navoiy misralarini kerakli tartibda joylash-tirib chiqildi.

Endi siqchonchaning ko'rsatkichi qaysi «area» ustiga borgan bo'lsa, undan keyin joylashgan «p class="izoh"» elementi ko'rinishidan bo'lishini ta'minlashimiz kerak. Buning uchun «qo'shni teg»ga [8.2.] murojaat qilish sintaksisidan foydalanib CSS faylga quyidagilarni qo'-shimcha qilish kifoya:

```
area[title="Jim"]:hover + p { visibility: visible; }
/* «Jim»dan keyingi <p> obyekti ko'rinadi */
area[title="Alif"]:hover + p { visibility: visible; }
/* «Alif»dan keyingi <p> obyekti ko'rinadi */
area[title="Nun"]:hover + p { visibility: visible; }
/* «Nun»dan keyingi <p> obyekti ko'rinadi */
area[title="Nuqta"]:hover + p { visibility: visible; }
/* «Nuqta»lardan keyingi <p> obyekti ko'rinadi */
```

Tasavvur to'liq bo'lishi uchun va ushbu mavzuda bayon etilgan imkoniyatlarni to'g'ri tushunib yetish maqsadida keltirilgan misollarni amalda sinab, ishlatib ko'rish tavsiya etiladi. Yagona hol uchun sichqoncha rasmdagi «alif» harfi ustiga borganda quyichagicha ko'rinish namoyon bo'ladi:



Alif

Andin so'ng «alif» toza niholingga fido.

10.7. Yig'iladigan maydon

Veb-saytning bosh sahifasida imkoni boricha barcha yo'nalishini qamrab oladigan tarzda ko'p ma'lumot keltirilishi kerak. Lekin inson e'tibori hamma narsaga teng tushmasligini inobatga olib ortiqcha ma'lumotlarni

ko'rsatishdan tiyilish maqsadga muvofiqdir. Masalan, saytlarda, odatda, qo'yiladigan «biz haqimizda» yoki «biz bilan aloqa» kabi bo'limchalarda yoziladigan manzillarning ko'pchilikka ahamiyati yo'q. Lekin murojaat qilmoqchilar bo'lsa, albatta, ular uchun ilk sahifadayoq imkoniyat yaratuvchi oyna qo'yilishi lozim. Bu kabi axborotlar yig'iladigan oyna yoki maydon ichida keltirilsa, qiziqqan foydalanuvchi nomi bo'yicha o'ziga kerakli bo'lganini ochib, undagi ma'lumotni ko'rish imkoniyati yaratiladi. Ko'p hollarda tajribasiz dasturlovchilar yuqori yoki yon tarafga joylashgan menyu ichida shu kabi axborotlarni ko'rsatishadi, bu bilan menyuning umumiyligi qoidasi buziladi, aslida, u yerlarda (alohida sahifada) boshqa ma'lumotlarni ko'rsatuvchi ishoratlardan tashkil topgan tugmalar ro'yxati bo'lishi lozim.

Hozircha biz faqatgina HTML va CSS imkoniyatlariidan kelib chiqib oyna tasavvurini beradigan yoyiladigan maydoncha yaratishni ko'rib chiqamiz. U «checkbox» tanlanganda, ya'ni darchaga belgi qo'yilganda sekin yoyiluvchi, va aksincha, sekin yig'iluvchi «div» obyektidan iborat bo'ladi. (*Keyingi kitobda JavaScript yordamida bir qancha hodisalarga bog'langan holda turlichalashchlarni ham ko'rib o'tamiz.*)

Oynaning tepe qismida grafikli operatsion tizimlarda ko'nikkanimiz kabi formaning nomi va nishonchalari joylashadi. Shu bilan birga, haqqoniy ko'rinishi uchun orqa rangi tuslanuvchan hamda chegaraviy qirralari yoysimon tarzda bo'ladi. Uning uchun CSS quyidagicha yoziladi:

```

.maydon {      /* umumiy tashqi maydon */
    width: 300px; /* oynaning eni, bo'yli tarkibiga qarab o'zgaradi */
    margin: 4px; /* qo'shni obyektlardan ajrab turishi uchun */
    position: relative; /* ichki joylashuvlarni o'ziga nisbatan oladi */
}

#bosh {        /* oynaning tepe sarlavha qismi */
    height: 24px; /* bo'yli bir qator, nishoncha o'lchamiga mos */
    cursor: pointer; /* ko'rchatkich ko'rinishi o'zgartirilgan */
    background-image: linear-gradient(to top, #666, #eee);
    border: 1px solid grey; /* chegara qalinligi */
    border-top-left-radius: 12px; /* tepasida qirralar yoysimon*/
    border-top-right-radius: 12px;
    box-shadow: 3px 3px 5px #333; /* soya o'ng-pastga yo'nalgan */
}

#tanla {       /* tanlash darchasi */
    position: absolute; /* joyini maydonga nisbatan oladi */
    top: 3px; /* «bosh» qismi ichida chiqish uchun */
    left: 8px;
}

#mavzu {       /* Sarlavha */
    position: absolute; /* ko'rsatilgan o'rinda chiqishi uchun */
    float: left;
    top: -12px; /* joylashuv o'rirlari: tepaga kirgan */
    left: 30px; /* joylashuv o'rirlari: chapga surilgan */
    font-weight: bolder; /* qalin yozuv */
    user-select: none; /* sichqoncha bilan matn belgilanmaydi */
    -moz-user-select: none;
    -ms-user-select: none;
    -webkit-user-select: none;
}

#tanla:checked + div { /* belgilanuvchi darchadan keyingi oyna */
    height: 200px; /* boshlang'ich balandligi */
}

```

```

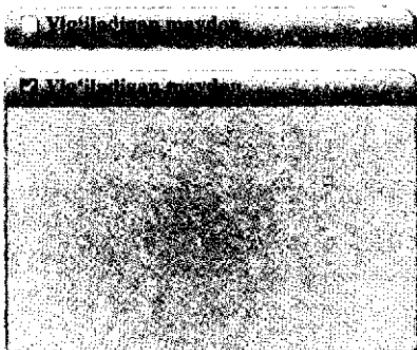
animation: ochil 0.7s ease-out;
} /* belgilanganda 700 millisoniya davomida yoyiladi */
#tanla:not(:checked) + div {
height: 0;
animation: yopil .7s ease-out;
} /* belgilash bekor qilinganda 0,7 soniyada yig'iladi */
#oyna { /* yig'ilib-yoyiluvchi oyna */
margin-top: -3px;
border: 1px solid grey;
height: 200px;
background-image: repeating-radial-gradient(
circle at center, #999, #eee 300px
);
box-shadow: 3px 3px 5px #333, inset 1px 2px 1px #333;
border-bottom-left-radius: 6px;
border-bottom-right-radius: 6px;
overflow: hidden;
} /* oyna kichrayayotganda sig'magan matn ko'rinxaydi */
@keyframes ochil { /*«ochil» nomli harakatlanishni ta'minlash*/
from { height: 0; } /* boshlang'ich vaziyatdagi balandlik */
to { height: 200px; } /* yakuniy (eng baland) vaziyat */
}
@keyframes yopil { /*«yopil» harakatlanishini ta'minlash */
from { height: 200px; } /* boshlang'ich (eng baland) vaziyat */
to { height: 0; } /* yakuniy vaziyat, balandligi mavjud emas */
}

```

Agar «.maydon»ga «position: relative;» qiymati berilmasa, uning ichida joylashgan elementlar uchun «position: absolute;» xususiyati sahifaga nisbatan hisoblanar edi. Maydon sahifaning qayerida joylashishi qat'iy aniqlanmagan vaziyatlarda, ayniqsa, ko'chuvchan

obyektlar yaratishda ushbu xususiyatga ahamiyat berish shart. Yana bir muhim jihat - qo'shni teglarni mutanosib joylashtirish va soyalarini umumlashtirish. HTML kodi CSSda yozilgandan ancha ixham bo'ladi:

```
<div class="maydon">
  <div id="bosh"></div>
  <p id="mavzu">
    Yig'iladigan maydon
  </p>
  <input id="tanla"
    type="checkbox">
  <div id="oyna"></div>
</div>
```



Bir marta yozilgan CSS qoidalari asosida bir qancha HTML obyektlari yaratilaveradi. Faqat unda qoidalarni «id»ga bog'lash tavsiya etilmaydi.

10.8. Mo'jaz hisobotlar

Dasturning eng muhim qismi u chiqarib beradigan hisobotlardadir. Birinchi o'rinda chiqayotgan ma'lumotlar to'g'riliги inobatga olinsa, ikkinchi o'rinda qanday ko'rinishda chiqayotganiga ahamiyat beriladi. Shunday ekan, hisobot ko'rinishiga alohida e'tibor qaratishimizga to'g'ri keladi.

Hozirgi zamonaviy texnologiya yuksalayotgan paytda foydalanuvchilar kompyuter va laptoplardan ham ko'ra

smartfon va planshetlarni ishlatishni ma'qul ko'rishi moqda. Sababi, ular ixcham, quvvati ham ko'proqqa yetadi, ortiqcha murakkab dasturiy ta'minotlardan xoli. Shunga mos ravishda taqdim etiladigan hisobotlarning ko'rinishi ham kichik va bejirim bo'lishi talab qilinmoqda. Yuqoridagilarni inobatga olib aynan uyali qurilma ko'rinishidagi quyidagi rasmda ko'rsatilgandek hisobot tayyorlash kodi bilan tanishib chiqamiz.

2-SAHIJA			
1. Osh			
1.	Olma	2,4	7 000
2.	Nok	1,2	8 200
3.	O'rik	3,6	9 300
4.	Behi	0,8	9 800
5.	Anor	1,3	19 300
6.	Shaftoli	1,8	2 600
7.	Gilos	2,7	3 700
2. Andozalar			
1.	Pomildori	3,6	9 300
2.	Bodring	2,9	7 100
3.	Baqlajon	1,8	5 800

Bitta sahifada ko'p ma'lumot ko'rsatish tavsiya etilmaydi. Eng kerakli axborotlarni jamlab, ularni toifasi bo'yicha bir necha sahifaga bo'lib chiqarish ma'qul. Foydalanuvchi barchasini ko'zdan kechirib, lozim topsa chop etib olaveradi.

Masala mukammalroq bo'lishi uchun uni sal murakkablashtirib, yuqoridagi ko'rinishdagi jadvalga yana bir ustun qo'shaylik. Unda har bir nomga mos izoh yozilgan bo'lib, faqat uyali qurilmalar uchun ko'rinsmasin. Keng maydonli monitorlarda, xususan, kompyuter ekranida yoki chop etilayotganda ko'rish mumkin bo'lsin.

Jadval yaratishdan oldin uning nechta ustun va satrdan iborat ekanligini aniqlab olish muhim. Qo'shma kataklar nechta yakkasining joyini egallashini ko'rsatish uchun oldindan jadval tuzilishi haqida tasavvur hosil qilib olish kerak. Rasmdan kelib chiqadigan bo'lsa, jadval to'qqizta ustun va yigirmata satrdan iborat. Bunda yashirin izohli ustun va bo'sh joylar ham sanalgan.

Jadval murakkab obyekt bo'lib, uning tashqi chegarasi kataklarning chegaralaridan alohida chiziladi. Shuning uchun bitta umumiyoq soya qo'yib bo'lmaydi. Chunki jadval to'g'ri to'rtburchak shaklini egallaydi, yuqoridagi rasmda esa qirralari yoysimon tarzda.

Avval har bir katakka o'ziga xos nomdag'i sinf berib HTMLda jadvalni tuzib olamiz:

```
<table align="center"> <!-- jadval o'rtada chiqadi -->
  <tr> <!-- yuqori hoshiya, bosh harflar bilan yozilgan -->
    <td class="top upper" colspan="9" align="center">
      2-sahifa
    </td>
  </tr>           <!-- bo'sh qator va ustunlarni ko'rsatadi -->
```

```

<tr class="bosh">      <!-- Chap chegara va ustunlar -->
    <td class="left" rowspan="18"></td>
    <td class="yoni" rowspan="18"></td>
    <td colspan="5"></td><!-- Ma'lumot chiquvchi 5 ustun -->
    <td class="yoni" rowspan="18"></td>
    <td class="right" rowspan="18"></td>
</tr>                  <!-- O'ng ustunlar va chegara -->
<tr>
    <th class="title tl">№</th>
    <th class="title">Nomi</th>
    <th class="title hideField">
        Qo'shimcha ma'lumot   <-- Yashirin katak -->
    </th>
    <th class="title">
        Vazni <p class="note">(kg)</p>
    </th>
    <th class="title tr">
        Narhi <p class="note">(so'm)</p>
    </th>
</tr>
<tr>          <!-- Kichik sarlavha -->
    <th class="value" colspan="5">Mevalar</th>
</tr>
<tr>
    <td class="value" align="center"> 1. </td>
    <td class="value"> Olma </td>
    <td class="value hideField"> Olma izohi </td>
    <td class="value" align="center"> 2,4 </td>
    <td class="value" align="right"> 7 000 </td>
</tr>
<tr>
    <td class="value" align="center"> 2. </td>
    <td class="value"> Nok </td>
    <td class="value hideField"> Nok izohi </td>

```

1,2	8 200
3.	O'rik izohi
3,6	9 300
4.	Behi izohi
0,8	9 800
5.	Anor izohi
1,3	19 300
6.	Shaftoli izohi
1,8	2 600
7.	Gilos

Gilos izohi	2,7	3 700
Jami:		
&ampnbsp		
13,8	59 900	
Sabzavotlar		
1.		
Pomidor		
Pomidor izohi		
3,6	9 300	
2.		
Bodring		
Bodring izohi		
2,9	7 100	
3.		
Baqlajon		
Baqlajon izohi		
1,8		

```

<td class="value" align="right"> 5 800 </td>
</tr>
<tr> <!-- Kichik sarlavha td katagida berilgan -->
<td class="title" align="right" colspan="2">
Jami:
</td>
<td class="title hideField"> &ampnbsp </td>
<td class="title" align="center"> 8,3 </td>
<td class="title" align="right"> 22 200 </td>
</tr>
<tr> <!-- Katta sarlavha th katagida berilgan -->
<th align="right" class="title bl upper" colspan="2">
Barchasi:
</th>
<th class="title hideField"> &ampnbsp </th>
<th align="center" class="title"> 22,1 </th>
<th align="right" class="title br"> 82 100 </th>
</tr>
<tr class="bosh">
<td colspan="4"></td>
</tr>
<tr> <!-- Pastki hoshiya -->
<td class="bottom" colspan="9" align="center">
<div class="tugma"> <!-- Ellips ichidagi millar -->
<a class="page" href="oldingi.html"
   title="Oldingi sahifa"> &lt; </a>
<a class="page" href="keyingi.html"
   title="Keyingi sahifa"> &gt; </a>
</div>
</td>
</tr>
</table>
```

Endi har bir «class»da ko'rsatilgan nomlar uchun CSSda qoidalar majmuasini yaratamiz. Yuqoridagi rasm-

da faqat tuslanuvchan qora (monoxrom) rang tanlangan. Agar tashqi chegara o'zgarmas aniq bitta ortranga ega bo'lganda, tasvir bu qadar haqqoniy ko'rinnagan bo'lardi. Ranglarni istalgancha o'zgartirish imkonini mayjud bo'lishi uchun SCSS tarzda [10.2.] qonuniyatlarini yozamiz. Umumlashtirilgan holda chegaraviy chiziq ko'rinishi va qirrasi uchun yoy radiuslarini ham o'zgaruvchilar orqali beramiz:

```
$black: #000; /* Qora rang */
$white: #fff; /* Oq rang */
$dark: $black + #333; /* Qoramtilrang */
$light: $white - #666; /* Oqish rang */
$line: 1px solid #aaa; /* Kataklar chegaralari */
$weight: 8px; /* Matndan chegaragacha yon masofa */
$border: 2px solid black; /* Tashqi chegaraviy chiziq */
$fontSize: 14px; /* Matn husnixaining olchami */
$radius: 40px; /* Tashqi chegara qirrasining yoy radiusi */

@media handheld /* faqat smartfonlarda chiqmasligi uchun */
.hideField {
    visibility: hidden; /* ko'rinnmasligi uchun */
    position: absolute; /* joy olmasligi uchun */
}
table {
    border-spacing: 0; /* kataklar orasidagi masofa */
    border-collapse: separate; /* har bir katakka chegara */
    margin: 10px; /* qoshni obyektlardan ajrab turishi uchun */
}
.top { /* Tepa tashqi hoshiya */
    border: $border;
    border-bottom: none; /* Faqat pastki chegara ko'rinnmaydi */
    border-top-left-radius: $radius;
    border-top-right-radius: $radius;
```

```

background-image: linear-gradient(
    to top, $dark, $light 65%
); /* tuslanuvcham qora */
font-size: $fontSize - 2; /* tepadagi matn sal kichikroq */
padding: $fontSize + 2; /* katak balandroq */
color: $white; /* ortrangi to'qligi uchun oq rangda chiqadi */
box-shadow: 8px 0 10px $light; /* soya faqat o'ngdan */
}

.bottom { /* Pastki tashqi hoshiya */
border: $border;
border-top: none; /* Faqat tepe chegara ko'rinnmaydi */
border-bottom-left-radius: $radius;
border-bottom-right-radius: $radius;
background-image: linear-gradient( /* tuslanishi */
    to bottom, $dark, $light 75%
); /* Pastki chegara bo'yicha soya ko'rinnadi */
box-shadow: 6px 3px 10px 3px $light;
}

th.title { /* Katta sarlavha */
border: $line;
padding: 2px $weight;
font-size: $fontSize;
color: $black;
font-weight: bolder;
line-height: $fontSize + $weight; /* Satr balandligi */
vertical-align: middle; /* Matn o'rtaga chiqadi */
background-color: $white - #666; /* Ortrangi to'q */
}

th.value { /* Kichik sarlavha */
font-size: $fontSize + 1;
padding: 4px $weight;
font-weight: bolder;
border: $line;
color: $dark; /* matn rangi to'q */
background-color: $white - #222; /* Ortrangi to'qroq */
}

```

```

td.title { /* Jami yoziladigan kichik sarlavha */
    @extend th.value; /* Kichik sarlavha jilolari bilan bir xil */
    border-bottom-width: 5px;
} /* pastgi qismida qalin chegara chiziladi */
td.value { /* Qiymat chiqadigan katak */
    font-size: $fontSize;
    padding: 2px $weight; /* Yozuv chegaradan ichkariroqda */
    border: $line;
    background-color: $white - #010101; /* oqsifat ortrang */
} /* Ichki jadval qirralarining yoyi, 4 marta kichik */
.tl { border-top-left-radius: $radius / 4; }
.tr { border-top-right-radius: $radius / 4; }
.bl { border-bottom-left-radius: $radius / 4; }
.br { border-bottom-right-radius: $radius / 4; }

.left { /* Chap tashqi chegara */
    width: 4px;
    border-left: $border; /* o'rtasi och, cheti to'q tuslanuvchan */
    background: radial-gradient(
        circle farthest-corner at center, $light, $dark
    );
}
.right { /* O'ng tashqi chegara */
    @extend .left; /* Chap tashqi chegara singari */
    border-left: none; /* chap chegaraviy chiziq ko'rinxaydi */
    border-right: $border; /* o'rniga o'ng tomoni ko'rinxadi */
    box-shadow: 7px 2px 10px 1px $light;
} /* o'ng tomonidan soya chiqadi */
.bosh { /* Ichki jadvaldan tepa va pastidagi masofa */
    height: $weight;
}
.yoni { /* yon chegaragacha masofa, 2 marta katta */
    width: $weight * 2;
}
/* Matn bosh harflar bilan yoziladi: */

```

```

.upper { text-transform: uppercase; }
.tugma { /* Ellips shaklidagi tugma */
    border: 1px solid $black;
    border-radius: 16px;
    width: 32px;
    line-height: $fontSize * 2;
    margin: $weight;
    text-align: center;
    box-shadow: inset 3px 3px 5px $light;
} /* Ajrab turishi uchun soya ichkariga tushgan */
.page { /* Ellips ichidagi millar shakli */
    font-weight: bold;
    font-family: Consolas;
    text-decoration: none; /* Ishoratda tagchiziq yo'q */
    color: $black;
}
.note { /* Sarlavhadagi qo'shimcha so'zlar */
    font-style: italic; /* Egri yozilgan */
    color: $dark;
    font-size: $fontSize - 2; /* O'lchami kichikroq */
    line-height: $fontSize - 6; /* Balandlik qisqa */
    margin: 0; /* Tashqi hoshiyasi hajm olmasligi uchun */
}

```

10.9. Forma darchalarining nuqsonlari

Dastavval matnlarni turli jiloda rasmlar bilan namoyish etishga mo'ljallangan brauzer internet taraqqiyoti va ommalashishi natijasida o'ziga ulkan vazifalarni ola boshladи. Endilikda veb-sahifalar server bilan muloqot, ma'lumot almashish vositasi sifatida keng xizmat ko'r-

satmoqda. Ma'lumotlarni yuborish va ularni qayta ishlash uchun HTML-dagi forma darchalari biz zamonaviy texnologiyalarda ishlab ko'nikkanimizday imkoniyatlarni to'laqonli bera olmaydi. Shuning uchun ham veb-interfeys (oraz) yasashda turli qo'shimcha kutubxonalar yoki dasturiy muhitlardan foydalanishga to'g'ri keladi. Ba'zi hollarda bunga sabab HTML talablariga brauzerlarning javob bermasligi (masalan **FireFox** va **Safarida** vaqt bilan ishlash uskunalari yo'q [7.3.]) bo'lsa, ba'zan imkoniyatlarining o'zi ham mukammal emasligi muammo tug'diradi. Masalan, darchaga «readonly» alomati ko'r-satilgan bo'lsa, undagi ma'lumotni o'zgartirish foydalanuvchi uchun imkonsiz. Ammo darcha tayinlanuvchi (select) bo'lsa, undagi ro'yxatdan boshqasini tanlab, qiymatini o'zgartirishga «readonly» qarshilik qila olmaydi. Buning uchun «disabled» alomati ishlatiladigan bo'lsa, obyektning ko'rinishi o'zgarib, xiralashib qoladi.

Bu tayinlanuvchi darchaning yagona kamchiligi emas. Agar unda bo'sh qiymatli ro'yxat bo'lmasa, tanlashni butkul bekor qilish uchun qo'shimcha skript ishlatishga majbur bo'linadi. Eng katta kamchiligi ro'yxatning ayni bir elementiga «option» tegida o'zgacha jilo berish mumkin, ammo u tanlanganda yozuvi ko'ri-nadi, jilosi esa yo'qoladi. Amaliy tajribadan ayon bo'l-diki, bundan-da dahshatli muammosi mavjud ekan. Ro'yxati keltirilgan har bir element uchun o'zining «option» tegida CSS yordamida alohida rasm qo'yilganda va tanlangan element qiymatiga mos ravishda «select»ning o'zining ham rasmi o'zgaradigan qilinsa, ushbu ajoyib tasvirli va matnli tayinlanadigan darcha

faqat FireFoxda ko'ngildagidek ishlar ekan, eng omma-bop Chrome ham shu oddiy narsani to'g'ri ko'rsatib bera olmas ekan. Albatta, HTML qoidalarida «select»-ning «option» elementlarida rasm qo'yilishi ham qo'llanilishi shart deyilmagan, ammo umumiy CSS imkoniyatlari bo'yicha bunday imkoniyat mavjud bo'lishi ko'zda tutilishi lozim edi.

Bu kabi nuqsonlarni ko'plab keltirish mumkin, shuning uchun ham veb-interfeys yaratuvchisiga mukammalroq uskunalarga ega dasturiy muhitlarni qo'llash tavsija etiladi. Lekin o'rganishni ulardan emas, aynan ushbu kitobda yoritilganidek HTML va CSSning o'zagi-dan boshlash zarur. Sababi, unday muhitlar o'z qoidalarini o'rnatib olib, o'shanday ishlashni taqozo etadi. Aslida, esa u qoidalar haqiqiy HTML yoki CSSda mavjud bo'lmasligi, o'zgacha maqsadda qo'llanilishi yoki butunlay boshqa ma'noni anglatishi ham mumkin. O'sha muhitda mavjud bo'limgan yoki qo'llash ko'zda tutilmagan narsalarni amalga oshirishda, albatta, tub HTML, CSS va JavaScript imkoniyatlarini ishlatishga majbur bo'linadi.

Umuman olganda, ushbu kitobda yoritilgan bilimlar hamda so'nggi bo'limidagi keltirilgan misollar yordamida HTML va CSSlar orqali veb-sahifada ixtiyoriy ko'rinishdagi obyektlarni shakllantirish mumkin. Bu bilan to'laqonli dizaynerlik faoliyatini ham yuritsa bo'ladi. Faqat ozgina musavvirlilik salohiyati ham mavjud bo'lsa, foydadan holi emas. Kelgusi kitoblarimizda dasturlovchi maqomiga ega bo'lish uchun bilimlar yoritiladi.

Teglar ko'rsatkichi

!--	15	dfn	48, 183
!DOCTYPE	17	dialog	247
a	28, 61, 182	dir	53
abbr	52, 182	div	27, 145, 186
acronym	52	dl	59, 186
address	186	dt	59, 186
applet	52	em	48, 183
area	76	embed	89
article	29, 186	fieldset	137
aside	28, 186	figcaption	90
audio	80, 186	figure	90, 186
b	48, 182	font	49, 55
base	22, 26	footer	29, 186
basefont	49, 55	form	118, 186
bdi	51, 182	frame	53
bdo	182	frameset	53
big	48, 55	h1	28, 186
blink	55	h2	28, 186
blockquote	186	h3	28, 186
body	20, 26	h4	28, 186
br	12, 47, 182	h5	28, 186
button	128, 186	h6	28, 186
canvas	93, 186	head	19, 21
caption	67	header	28, 186
center	53	hgroup	28
cite	48, 182	hr	186
code	48, 182	html	21
col	67	i	48, 183
colgroup	68	iframe	90, 186
datalist	132	img	78, 85, 183
dd	59, 186	input	121, 183
del	48, 183	ins	48, 183
details	60	isindex	53

kbd	48, 183	rt	183
keygen	133	ruby	187
label	137, 182	s	48, 55, 183
legend	137	samp	48, 183
li	57, 58, 187	script	88
link	22	section	29, 187
listing	55	select	131, 183
main	29	small	48, 183
map	76, 80	source	84
mark	48, 183	span	51, 183
marquee	170	strike	48, 55
menu	54, 58	strong	48, 184
meta	21, 25	style	141
meter	135, 183	sub	48, 183
nav	28, 187	summary	60
nextid	54	sup	48, 184
nobr	54	SVG	94
noembed	54	table	64, 187
noframes	54	tbody	68
noscript	88	td	69
object	54	textarea	129, 184
ol	57, 187	tfoot	68
optgroup	131	th	71
option	131	thead	68
output	136	time	50, 184
p	47, 187	title	25
param	138	tr	69
picture	85	tt	48, 55
plaintext	55	u	48, 55
pre	15, 187	ul	58, 187
progress	134	var	48, 184
q	48, 183, 214	video	82, 184
rb	55	wbr	182, 215
rp	183	xmp	55

CSS xususiyatlari

!important	142	border-top-right-radius	194
align-content	207	border-width	190
align-items	208	bottom	196
align-self	208	box-shadow	190
all	151	box-sizing	189
animation	170	caption-side	212
animation-delay	172	clear	205
animation-direction	173	clip	208
animation-duration	171	color	215
animation-fill-mode	173	column-count	209, 210
animation-iteration-count	172	column-fill	209
animation-name	170	column-gap	209
animation-play-state	174	column-rule	209
animation-timing-function	171	column-rule-color	209
backface-visibility	179	column-rule-style	209
background	205	column-rule-width	209
background-attachment	201	columns	210
background-clip	201	column-span	209
background-color	201	column-width	210
background-image	159	content	229
background-origin	202	counter-increment	229
background-position	203	counter-reset	228
background-position-x	204	cursor	167
background-position-y	204	direction	210
background-repeat	204	display	184
background-size	202	empty-cells	213
border	191	filter	155
border-bottom	192	flex	200
border-bottom-left-radius	194	flex-basis	200
border-bottom-right-radius	194	flex-direction	199
border-collapse	192	flex-flow	199
border-color	191	flex-grow	200
border-image	194	flex-shrink	200
border-left	192	flex-wrap	199
border-radius	193	float	206
border-right	192	font	217
border-spacing	193	font-family	216
border-style	191	font-kerning	216
border-top	192	font-size	216
border-top-left-radius	194	font-stretch	217

font-style	217	page-break-before	226
font-variant	217	page-break-inside	226
font-weight	217	pointer-events	227
height	189	position	197
hyphens	215	quotes	214
image-rendering	203	resize	214
justify-content	208	right	196
left	196	table-layout	213
letter-spacing	215	tab-size	214
line-height	215	text-align	219
list-style	219	text-align-last	220
list-style-image	218	text-decoration	221
list-style-position	218	text-decoration-color	221
list-style-type	218	text-decoration-line	220
margin	188	text-decoration-style	221
margin-bottom	188	text-indent	221
margin-left	188	text-overflow	221
margin-right	188	text-shadow	222
margin-top	188	text-transform	222
marks	225	top	196
max-height	190	transform	180
max-width	190	transform-origin	179
min-height	190	transform-style	179
min-width	190	transition	177
object-fit	210	transition-delay	178
opacity	219	transition-duration	178
order	200	transition-property	178
orphans	225	transition-timing-function	178
outline	206	unicode-bidi	222
outline-color	206	user-select	227
outline-offset	206	vertical-align	211
outline-style	206	visibility	188
outline-width	206	white-space	223
overflow	211	widows	226
overflow-x	211	width	189
overflow-y	211	word-break	223
padding	188	word-spacing	224
padding-bottom	188	word-wrap	224
padding-left	188	writing-mode	224
padding-right	188	z-index	207
padding-top	188	zoom	180
page-break-after	226		

Foydalanilgan adabiyotlar

- «Ахборот коммуникация технологиялари изоҳли лӯғати» Амиров Д.М. вб 2-н. Тошкент 2010
- «Веб-мастеринг: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка» Петр Ташков. «Питер» 2010
- «HTML и CSS. Путь к совершенству» Бен Хеник. «Питер» 2011
- «HTML5, CSS3 и Web 2.0. Разработка современных веб-сайтов» Дронов В. «БХВ-Петербург» 2011
- «HTML5. Рецепты программирования» Шмитт К. Симпсон К. «Питер» 2012
- «Пуленепробиваемый веб-дизайн. Библиотека специалиста» Седерхольм Д. З-н. «Питер» 2012
- «HTML и CSS. Разработка и дизайн веб-сайтов» Дакетт Джон. «Эксмо» 2013
- «Создай свой веб-сайт с помощью HTML и CSS» Йен Ллойд «Питер» 2013
- «Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5» Робин Никсон. 4-н. «Питер» 2016
- «Большая книга CSS3» Дэвид Сойер Макфарланд. «Питер» 2014
- «HTML5 и CSS3. Веб-разработка по стандартам нового поколения» Хоган Б. 2-н. «Питер» 2014
- «HTML5, CSS3 и JavaScript. Исчерпывающее руководство» Джениффер Роббинс. «Эксмо» 2014
- «Современный веб-дизайн. Настольный и мобильный» Юлия Сырых «Вильямс» 2014
- «CSS. Карманный справочник» Эрик А. Мейер. «Вильямс» 2016
- «Новая большая книга CSS» Дэвид Сойер Макфарланд. «Питер» 2016
- «Изучаем HTML, XHTML и CSS» Элизабет Фримен, Эрик Фримен. 2-н. «Питер» 2016
- «HTML5. Карманный справочник» Джениффер Роббинс. 5-н. «Вильямс» 2016
- «PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов» Владимир Дронов. «БХВ-Петербург» 2016
- «HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств» Бен Фрейн. «Питер» 2017
- «Murach's HTML5 and CSS3» Anne Boehm. 4-н. «Mike Murach & Associates» 2018
- «HTML and CSS: Visual QuickStart Guide» Joe Casabona. 9-н. «Peachpit Press» 2020
- «Responsive Web Design with HTML5 and CSS» Ben Frain. 3-н. «Packt Publishing» 2020
- «HTML5 & CSS3 - Create a responsive website» Roxane Anquetil 2020

MUNDARIJA

1-BO'LIM. KIRISH

1.1. Kitob hususida.....	3
1.2. Qisqacha tarix	5

2-BO'LIM. HTMLning ASOSIY TARKIBI

2.1. Asosiy tushunchalar.....	9
2.2. Ilk bosqich.....	15
2.3. Asosiy qismlar	18
2.4. Sahifaning bosh qismi	21
2.5. Sahifaning asosiy qismi.....	26
2.6. Alomatlar qiymatlarining turlari.....	31
2.7. Universal alomatlar	36
2.8. Hodisalar.....	42
2.8.1. Sahifaga bog'liq hodisalar.....	42
2.8.2. Sichqoncha harakatiga bog'liq hodisalar	42
2.8.3. Klaviaturaga bog'liq hodisalar	44
2.8.4. Forma hodisalari.....	44
2.8.5. Media hodisalar:	45

3-BO'LIM. MATNNI JOYLASHTIRISH

3.1. Xatboshi va bloklar.....	46
3.2. Matnni jilolash	48
3.3. Eskirgan teglar	51
3.4. Ro'yxat	56
3.4.1. Sanoqli ro'yxat.....	56
3.4.2. Belgilangan ro'yxat	58
3.4.3. Izohli ro'yxat.....	59
3.4.4. Izohi yashirin ro'yxat.....	59
3.5. Ishorat.....	60

4-BO'LIM. JADVALLAR

4.1. Jadvalni e'lon qilish.....	64
4.2. Jadvalning sarlavhasi	66
4.3. Jadvalni shakllantirish.....	69
4.4. Jadval chizish	72

5-BO'LIM. MEDIA FAYLLARNI BOG'LASH

5.1. Xaritali rasm qo'yish	76
5.2. Tovushli fayl qo'yish.....	80
5.3. Lavhali fayl qo'yish.....	82
5.4. Tanlangan faylni yuklash	83
5.5. Nostandard media qo'yish.....	88
5.6. Obyektlarni biriktirish.....	89
5.7. Tashqi HTML faylni akslantirish.....	90

6-BO'LIM. GRAFIK VA TASVIRLAR CHIZISH

6.1. CANVAS tegi.....	93
6.2. SVG tegi.....	94
6.3. Geometrik shakllar qo'yish.....	98
6.3.1. To'g'ri to'rtburchak	100
6.3.2. Doira	100
6.3.3. Ellips	101
6.3.4. Kesma	101
6.3.5. Siniq chiziq	102
6.3.6. Ko'pburchak	102
6.4. Murakkab shakllar chizish.....	103
6.4.1. PATH tegi.....	103
6.4.2. Markerdan foydalanish	107
6.4.3. Tasvirni o'zgacha shakllantirish.....	109
6.4.4. Tuslantirish	110
6.5. Harakatlanuvchi tasvirlar	112
6.5.1. SET tegi.....	113

6.5.2. ANIMATE tegi.....	114
6.5.3. AnimateMotion tegi.....	115
6.5.4. AnimateTransform tegi.....	116

7-BO'LIM. FORMA ELEMENTLARI

7.1. Forma yaratish.....	118
7.2. INPUT darchasi.....	121
7.3. INPUT darchasi turlari.....	125
7.4. Forma darchalari	128
7.4.1. Tugma	128
7.4.2. Matn yozish darchasi.....	129
7.4.3. Tayinlanuvchi (select) darcha	130
7.4.4. Tanlanuvchi satr.....	132
7.4.5. Ma'lumotlarni shifrlash	133
7.4.6. Jarayonlarni ko'rsatish	134
7.5. Formaga obyektlarni joylash.....	137

8-BO'LIM. CSS TARKIBI

8.1. CSSni bog'lash.....	140
8.2. Umumiylar sintaksis.....	143
8.3. Qiymatlarning turlari.....	150
8.4. Funksiyalar.....	154
8.4.1. Xiralashtirish	155
8.4.2. Yorqinlashtirish	156
8.4.3. Xijillashtirish	156
8.4.4. To'yintirish	156
8.4.5. Shaffoflik	157
8.4.6. Soya qo'shish	157
8.4.7. Oq-qora tasvir	158
8.4.8. Rasm rangini o'zgartirish.....	158
8.4.9. Orttasvir ranglari	159
8.4.10. Hisoblangan qiymat.....	162

8.5. Tanlov asosida biriktirish	163
8.6. Tabaqalash	164
8.7. Harakatlantirish	170
8.7.1. Harakatlantiruvchi xususiyatlar.....	170
8.7.2. Harakatlanishni ta'minlash.....	174
8.7.3. Xususiyatlarlarni umumlashtirish.....	175
8.7.4. Sodda harakatlashtirishlar	177
8.7.5. Harakatlashtirish samaralari.....	179
8.8. Qatorli va blokli teglar.....	181
8.9. Obyekt shakli.....	187

9-BO'LIM. OB'YEKTLARNI JOYLASH VA JILOLASH

9.1. Obyektni joylashtirish	196
9.2. Orqa tasvir.....	200
9.3. Nisbiy joylashtirish	205
9.4. Ichki elementlarini akslantirish	207
9.5. Xos xususiyatlar	212
9.6. Ichki matnni jilolash	215
9.7. Chop etish xususiyatlari	225
9.8. Hodisalarни boshqarish	227
9.9. Vaziyatan sinflash	228

10-BO'LIM. MUKAMMAL SAYT YARATISH TALABLARI

10.1. Bexato kod yozish.....	232
10.2. Moslashuvchan CSS yaratish.....	237
10.2.1. O'zgaruvchilar.....	239
10.2.2. Biriktirish.....	240
10.2.3. Fayldan yuklab olish	241
10.2.4. Funksiyadan foydalanish	242
10.2.5. Meros olish	244
10.2.6. Matematik amallar.....	245

10.3. Modal oyna.....	247
10.4. Qisqa izoh	250
10.5. Harakatlanuvchi menuy.....	252
10.6. Qalqib chiquvchi oyna.....	255
10.7. Yig'iladigan maydon.....	261
10.8. Mo'jaz hisobotlar.....	265
10.9. Forma darchalarining nuqsonlari.....	275
Teglar ko'rsatkichi	278
CSS xususiyatlari	280
Foydalaniłgan adabiyotlar	282

Javlon ABDULLO

MUKAMMAL DASTURLASH

1-kitob:

HTML VA CSS

Muharrir: Abdulla SHAROPOV

Badiiy muharrir: Bahriiddin BOZOROV

Texnik muharrir: Dilshod NAZAROV

Sahifalovchi: Inomjon O'SAROV

Musahhih: Mahfuza IMOMOVA

Nashriyot litsenziyası: AI №134, 27.04.2009

Terishga berildi: 05.04.2021-y.

Bosishga ruxsat etildi: 15.06.2021-y.

Ofset qog'oz. Qog'oz bichimi: 60x84 $\frac{1}{16}$.

Georgia garniturasi, Ofset bosma.

Hisob-nashriyot t.: 12,3. Shartli b.t.: 16,7.

Adadi: 1500 nusxa.

Buyurtma №72

«AKADEMNASHR» nashriyotida tayyorlandi va chop etildi.
100156, Toshkent shahri Chilonzor tumani 20⁴-mavze 42-uy.

Tel.: (+99871) 216-87-81

e-mail: info@akademnashr.uz

web: www.akademnashr.uz

56000

3d. 973 - 018.

AliceBlue	#F0FFF	DarkSlateGray	#2F4F4F	LightSeaGreen	#20B2AA	PapayaWhip	#FFEFDS
AntiqueWhite	#FAEBD7	DarkTurquoise	#000CED1	LightSkyBlue	#87CEFA	PeachPuff	#FFDAB9
Aqua	#00FFFF	DarkViolet	#9400D3	LightSlateGray	#778899	Peru	#CD853F
Aquamarine	#7FFF4D	DeepPink	#FF1493	LightSteelBlue	#B0C4DE	Pink	#FFFC0CB
Azure	#F0FFFF	DeepSkyBlue	#00BFFF	LightYellow	#FFFFE0	Plum	#DDA0DD
Beige	#F5F5DC	DimGray	#696969	Lime	#00FF00	PowderBlue	#B0E0E6
Bisque	#FFE4C4	DodgerBlue	#1E90FF	Linen	#FAF0E6	Purple	#800080
Black	#000000	FireBrick	#B22222	LightCoral	#FF0000	RebeccaPurple	#663399
BlanchedAlmond	#FFEB3D	FloralWhite	#FFF0F0	Magenta	#FF00FF	Red	#FF0000
Blue	#0000FF	ForestGreen	#228B22	Maroon	#800000	RosyBrown	#BC8F8F
BlueViolet	#8A2BE2	Fuchsia	#FF00FF	MediumAquaMarine	#66CDAA	RoyalBlue	#4169E1
Brown	#A52A2A	Gainsboro	#DCDCDC	MediumBlue	#0000CD	SaddleBrown	#8B4513
BurlyWood	#DEB887	GhostWhite	#F8F8FF	MediumOrchid	#BA55D3	Salmon	#FA8072
CadetBlue	#5F9EA0	Gold	#FFD700	MediumPurple	#9370DB	SandyBrown	#F4A460
Chartreuse	#7FFF00	GoldenRod	#DAA520	MediumSeaGreen	#3CB371	SeaGreen	#2E8B57
Chocolate	#D2691E	Gray	#808080	MediumSlateBlue	#7B68EE	SeaShell	#FFF5EE
Cornsilk	#FFF8DC	Green	#008000	MediumSpringGreen	#008080	Sienna	#A0522D
Coral	#FF7F50	GreenYellow	#ADFF2F	MediumTurquoise	#48D1CC	Silver	#C0C0C0
CornflowerBlue	#6495ED	HoneyDew	#F0FFF0	MediumVioletRed	#C71585	SkyBlue	#87CEEB
Crimson	#DC143C	HotPink	#FF69B4	MidnightBlue	#191970	SlateBlue	#6A5ACD
Cyan	#00FFFF	Indigo	#4B0082	MintCream	#F5FFFA	SlateGray	#708090
DarkBlue	#00008B	IndianRed	#CD5C5C	MistyRose	#FFE4E1	Snow	#FFFAFA
DarkCyan	#008080	Ivory	#FFFFF0	Moccasin	#FFEB3B	SpringGreen	#00FF7F
DarkGoldenRod	#B8860B	Khaki	#F0E68C	NavajoWhite	#FFDEAD	SteelBlue	#4682B4
DarkGray	#A9A9A9	Lavender	#E6E6FA	Navy	#000080	Tan	#D2B48C
DarkGreen	#006400	LavenderBlush	#FFF0F5	OldLace	#FDF5E6	Teal	#008080
DarkKhaki	#BDB76B	LawnGreen	#70CF00	Olive	#808000	Thistle	#D8BFDB
DarkMagenta	#80008B	LemonChiffon	#FFFFACD	OliveDrab	#6B8E23	Tomato	#FF6347
DarkOliveGreen	#556B2F	LightBlue	#ADD8E6	Orange	#FFA500	Turquoise	#4680D0
DarkOrchid	#9932CC	LightCoral	#F08080	OrangeRed	#FF4500	Violet	#EE82EE
DarkOrange	#FF8C00	LightCyan	#E0FFFF	Orchid	#DA70D6	Wheat	#F5DEB3
DarkRed	#8B0000	LightGray	#D3D3D3	PaleGoldenRod	#EEE8AA	White	#FFFFFF
DarkSalmon	#E9967A	LightGreen	#90EE90	PaleGreen	#98FB98	WhiteSmoke	#F5F5F5
DarkSeaGreen	#87BC8F	LightPink	#FFB6C1	PaleTurquoise	#AFEEEE	Yellow	#FFFF00
DarkSlateBlue	#483D8B	LightSalmon	#FFA07A	PaleVioletRed	#DB7093	YellowGreen	#9ACD32

ISBN 978-9943-6964-4-0



9 789943 696440

