

CSS

CSS 3 haqida

CSS - bu stillar bilan ishlash uchun mo'ljallangan kodlar majmuasidir. Hozirgi kunga kelib yangi CSS3 versiyasi ishlab chiqilmoqda, lekin hali ko'p brauzerlar bu stillarni aks ettira olmayapti. Shuning uchun CSS3 yaratuvchilari brauzer o'rtasida kelishmovchiliklarni oldini olish maqsadida brauzer turiga qarab har xil prefikslar ishlab chiqishdi va bu prefikslardan CSS3 to'liq yaratilmaguncha ishlatish tavsiya etilgan, CSS3 to'liq yaratilgandan so'ng bu prefikslar olib tashlanadi. Bu prefikslar quydagilar:

Safari va Chrome brouzerlari uchun -webkit-

Opera brouzeri uchun -o-

Firefox brouzeri uchun -moz-

Internet Explorer(IE) brouzeri uchun esa -ms-

Man bu prefikslarni ishlatmagan holda misollar keltirdim, agar sizda bu misollar ishlamasa oldiga prefikslarni qo'yib ishlatib ko'ring.

Css3 da eng asosiy qo'shilgan stillardan biri bu animastiyalardir, ya'ni stillar almashinish jarayoni birdaniga emas, balki sekin astalik bilan sodir bo'lishidir. Quyida Css3 versiyasiga yangi qo'shilgan ba'zi stillarni misol tariqasida ko'rsatib o'taman.

- Elementlarni(matn, rasm, fon,...) och(прозрачный) ko'rinishda aks ettirish.

background-color: rgba(10,0,255,0.7);

Bu erda sahifa foniga rang berib, shu rangni 0.7 darajali och tusga keltirilmoqda, tanishing CSS3 ning yangi elementi rgba.

- Elementlarning istalgan burchagiga(qismiga) bir vaqtning o'zida har xil rasmlarni joylashtirish. Bu saytni verstka qilish jarayonini yengillashtiradi.

background:

url(top.gif) top left no-repeat,

url(center.png) top 11px no-repeat,

url(bottom.png) bottom left no-repeat,

url(middle.png) left no-repeat;

- Resize buyrug'i. Bu buyruq orqali foydalanuvchi istalgan elementning o'lchamini o'zgartirish mumkin bo'ladi.

div.resize {

width: 25px;

height: 35px;

resize: both;

}

- CSS3 ning yana bir yangiligi bu – elementlar(shakllar, bloklar) burchagini istalgan radiusda burish mumkinligidir. Ko'p web saytlar forma yaratish jarayonida, formaning burchaklarini burishni verstka paytida biror rasm orqali amalga oshirishadi, bu esa qo'shimcha ish va vaqt yo'qotishga olib keladi, CSS3 da bu muammo bartaraf etilgan va quyidagicha amalga oshirish mumkin bo'ladi.

#forma {

border-bottom-right-radius: 2em;

border-bottom-left-radius: 1em;

border-top-left-radius: 5em;

border-top-right-radius: 3em;

}

- CSS3ning soyalar bilan ishlash qismi. Barcha “p” teglari uchun soyalar hosid qilish.

p {

text-shadow: #003471 /* soya rangi */ 2px /* o'ng tomonga surilishi*/ 5px

/*pastga surilish */ 2px /* размытие*/;

}

- Shriftlar. Internetda ko'p foydalanuvchilar verdana shrifti bilan ishlaydi. Nega? Chunki bu shrift barcha kompyuterlarda mavjud va brouzerda chiroyli ko'rinishga ega. Agar stillarda qo'llanilgan shrift foydalanuvchi kompyuterida mavjud bo'lmasa, brouzer matnni istalgan boshqa shriftda ko'rsatishi mumkin. Bu esa shriftlar rang barangligiga olib keladi. CSS3 da shriftlar bilan ishlash uchun yangi komanda @font-face.

@font-face {

font-family: shrift_akm;

src: url('http://blabla.uz/fonts/shrift_akm.ttf');

}

h1 {

font-family: shrift_akm;

}

- Web sahifada kolonkalar. Bunisiga nima deysiz? Element ichidagi matnlarni bir necha ustunli kolonkalarda chiqarishingiz mumkin.

div {

column-width: 15em;

column-gap: 2em; /* yashil rangda */

column-rule: 4px solid red; /* qizil rangda */

padding: 5px; /* qora rangda */

}-----

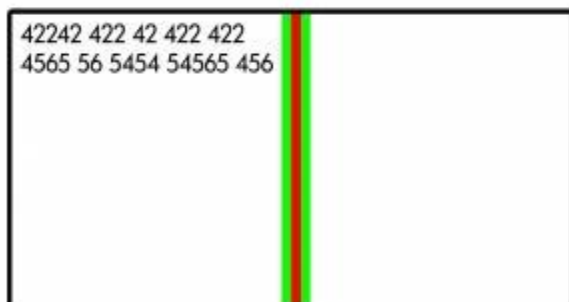
<div>

42242 422 42 422 422

4565 56 5454 54565 456

...

</div>



№1 - CSS dars: Hamma joyda CSS, nima o'zi u???

CSS – bu stillar bilan ishlay oladigan **kaskadli stillar majmuasi**dir. Web sahifalarga har xil stillar berish uchun ishlatiladi. HTML xujjat ichida teglar bilan birga foydalaniladi. CSS kodlarini kompilyastiya qilish uchun qandaydir kompilyatorlar, qandaydir dasturlar, va bu kod tushuna oladigan qandaydir redaktorlar kerak bo'lmaydi. Bu kodlarni HTML singari web brauzerning o'zi kompilyastiya qiladi va natijani chiqarib beradi.

Bu stillar jadvali(CSS) to'liq holatda **1997** yilda tashkil topib, "**WWW Consorcium**" ida qo'llab quvvatlangan va foydalanishga taqdim etilgan. O'sha paytlarda **HTML ning 3** versiyasi ommabop edi. CSS ni dastlab **Netscape Navigator 4.0** va **Internet Explorer 4.0** brouzerlarida ko'rish mumkin bo'lgan. Hozirda barcha brouzerlar css da yozilgan kodlarni tushuna oladi. Bu stillarga oid dasturlash tilini o'rganish sodda bo'lib, yangi stillar qo'shilishi natijasida kengayib bormoqda. CSS ni HTML xujjati ichida ishlatilishini birin ketin ko'rib o'tamiz:

1. Web sahifada ishlatiladigan CSS kodlari orqali hosil qilinadigan stillarni hammasini bitta faylga yozib, uni serverda saqlaymiz. So'ng, sahifadan, serverdagi saqlab qo'yilgan faylga murojaat qilib, kerakli stilni olamiz. Bu usul katta hajmdagi web sahifalar yaratishda foydali hisoblanadi. Faylga murojaat, **<head>** tegi ichida amalga oshiriladi va quyidagi ko'rinishga ega bo'ladi.

```
<LINK REL=STYLESHEET TYPE="text/css" HREF="URL">
```

<link> tegi sahifa biror faylga murojaatni amalga oshirmoqda degani, **REL**– fayl qandaydir stillardan iborat degani, **TYPE** - stillar css kodlaridan iborat va fayl joylashgan manzil yoziladi.

1. CSS kodlar veb sahifa ichidagi **<head>** tegi orasida beriladi. Bunda yaratilgan stillar shu sahifaga tegishli bo'ladi, bu degani faqat shu sahifada ishlatilish mumkindir.

```
<head>< STYLE TYPE="text/css">
```

```
<!--
```

```
A {text-decoration:none; }
```

```
-->
```

```
</--STYLE></head>
```

2. Navbatdagi turi, bu har bir element uchun alohida teg ichida stil berib chiqishdan iborat. Misol uchun, "p" tegi ichida biror matn yozilsa, shu matn uchun teg ichida alohida stil beriladi va bu stil shu teg uchun xususiy bo'ladi. Bu usuldan foydalanishni maslahat bermiman, chunki bu usul kodlarni chalkash va tushunarsiz bo'lishga olib keladi. Shoshilinch vaziyatlarda yoki stil qay tartibda namoyon bo'lishni ko'rib olish uchun bu usuldan foydalanishni maslahat beraman.

Undan tashqari **<style>** tegi orqali css faylni import qilish mumkin bo'ladi.

```
@import: url(mystyles.css);
```

CSS stillarini ishlatishda, quyidagi guruhlash usullaridan foydalanish, tartibli web sahifa tuzishga olib keladi:

- Kodlarni ixchamlash uchun, har hil elementlar uchun bir hil tipdagi stillarni guruhlab ishlatish maqsadga muvofiqdir.

```
H1 {font-family: Verdana}
```

```
H2 {font-family: Verdana}
```

Yuqoridagi stilni guruhlab quyidagicha yozamiz.

H1, H2 {font-family: Verdana}

- Guruhlashda bir hil element uchun bir necha stillarni birlashtirish mumkin.

H2 {font-weight: bold}

H2 {font-size: 14pt}

H2 {font-family: Verdana}

Barcha stillarni bittaga yifamiz.

H2 {font-weight: bold; font-size: 14pt; font-family: Verdana;}

- Ba'zi stillar bitta stil ichida berilishi ham mumkin.

H2 {font: bold 14pt Verdana}

CSS stillari bilan ishlaganda, element ichidagi elementlar ham o'zidan yuqori turgan element stilini qabul qiladi, ya'ni o'zidan oldingi element "ota" vazifasini bajaradi va o'z "farzandlari"ga ham o'zida borini beradi. Misol uchun, <p> tegi ichidagi matn ko'k rangda yozilishi lozim bo'lsa(P {color: blue}), <p> tegi ichidagi tegiga tegishli bo'lgan matn ham ko'k rangda yoziladi.

Ba'zi stillar faqat yakka tartibda ishlaydi, ya'ni ichki teglarga stillari o'tmaydi(misol uchun, background), shuning uchun barcha stillarni birma bir ishlatib, o'rganib olish lozim. <body> tegiga stil berib, dastlabki o'rnatilish(po umolchaniyu) jarayonini hosil qilish mumkin. Shunda biror elementga stil berish esdan chiqib qolsa, <body> tegida ta'luqli stil esdan chiqqan elementga o'rnatiladi.

BODY {color: green;

font-family: "Verdana";

background: url(joke.gif) white; }

№2 - CSS dars: Selektor, Klass, Identifikator tushunchasi

Oldingi maqolamizda CSS haqida ma'lumot berishga harakat qildim. Endi chuqurroq (chuqurroooooq qaziganimiz uchun)o'rganishni boshlaymiz. Bu maqolada, CSS ga tegishli 3 ta yangi terminni o'rganib chiqamiz: selektor, klass, identifikator. Demak boshladik...

Selektor.

CSS da **selektor** vazifasini **HTML teglar** bajaradi. Veb sahifadagi barcha bir hil tipdagi teglarga(selektorlarga) umumiy stil beramiz. Misol uchun saytdagi barcha "ссылка"larning tegiga chizishni olib tashlaymiz(sizga ma'lumki, "ссылка" larga stil berilmasa odatiy tegiga chizilgan holatda qoladi).

```
1 <STYLE TYPE="text/css">
```

```
2 A {text-decoration:none; }
```

```
3 </STYLE>
```

Klass.

Agar bizga, kod ichida foydalaniladigan bir xil teglarga har xil stillarni o'rnatishimiz lozim bo'lsa, klass yoki identifikatorlardan foydalanish juda qulay hisoblanadi. Dastlab klass haqida to'xtalib o'tamiz. Bir xil elementlarga har xil stil berish uchun, element nomi va nuqta(.) qo'yib, stilga biror nom beramiz. So'ng shu nomga tegishli bo'ladigan

stillar ketma-ketligini yozib chiqamiz. Shundan so'ng, kod ichida kerakli bo'lgan joyda, **class** so'zi yordamida o'rnatilgan stil nomi chaqiriladi.

Misol ko'ramiz:

```
1 <head>
2 <style TYPE="text/css">
3 H1.rd {color:red;}
4 H1.be {color:red;background-color:blue;}
5 </style>
6 </head>
```

E'lon qilingan stilni kerakli joyda chaqiramiz.

```
1 <body>
2 <h1 class="rd">Qizil shrift</h1>
3 <h1 class="be">Ko'k fonda qizil
4 shrift</h1>
5 </body>
```

Shu tariqa istalgan elementga istalgan stillarni o'rnatish mumkin bo'ladi. YUqoridagi misolda faqat bitta element(H1) uchun stil yaratdik. Agar biz biror stilni yaratib, uni istalgan element uchun ishlatmoqchi bo'lsak, u holda yaratilgan stilga element nomini qo'ymasdan, faqatgina nuqta orqali stil uchun doimiy nom beramiz va kerakli joyda chaqiramiz. Tushunmadingizmi? Unday bo'lsa quyidagi misolni yaxshilab tahlil qilib chiqing.

```
1 <STYLE TYPE="text/css">
2 .red{color: red; }
3 .blue {color:red; background-color: blue}
4 </STYLE>
```

Biz 2 ta stil yaratdik, "**red**" va "**blue**" deb nomladik, endi bu stillarni istalgan element uchun ishlatishimiz mumkin.

```
1 <P CLASS="red">Qizil shrift</P>
2 <P CLASS="blue">Ko'k fonda
3 qizilshrift</P>
```

Identifikator.

Bu o'z ismi bilan etib turilibdi **IDENTIFIKATOR**, ya'ni butun veb sahifasida faqatgina bitta elementga stil berish uchun ishlatiladi. Dastlab stil yozilib, unga "#" belgisi orqali nom beriladi. "#" bu identifikatorli stillar to'plami degan ma'noni beradi.

#akm{font-family:verdana; color:green;}

So'ng kod ichida bu identifikator **ID** kalit so'zi orqali chaqiriladi va yagona element uchun ishlatiladi. Bu ta'rif identifikatorga berilgan xaqiqiy ta'rifdir. Lekin hozirgi kundagi ko'p brouzerlarda id orqali bir necha elementga ham stil berish mumkin bo'lib qoldi, brouzerlar bunga ruhsat berishmoqda.

```
1 <STYLE TYPE="text/css">
2 #myID {letter-spacing: 1em; }
3 H1#form3 {color:red; background-color:
4 blue}
5 </STYLE>
6 <BODY>
7 <P ID=myID> 1em lik yozuv</P>
```

```
<H1 ID=form2>Qora rangda</P>
```

YUqoridagi kodda "form3" identifikatori e'lon qilindi, lekin "form2" identifikatori ishlatildi. Bunda "form2" bo'lmaganligi sababli yozuv odatiy holda namoyon bo'ldi. Hozirgi kunda identifikatordan klassda yaratilgan stillarning ma'lum bir qismini o'zgartirish uchun ishlatishmoqda. Misol uchun, bitta klass ochib, unga nom beramiz va stillar ketma-ketligini kiritamiz. HTML kod ichida shu klassni chaqirib, undan foydalanamiz, agar bizga stillar ketmaketligini biror elementini o'zgartirish kerak bo'lib qolsa, identifikatorni ishga solamiz. Shunda butun stilni bitta elementini o'zgartirish uchun boshidan yozib o'tirmaymiz. HTML tegida klass va identifikatorni birga ishlatsak, identifikator kuchliroq hisoblanadi. Misol ko'ramiz.

```
1 <style>
2 .klas{color:blue;font-family:verdana;}
3 #iden{color:red;}
4 </style>
5 <p class="klas" id="iden">test_klas</p>
6 <p id="iden">test_klas</p>
```

Ikkala holatda ham matn qizil rangda chiqadi, identifikatorda o'rnatilgani kabi. Oxirgi berilgan stil natijaga ta'sir qiladi.

№3 CSS dars - Sintaksis va ishlash printsiplari: ID, CLASS, STYLE

Oldingi darslardan CSS haqida biroz bo'lsa ham ma'lumotga ega bo'ldingiz. Bu maqola to'liq, CSS kodlarini HTML kodlarida ishlatishni ko'rsatib o'taman. CSS kodlarini tushuntirib o'tmayman, faqat ulashni va yozilish sintaksisini ko'rsataman, keyingi maqolalardan boshlab, CSS stillarini birma-bir ko'rib chiqamiz. CSS natijalarini brouzerda ko'rish mumkin bo'ladi, yana bir narsa, CSSga tegishli barcha darslar misollari "**Firefox Mozilla**" brouzeri orqali tekshiriladi, agar sizda boshqa brouzer bo'lsa, misol natijalari sal farq qilishi mumkin.

CSS stillar quyidagicha yoziladi:

```
H1{color: blue; font-size: 14 px}
```

1. **Selektor nomi** - yozilgan stilni qaysi tegda foydalanish(misolda **H1**).
2. **Qavs** - stillar boshlanganini bildiradi.
3. **Stil xususiyati(Свойство)** - qaysi stil xususiyatidan foydalanishni o'rnatish(misolda **color, font-size**).
4. **Qiymat** - tanlangan stil xususiyat qiymati(misolda **blue, 14 px**).

Agar bir necha stil xususiyatlaridan foydalaniladigan bo'lsak, ular qo'shtirnoq(;) orqali ajratiladi, oxirida qo'shtirnoq qo'yish shart emas. Stil xususiyatlari yig'indisi shaklli qavs ichiga olinishi shart({ }) va har bir stil xususiyatidan so'ng, uning qiymatini berishdan oldin ikki nuqta(:) qo'yilishi kerak.

Sintaksisi bilan tanishib chiqdik, endi HTML kodga ulashni ko'rsatib o'taman. Bir necha xil usulda ulash mumkin:

1. CSS stillarini alohida faylga yozib, bu faylni HTML kodida chaqirib ishlatish mumkin. Misol uchun, "**style.css**" nomli fayl yaratib ichiga quyidagi stil xususiyat va qiymatlarini kiritamiz.

```
1 h1{color:blue;font-size:14px}
```

Css ni bilish uchun HTML haqida ko`proq ma`lumotlarni bilgan yaxshi

Undan so'ng, "**index.html**" fayl yaratib, bu faylni yuqorida yaratgan "**style.css**" nomli fayl bilan bitta papkaga joylashtiramiz, so'ng bu faylga quyidagi HTML teglarni kiritamiz(HTML teglar bilan tanishish).

```
1 <html>
2 <head>
3 <title>CSS bilan ishlash</title>
4 <link rel="stylesheet" type="text/css"
5 href="style.css" >
6 </head>
7 <body>
8 <h1>Hello World CSS!</h1>
9 </body>
10 </html>
```

HTML kodni ishga tushiramiz va quyidagi natijani olamiz,

Hello World CSS!

Ko'rib turganingizdek "**Hello World CSS!**" yozuvi, "**style.css**" faylida ko'rsatgan stilimizni oldi, ya'ni ko'k rang va 14 piksel o'lcham.

2. HTML hujjat ichida, stil xususiyatlarini aniqlashtirib qo'yish mumkin.

```
1 <html>
2 <head>
3 <title>CSS bilan ishlash</title>
4 <style type="text/css">
5 h1{color:blue;font-size:14px}
6 </style>
7 </head>
8 <body>
9 <h1>Hello World CSS!</h1>
10 </body>
11 </html>
```

Natija, yuqoridagi bilan bir xil, bunda bitta tegga bitta stil berildi, agar o'zgarishlar qilinmasa, shu teg qayerda ishlatilsa, hamma yerda shu stil dan foydalaniladi.

3. Identifikator orqali bog'lash. O'zgarmas nomga ega bitta stillar xususiyati yoziladi va u faqat bitta element uchun ishlatiladi. Ishlatishda "**ID**" kalit so'zidan foydalaniladi, identifikator esa "#" belgisi bilan e'lon qilinadi.

```
1 <html>
2 <head>
3 <title>CSS bilan ishlash</title>
4 <style type="text/css">
5 #akm{color:blue;font-size:14px}
6 </style>
7 </head>
8 <body>
9 <h1 id="akm">Hello World CSS!</h1>
10 </body>
11 </html>
```


11

4. Klass orqali bog'lash. Bitta klass ichida bir necha stil xususiyatlari aniqlashtirib olinadi, so'ng, "**class**" kalit so'zi orqali chaqirilib foydalaniladi, nuqta(.) orqali e'lon qilinadi. Identifikatordan farqli ravishda, bir yoki bir necha element orqali ishlatish mumkin.

```
1 <html>
2 <head>
3 <title>CSS bilan ishlash</title>
4 <style type="text/css">
5 .akm{ color:red;font-size:14px}
6 </style>
7 </head>
8 <body>
9 <h1 class="akm">Hello World CSS!</h1>
10 <h3>Hello World CSS!</h3>
11 <h3 class="akm">Hello World CSS!</h3>
12 </body>
13 </html>
```

Hello World CSS!

Hello World CSS!

Hello World CSS!

5. HTML kodi ichida(**body** tegi orasida) bog'lanish, istalgan HTML kodi ichida, tegning "**style**" attributi orqali stil yaratish mumkin, bu stil faqat shu elementga xos bo'lib, boshqa element uchun ishlatib bo'lmaydi.

```
1 <html>
2 <head>
3 <title>CSS bilan ishlash</title>
4 <style type="text/css">
5 .akm{ color:red;font-size:14px}
6 </style>
7 </head>
8 <body>
9 <h1 style="color:green; font-
10 family:verdana">Hello World CSS!</h1>
11 </body>
12 </html>
```

Natija quyidagicha bo'ladi

Hello World CSS!

Bog'lanishni, qo'yilgan masala asosida, yuqoridagi biror turidan foydalanib amalga oshirishingiz mumkin bo'ladi.

№4 CSS dars - Fon va rang bilan ishlash: COLOR, BACKGROUND, BACKGROUND-*

Darsimiz, CSS orqali veb sahifa fonini o'zgartirish va rang berishga bag'ishlanadi. Bu ishlarni HTML orqali ham qilish mumkin, lekin bu darajadagi fonlarni chiqarish qiyinchilik tug'diradi. Shuning uchun veb sahifa yaratishda HTML gippermatn tili va CSS stil xususiyatlari aralashtirib ishlatiladi, shu tariqa birini kamchiligini biri yopadi. CSS da asosiy fon va rang uchun javob beradigan stil xususiyatlari quyidagilardan iborat:

- **color;**
- **background-color;**
- **background-image;**
- **background-repeat;**
- **background-attachment;**
- **background-position;**
- **background.**

Bu stil xususiyatlarini birma-bir ko'rib chiqamiz.

Color - u yoki bu elementga rang berish uchun ishlatiladi, misol uchun, hamma paragraf teglari qizil rangda, barcha h1 teglar qora rangda bo'lish kerak bo'lsa, quyidagicha stil beriladi.

```
1      <html>
2      <head>
3      <title>Rang_1</title>
4      <style type="text/css">
5      p{color:red}
6      h1{color:black}
7      </style>
8      </head>
9      <body>
10     <h1>Qora</h1>
11     <p>Qizil</p>
12     </body>
13     </html>
```

Qayerda bu teglar ishlatilsa, shu tegga mos holda stillar ham ishlaydi, natijani ko'ramiz:

Qora

Qizil

Background-color - biror element yoki butun veb sahifa foniga rang berish. HTML dagi "**bgcolor**"dan farqli ravishda, bu stil xususiyati istalgan element foniga rang berish mumkin: havolalarga, paragraflarga, jadvallarga,...

Misol ko'ramiz, butun sahifa foniga va barcha **H3** teglari foniga rang beramiz.

```
1      <html>
2      <head>
3      <title>Rang_2</title>
```

```

3      <style type="text/css">
4      body{background-color:gray}
5      h3{color:orange}
6      </style>
7      </head>
8      <body>
9      <h3>Fongini rangini qara, daxshatu a!</h3>
10     </body>
11     </html>
12

```

Rang berishda odatagidek rang nomi(ingliz tilida) yoki 16 lik sanoq tizimda berish lozim(16lik tizimdagi rangi aniqlash), endi natijani ko'ramiz:

Fongini rangini qara, daxshatu a!

Background-image - sahifa foni sifatida biror rasmni qo'yish, nafaqat sahifa boshqa elementlarga ham qo'yish mumkin, faqat yaxshi chiqmasligi ehtimoli katta. Misol ko'ramiz,

```

1      <html>
2      <head>
3      <title>Rang_3</title>
4      <style type="text/css">
5      body{
6      background-image: url(bum.jpg);
7      }
8      </style>
9      </head>
10     <body>
11     </body>
12     </html>

```



E'tibor bergan bo'lsangiz, rasmni ko'rsatish HTMLga nisbatan boshqacharoq. "**url**" kalit so'zi yozilib, so'ng rasm adresi berilmoqda, bu holda rasm sahifa fayli saqlangan papka bilan bir joyda joylashgan, agar boshqa joyda bo'lsa, (**image/bum.jpg**) ko'rinishida(rasmning to'liq adresi) berish kerak.

Background-repeat - qaytarilishni amalga oshirish. Agar yuqoridagi misolga e'tibor bergan bo'lsangiz, kichik rasmlar yuqori chap burchak bo'lab qaytarilib takrorlanmoqda va shu tariqa butun veb sahifa foni rasmga to'lmoqda.**Background-repeat** shu qaytarilishni boshqara oladi.

Bu stil xususiyatining quyidagi qiymatlari mavjud:

Background-repeat: repeat-x ; - gorizontal bo'yicha qaytarilish.

Background-repeat: repeat-y ; - vertikal bo'yicha qaytarilish.

Background-repeat: repeat ; - vertikal va gorizontal bo'yicha qaytarilish.

Background-repeat: no-repeat ; - qaytarilishni amalga oshmaydi.

Bunday xususiyatlar HTML gippermattn tilida mavjud emas. Keling misol ko'ramiz,

```
1 BODY {
2   background-image : url(bum.jpg) ;
3   background-repeat: repeat-x;
4 }
```

Fon sifatida tanlangan rasm, faqat gorizontal bo'yicha takrorlanadi.



Background-attachment - agar fon sifatida rasm bo'lsa, bu stil xususiyati o'rnatiladi. Bunda, agar sahifada o'tkazgich(prokrutka)bo'lsa, fon rasmi ham o'tish kerak yoki fon rasmi sahifaga qotirib qo'yilishi kerakligi ko'rsatiladi. Qiymatlari:

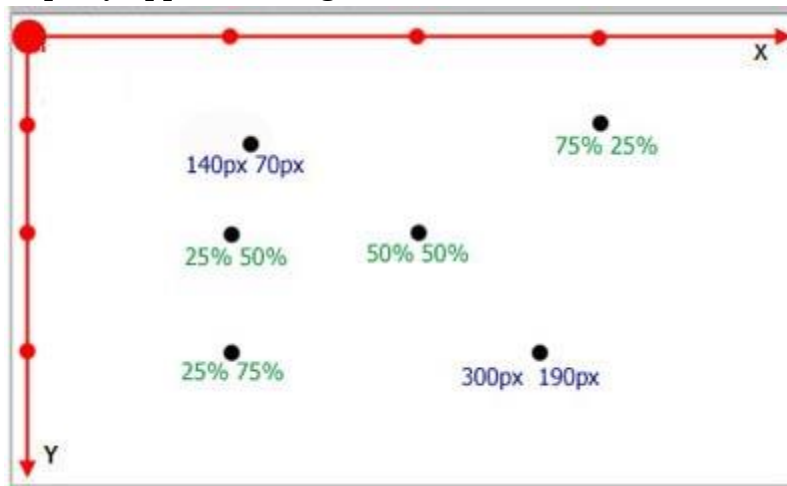
scroll - fon sifatida o'rnatilgan rasm, veb sahifa oynasi bilan birga o'tadi(prokrutka).

fixed- fon sifatida o'rnatilgan rasm, veb sahifa oynasiga qotirib qo'yilgan bo'ladi va o'zgarmaydi.

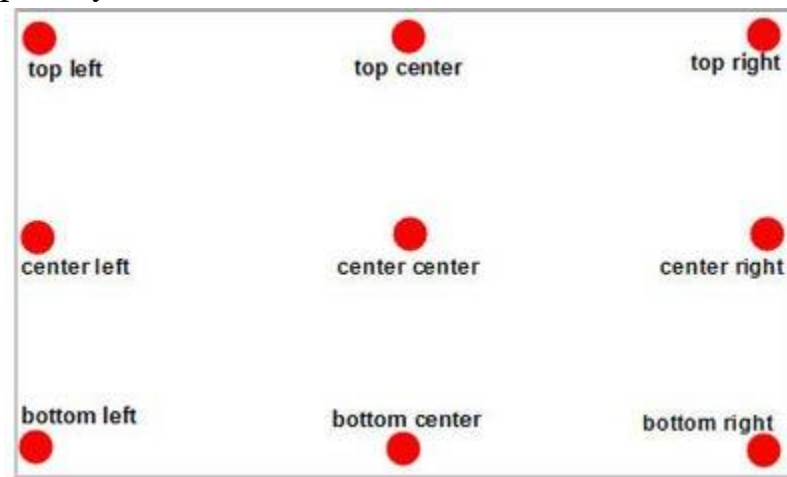
```
1 BODY {
2   background-image : url(bum.png) ;
3   background-repeat: no-repeat;
4   background-attachment: scroll;
5 }
```

Background-position - fon sifatida tanlangan rasmni pozitsiyasini o'rnatadi, ya'ni fon rasmi aynan qayerda joylashishini aniqlashtirish, qiymat foiz yoki uzunlik o'lchovida

beriladi. Boshlanish, chap yuqori tomondan boshlanadi. Quyidagi rasmda pozitsiya haqida yaqqol ko'rsatilgan.



Dastlab, gorizontaal bo'yicha qiymat beriladi, keyin vertikal bo'yicha. Qiymat brauzer o'lchamiga qarab piksel yoki foizda beriladi. Undan tashqari kalit so'zlar orqali ham pozitsiyani berish mumkin.



Keling endi misol ko'ramiz,

```
1      <html>
2      <head>
3      <title>Rang_4</title>
4      <style type="text/css">
5      BODY {
6      background-image : url(bum.jpg);
7      background-repeat: no-repeat;
8      background-position: center center;
9      }
10     </style>
11     </head>
12     <body>
13     </body>
14     </html>
```



yana bir misol,

```
1 BODY {  
2   background-image : url(bum.jpg) ;  
3   background-repeat: no-repeat;  
4   background-position: 300px 500px ;  
5 }
```

Background - bu umumiy, hamma xarakteristikalarini ishlatish uchun foydalaniladi, ya'ni **background-color**, **background-image**, **background-repeat**, **background-attachment**, **background-position** stil xususiyatlarini bitta shu stil orqali ishlatish mumkin, ya'ni

```
1 BODY {  
2   background-color:#ffee8c ;  
3   background-image : url(bum.png) ;  
4   background-repeat: no-repeat;  
5   background-attachment: fixed;  
6   background-position: top right;  
7 }
```

Shu uzun kodni, bitta "**background**" orqali yozish mumkin:

```
1 BODY {  
2   background: #ffee8c url(bum.png) no-repeat fixed top right ;  
3 }
```

Agar biror xususiyatini kiritmasak, doimiy qiymatni qabul qilib, ishlatadi.

№5 CSS dars - Shriftlar bilan ishlash: FONT, FONT-*

Maqola, shrift ko'rinishlari va matnlarni shriftini o'zgartirish haqidagi mavzuga bag'ishlanadi. CSS, shriftlar bilan qanday ishlay olishini ko'ramiz. Foydalanuvchiga biror matnni o'qish uchun qulay qilib berish, veb sahifa unumdorligini oshiradi, shuning uchun bu maqolani puxta o'zlashtirib oling.

Shriftlar deganda, yozuv qanday shriftlar oilasida yozilganligi, uning o'lchami, rangi va yana bir necha xususiyati tushuniladi. Shriftlarga quyidagi stil xususiyatlari javob bradi:

- **font-family;**
- **font-style;**
- **font-variant;**
- **font-weight;**
- **font-size;**
- **font.**

Bularning barchasini birma-bir ko'rib chiqamiz.

Font-family - shriftlar oilasi. Shriftlar ko'rinishiga qarab ma'lum bir oilalarga bo'linadi, asosiylari quyidagilar:

1. **Serif.** Quyidagi shriftlar bu oilaga tegishli: Times, Times New Roman, New York.

2. **Sans-serif.** Quyidagi shriftlar bu oilaga tegishli: Verdana, Geneva, Arial.

3. **Monospace.** Quyidagi shriftlar bu oilaga tegishli: Courier, Courier New.

Odatda, agar shrift nomida bo'sh joy bo'lsa, bu shrift qo'shtirnoqqa olinadi. Misol ko'ramiz,

```
1 <html>
2 <head>
3 <title>Shrift_1</title>
4 <style type="text/css">
5 .sh{
6 font-family: verdana, Arial, Helvetica;
7 }
8 </style>
9 </head>
10 <body>
11 <h3 class="sh">AkmX.uz - IT Site</h3>
12 </body>
13 </html>
```

AkmX.uz - IT Site

Nimaga bir necha shrift yozilmoqda? Sababi agar kompyuterda kerakli shrift bo'lmasa, brouzer ikkinchi turgan shriftni qidiradi va shu orqali yozuvni chiqaradi, misoldan, agar "**Verdana**" shrifti bo'lmasa, yozuv "**Arial**" shriftida yoziladi, agar u ham bo'lmasa, "**Helvetica**" shriftida. Shriftlar oilasi orqali ham berish mumkin:

```
1 p {font-family: "Times New Roman",
    serif;}
```


Font-style - bu xususiyat shriftga stil beradi va uch xil qiymatni qabul qilishi mumkin:

1. Normal - oddiy ko'rinish.
 2. Italic - biroz qiyshaygan ko'rinish.
 3. Oblique - bu ham qiyshaygan ko'rinish, lekin ma'lum bir burchakka, suniy yaratilgan.
- Barcha shriftlarning oddiy biroz qiyshaygan va qalin yozuvli ko'rinishi bo'ladi, qiyshaygan ko'rinishi bu "**Italic**", "**Oblique**" esa suniy ravishda yaratilgan qiyshayish.

```
1 <html>
2 <head>
3 <title>Shrift_2 </title>
4 <style type="text/css">
5 h1{
6 font-family: verdana, arial, sans-serif;
7 font-style:normal;
8 }
9 h2{
10 font-family: verdana, arial, sans-serif;
11 font-style:italic ;
12 }
13 .sun{
14 font-family: verdana, arial, sans-serif;
15 font-style:oblique;
16 }
17 </style>
18 </head>
19 <body>
20 <h1>Oddiy yozuv ko'rinishi</h1>
21 <h2>Biroz qiyshaygan ko'rinish</h2>
22 <h2 class="sun">Suniy qiyshaytirilgan
23 ko'rinish</h2>
24 </body>
</html>
```

Oddiy yozuv ko'rinishi

Biroz qiyshaygan ko'rinish

Suniy qiyshaytirilgan ko'rinish

Font-variant - bu stil xususiyati, kichik harflarni tanlangan variant asosida yozib beradi. 2 xil qiymati mavjud:

1. Normal - oddiy hol, yozuv odatiy ko'rinishda chiqadi(odatiy hol).
2. Small-caps - kichik harflar, kichiklashtirilgan katta harf ko'rinishida namoyon bo'ladi.

Misol ko'ramiz,

```
1 <html>
2 <head>
3 <title>Shrift_3 </title>
4 <style type="text/css">
5 h1{
6 font-family: verdana, arial, sans-serif;
7 font-variant:small-caps;
8 }
9 h2{font-family: verdana, arial, sans-
10 serif;}
11 </style>
12 </head>
13 <body>
14 <h1>Akmx.uz - it Site</h1>
15 <h2>Akmx.uz - it Site</h2>
16 </body>
17 </html>
```

Natija esa quyidagicha



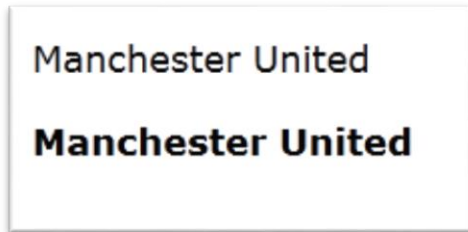
Font-weight - shrift qalinligini kattalashtirish uchun ishlatiladi. Kalit so'zli qiymatlari 2 ta:

1. Normal - oddiy ko'rinish.
2. Bold - qalinlashtirilgan ko'rinish.

Ba'zi brauzerlar raqamli qiymatlarni ham qabul qiladi: **100, 200, 300, 400, 500, 600, 700, 800, 900**. Ma'lumot uchun **400** qiymat "**normal**" bilan bir hil. Misol ko'ramiz,

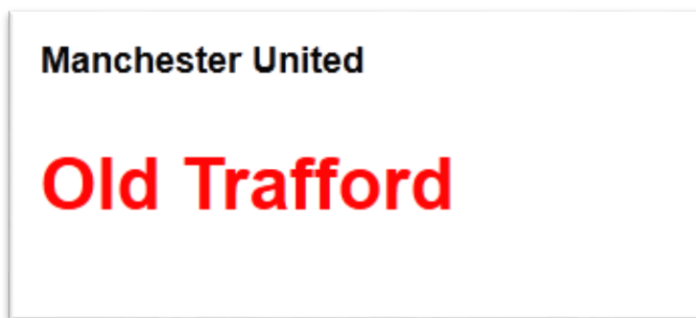
```
1 <html>
2 <head>
3 <title>Shrift_4 </title>
4 <style type="text/css">
5 #sh1{
6 font-weight: normal;
7 font-family: verdana;
8 }
9 #sh2{
10 font-family: verdana;
11 font-weight: bold;
12 }
```

```
13 </style>
14 </head>
15 <body>
16 <p id="sh1">Manchester United</p>
17 <p id="sh2">Manchester United</p>
18 </body>
19 </html>
```



Font-size - yozuv o'lchamini berish xususiyati, pikselda berishni maslahat beraman, shunda barcha brauzerlarda bir xil ko'rinadi.

```
1 <html>
2 <head>
3 <title>Shrift_5 </title>
4 <style type="text/css">
5 h1 {
6 font-family: arial, verdana, sans-serif;
7 font-size: 18px;
8 }
9 h2{
10 font-family: arial, verdana, sans-serif;
11 font-size: 36px;
12 color: red;
13 }
14 </style>
15 </head>
16 <body>
17 <h1>Manchester United</h1>
18 <h2>Old Trafford</h2>
19 </body>
20 </html>
```



Font - yuqoridagi barcha stil xususiyatlarini mana shu birgina xususiyat bilan amalga oshirish mumkin, biroz tushunarsiz bo'lishi mumkin(boshlovchilar uchun), lekin juda qulay hisoblanadi.

```
1 P{
2  font-style: italic;
3  font-variant: normal;
4  font-weight: bold;
5  font-size: 30px;
6  font-family: arial, sans-serif;
7 }
```

Shu CSS kodini quyidagi bilan almashtirib ishlatish mumkin:

```
1 P{ font : italic normal bold 30px
   arial,sans-serif; }
```

№6 CSS dars - Matnlarni formatlash: TEXT-*, LETTER-SPACING, WORD-SPACING, LINE-HEIGHT, WHITE-SPACE, DIRECTION, UNICODE-BIDI

Bu darsda matnlarni formatlashga oid CSS xususiyatlari bilan tanishib chiqamiz, ular unchalik qiyin emas va ishlatilishi ham tushunarli, aytmoqchimanki qandaydir qiyin mantiqiy narsalar yo'q. Bu xususiyatlarni yodlab olish shart emas, shunchaki shunday narsa borligini bilib qo'ying, kerak bo'lib qolsa, sintaksisini istalgan joydan topib olishingiz mumkin.

Umumiy olgandan, matnlar bilan ishlash uchun quyidagi xususiyatlar mavjud:

- **text-align;**
- **text-decoration;**
- **text-indent;**
- **text-transform;**
- **letter-spacing;**
- **word-spacing;**
- **line-height;**
- **white-space;**
- **direction;**
- **unicode-bidi.**

Endi, stillarga oid bu xususiyatlarni ko'rib chiqamiz.

Text-align - matnni tekislash, HTML dagi "**align**" attribut kabi ishlaydi, qiymatlari ham bir xil:

left - chap tomonga tekislash(odatiy hol);

right - o'ng tomonga tekislash;

center - markazga tekislash;

justify - matn uzunligi bo'yicha tekislash.

```
1 h1{text-align:center;}
```

Text-decoration - matnni chizish kerak bo'lsa, shu xususiyat ishlatiladi, uning qiymatlari quyidagilar:

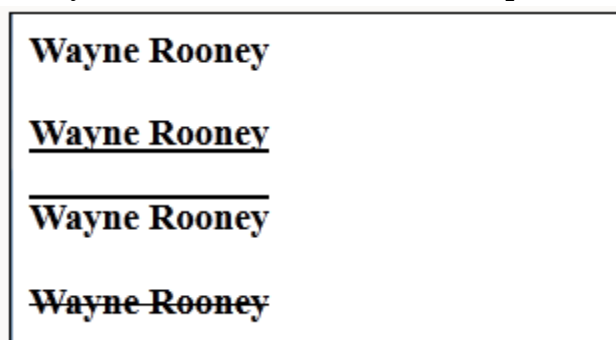
none - hech qanday chizishni amalga oshirmaydi;

underline - matnni tegiga chizish;

overline - matnni ustig chizish;
line-through - matnni chizib tashlash.
Barcha qiymatlarga misol ko'ramiz,

```
1 <html>
2 <head>
3 <title>decoration </title>
4 <style type="text/css">
5 #a1{
6 text-decoration:none;
7 }
8 #a2{
9 text-decoration:underline;
10 }
11 #a3{
12 text-decoration:overline;
13 }
14 #a4{
15 text-decoration:line-through;
16 }
17 </style>
18 </head>
19 <body>
20 <h3 id="a1">Wayne Rooney</h3>
21 <h3 id="a2">Wayne Rooney</h3>
22 <h3 id="a3">Wayne Rooney</h3>
23 <h3 id="a4">Wayne Rooney</h3>
24 </body>
25 </html>
```

Bitta yozuvni har xil ko'rinishda chiqardik, natijani ko'ramiz.



Text-indent - birinchi qatorning boshida joy qoldirish, ya'ni abzatsni o'rnatish uchun ishlatiladi. Qiymatni pikselda berish qulay va ishonchli.

```
1 <html>
2 <head>
3 <title>indent </title>
4 <style type="text/css">
5 p {
6 text-align:justify;
```

```

7  text-indent: 40px;
8  }
9  </style>
10 </head>
11 <body>
12 <p>Ko'pchilikka ma'lum holat, telefoningiz sizga SMS
13 kelganidan xabar beradi.
14 Kelgan xabarni o'qir ekansiz bu sizga kerak bo'lmagan
15 Reklama ekani ma'lum bo'ladi.
16 Eng g'azabingizni chiqaradigan holat, siz bu reklamaga
17 a'zo bo'lmagansiz va
   kelgan xabarni umuman qiziq va kerak emasligida.</p>
   </body>
   </html>

```

Ko'pchilikka ma'lum holat, telefoningiz sizga SMS kelganidan xabar beradi. Kelgan xabarni o'qir ekansiz bu sizga kerak bo'lmagan Reklama ekani ma'lum bo'ladi. Eng g'azabingizni chiqaradigan holat, siz bu reklamaga a'zo bo'lmagansiz va kelgan xabarni umuman qiziq va kerak emasligida.

Text-transform - matndagi yozuv harflarini katta-kichik qilib o'zgartirish stil xususiyati. 4 xildagi qiymatlarni qabul qiladi:

capitalize - matndagi har bir so'zning birinchi harfini katta harf qilib chiqaradi;

uppercase - matndagi barcha harflarni katta harfga almashtirib chiqaradi;

lowercase - matndagi barcha harflarni kichik harfga almashtirib chiqaradi;

none - registrni o'zgartirmaydi, ya'ni qanday yozilgan bo'lsa shunday chiqaradi.

```

<html>
<head>
1  <title>text </title>
2  </head>
3  <body>
4  <p style="text-transform:capitalize">Manchester
5  United Football club<p>
6  <p style="text-transform:uppercase">Manchester
7  United Football club<p>
8  <p style="text-transform:lowercase">Manchester
9  United Football club<p>
10 <p style="text-transform:none">Manchester United
11 Football club<p>
   </body>
   </html>

```

Manchester United Football Club

MANCHESTER UNITED FOOTBALL CLUB

manchester united football club

Manchester United Football club

Letter-spacing - matndagi harflar orasidagi masofani oʻrnatish, qiymatlarni pikselda beramiz.

```
1 <html>
2 <head>
3 <title>Letter </title>
4 </head>
5 <body>
6 <p style="letter-spacing:3px">Contana</p>
7 <p style="letter-spacing:8px">Contana</p>
8 </body>
9 </html>
```

Contana

C o n t a n a

Word-spacing - matndagi soʻzlar orasidagi masofani oʻrnatish, albatta pikselda qiymatlarni beramiz.

```
<html>
1 <head>
2 <title>Word </title>
3 </head>
4 <body>
5 <h3 style="word-spacing:3px">Roy
6 Kean</h3>
7 <h3 style="word-spacing:8px">Roy
8 Kean</h3>
9 </body>
</html>
```

Roy Kean

Roy Kean

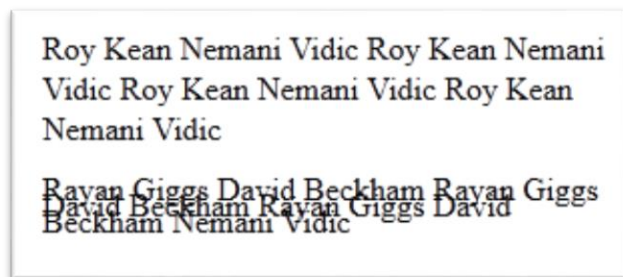
Line-height - qator orasidagi masofani oʻrnatish, pikselda qiymat kiritamiz.

```
1 <html>
```

```

2 <head>
3 <title>Line </title>
4 </head>
5 <body>
6 <p style="line-height:20px">Roy Kean Nemani Vidic Roy Kean Nemani Vidic
7 Nemani Vidic Roy Kean Nemani Vidic</p>
8 <p style="line-height:8px">Rayan Giggs David Beckham Rayan Giggs David
9 Giggs David Beckham Nemani Vidic</p>
</body>
</html>

```



Natijadan ko'rib turibsizki, oradagi masofa kichik bo'lsa, juda tushunarsiz matn hosil bo'lar ekan. Bu stilni "**normal**" degan qiymati ham bor, bu qiymat odatiy hisoblanadi.

White-space - so'zlar orasidagi probellarni qanday chiqarishni o'rnatish. Odatda HTML gipermatn tilida barcha probellar bitta probel ko'rinishida chiqadi, pre tegidan tashqari(pre tegi probellarni hisobga olib, shriftni ham sal o'zgartirib chiqaradi), "**white-space**" xususiyati "**pre**" tegi bajargan vazifani bajaradi, faqat shriftni o'zgartirmaydi. "**White-space**"ning qiymatlari:

normal - oddiy ko'rinish, matndagi probellar bitta probel ko'rinishida chiqadi va so'zlar keyingi qatorga avtomat o'tadi;

nowrap - matndagi probellar hisobga olinmaydi, so'zlar avtomat keyingi qatorga o'tmaydi, "br" tegi orqali keyingi qatorga o'tish kerak;

pre - matndagi probellar va keyingi qatorga o'tishlar hisobga olinadi, agar kodda keyingi qatorga o'tish bo'lmasa, brauzerda o'tkazgich(prokrutka) paydo bo'ladi;

pre-line - matndagi probellar hisobga olinmaydi, belgilangan chegaradan so'zlar chiqib ketsa, avtomat keyingi qatorga chiqadi;

pre-wrap - matndagi probellar va keyingi qatorga o'tishlar saqlanib qoladi, agar matn belgilangan joydan chiqib ketsa, avtomat keyingi qatorga o'tadi.

Misol ko'ramiz,

```

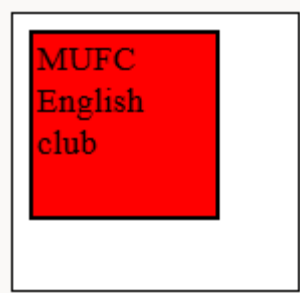
1 <html>
2   <head>
3     <title>Margin Padding</title>
4     <style>
5     .sloy1 {
6       border:2px solid black;
7       padding:2px;
8       height:80px;
9       width:80px;
10      white-space:pre-wrap;
        background-color:red;

```

```

11  }
12  </style>
13  </head>
14  <body>
15  <div>MUFC
16  English club
17  </div>
18  </body>
19  </html>
20

```



Yozuv matn ichiga sig'madi va keyingi qatorga o'tdi.

Direction - matnni chiqarish yo'nalishini o'rnatish, 2 xil qiymati mavjud:

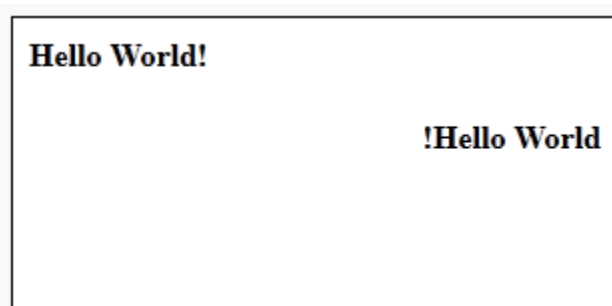
ltr - chapdan o'nga qarab chiqarish;

rtl - o'ngdan chapga qarab chiqarish.

```

<html>
1  <head>
2  <title>Direct </title>
3  </head>
4  <body>
5  <h4 style="direction:ltr">Hello
6  World!</h4>
7  <h4 style="direction:rtl">Hello
8  World!</h4>
9  </body>
</html>

```



Unicode-bdi - matnni qanday tartibda chiqarishni o'rnatish, qiymatlari:

normal - brauzer o'zi qanday chiqarishni hal qiladi;

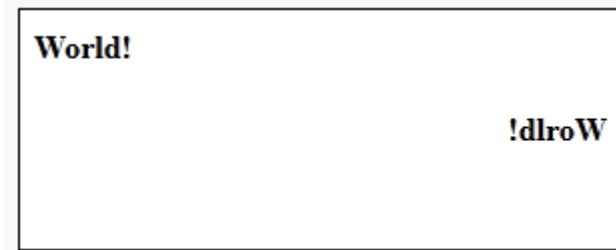
embed - "**direction**" xususiyatini qayta aniqlashtiradi, "**direction**"da qanday yozilgan bo'lsa shunaqa chiqadi;

bidi-override - biz uchun qiziqarli qiymat, simvollar ketma-ketligi tartibi o'zgartirib chiqaradi.


```

    <html>
1  <head>
2  <title>Rang </title>
3  </head>
4  <body>
5  <h4 style="direction:ltr; unicode-
6  bidi:embed">World!</h4>
7  <h4 style="direction:rtl; unicode-bidi:bid-
8  override">World!</h4>
9  </body>
    </html>

```



№7 CSS dars - Ro'yhatlar haqida: LIST-STYLE, LIST-STYLE-*

CSS stillar bilan ishlash tilida ham HTML gipermatn tiliga o'xshab ro'yxatlar tushunchasi mavjud. Ro'yxatlar uchun uncha ko'p stil xususiyatlari ajratilmagan, shu sababli maqolamiz qisqa bo'ladi, lekin kerakli xususiyatlar bilan tanishamiz. Ro'yxatlarni, CSS xususiyatlarini ishlatmasdan HTML gipermatn tilida ham hosil qilishingiz mumkin, lekin CSSdagidek chiroyli chiqmaydi, shuning uchun maslahat, ikkala hilni ham ko'rib chiqing va ma'qulidan foydalaning. Ro'yxatlar uchun quyidagi stil xususiyatlarini ishlatamiz:

- **list-style-type;**
- **list-style-position;**
- **list-style-image;**
- **list-style.**

Bu xususiyatlar universal bo'lib, tartiblangan va tartiblanmagan ro'yxatlar uchun ishlatilaveriladi. Bular orqali tartibli va tartibsiz ro'yxatlar hosil qilish mumkin, keling ular bilan tanishib chiqamiz.

List-style-type - ro'yxatlar uchun markerlarni aniqlashtirish uchun ishlatiladi, markerlar raqam, harf, kvadrat, dumaloq va boshqa ko'rinishlarda bo'lishi mumkin. Markerlarni aniqlashtirish uchun quyidagi qiymatlar ishlatiladi:

disk - marker sifatida bo'yalgan doira;

circle - marker sifatida bo'yalmagan aylana;

square- marker sifatida bo'yalgan kvadrat.

decimal - marker sifatida oddiy arab raqamlar: 1, 2, 3, 4, 5...

decimal-leading-zero - marker sifatida oddiy arab raqamlar "0" bilan: 01, 02, 03, 04, 05...

upper-roman - marker sifatida katta rim raqamlari: I, II, III, IV,...

lower-roman - marker sifatida kichik rim raqamlari: i, ii, iii, iv,...

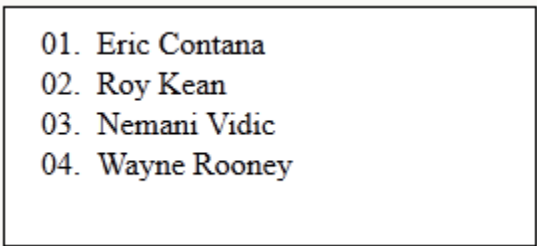
upper-alpha - marker sifatida katta ingliz harflari: A, B, C, D,...

lower-alpha - marker sifatida kichik ingliz harflari: a, b, c, d,...

none - ro'yxatni markersiz qilish.

YUqoridagilarga misol ko'ramiz,

```
1 <html>
2   <head>
3     <title>List</title>
4     <style>
5       ol {
6         list-style-type: decimal-leading-
7         zero;
8       }
9     </style>
10  </head>
11  <body>
12    <ol>
13      <li>Eric Contana</li>
14      <li>Roy Kean</li>
15      <li>Nemani Vidic</li>
16      <li>Wayne Rooney</li>
17    </ol>
18  </body>
19 </html>
```

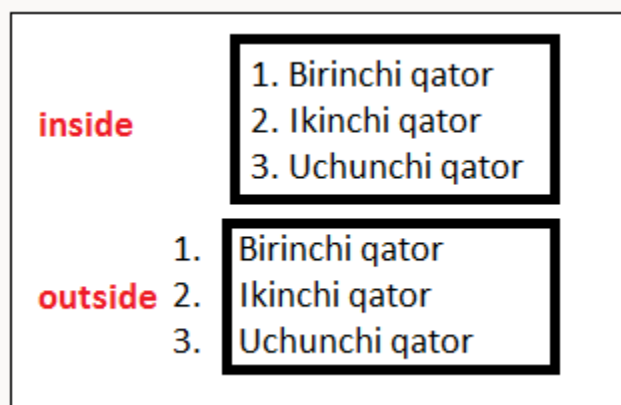


01. Eric Contana
02. Roy Kean
03. Nemani Vidic
04. Wayne Rooney

List-style-position - bu stil xususiyati markerni joylashishini aniqlashtiradi, 2 xil qiymat qabul qilishi mumkin:

outside - markerni ro'yxat bloki tashqarisiga joylashtirish;

inside - markerni ro'yxat bloki ichiga joylashtirish.



Bu stil eskirgan bo'lib, yangi brauzerlarda ishlamasligi mumkin.

List-style-image - ro'yxatning eng qiziq xususiyati bo'lib, markerga o'zingiz hohlagan rasmni qo'yishingiz mumkin bo'ladi. "**url**" kalit so'zidan keyin, rasm adresini yozasiz.

```
1 <html>
```

```

2   <head>
3     <title>List</title>
4     <style>
5       ol {
6         list-style-image: url(rasm.png);
7       }
8     </style>
9   </head>
10  <body>
11    <ol>
12      <li>Alex Fergusion</li>
13      <li>David Moyes</li>
14      <li>Lui Van Gal</li>
15    </ol>
16  </body>
17 </html>

```



List-style - yuqoridagi ro'yxatga oid barcha xususiyatlarni birlashtiruvchi xususiyat.

```

1  ul {
2    list-style-type:square;
3    list-style-position: inside;
4    list-style-image: url(rasm.png);
5  }

```

Shu 5 qatorli kodni, "**list-style**" orqali quyidagicha qisqartirish mumkin.

```

1  ul {list-style:square inside url(rasm.png)}

```

№8 CSS dars - Havolalar(ссылка) bilan ishlash: LINK, ACTIVE, HOVER, VISITED

Biror bir saytga kirsangiz, u yerdagi havola(ссылка)ga sichqonchani oborsanigiz, bu havola o'z ko'rinishini o'zgartirishini ko'rgan bo'lsangiz kerak, ya'ni tegiga chizadi yoki havola rangi o'zgaradi vahakazo. Bular aynan CSS orqali amalga oshiriladi, to'g'riroq aytadigan bo'lsak, psevdoklasslar bu ishni amalga oshirib beradi. Maqolamiz aynan havolalarga - psevdoklasslarga bag'ishlanadi.

Havolalar uchun quyidagi 4 ta psevdoklass ishlatiladi:

- **link** - havolaning doimiy ko'rinishi;
- **active** - havolaning aktiv holatdagi(bosilayotgandagi) ko'rinishi;

- **visited** - havolaga kirilgandagi holati;
- **hover** - sichqoncha havolaga olib borilgandagi ko'rinishi.

Psevdoklasslar quyidagicha ishlatiladi:

A: psevdoklass nomi {...stil...}

Keling biror misol ko'ramiz.

```

1  <html>
2    <head>
3      <title>Havola</title>
4      <style>
5        a:link{color:red;}
6        a:active{color:black;}
7        a:hover{color:blue;}
8        a:visited{color:yellow;}
9      </style>
10     </head>
11     <body>
12       <a href="#1">Doimiy havola qizil
13       rangda</a></br>
14       <a href="#2">Kirilgan havola
15       sariqda</a></br>
16       <a href="#3">sichqoncha oborilgandagi
17       havola ko'k</a></br>
18       <a href="#4">Aktiv havola qora</a></br>
19     </body>
20   </html>

```

Natijada hamma stil xususiyatlarini ishlatib chiqardim, ya'ni rang barang bo'lishi uchun

Doimiy havola qizil rangda
Kirilgan havola sariqda
sichqoncha oborilgandagi havola ko'k
Aktiv havola qora

Ranglardan tashqari boshqa attributlarni ham qo'llash mumkin:

```

1  a:link {
2    color: blue ;
3  }
4  a:visited {
5    color:gray;
6  }
7  a:hover {
8    color:red ;
9    text-decoration:none;
10   font-weight:bold;
11 }
12 a:active {

```

```
12 color:green;
13 text-decoration:none;
14 text-transform:uppercase;
15 }
16
```

Agar, havolaning barcha holatlari bir xil bo'lishini hohlasangiz, quyidagicha yozishingiz mumkin.

```
1 a:link, a:visited, a:hover, a:active {color:black;
  text-decoration:none;}
```

YUqoridagi holatni psevdoklassiz ham amalga oshirish mumkin.

```
1 a {color:black; text-decoration:none;}
```

Xullas, havolalarning tashqi ko'rinishi CSS orqali boshqarilur ekan.

№9 CSS dars: BLOK tushunchasi haqida

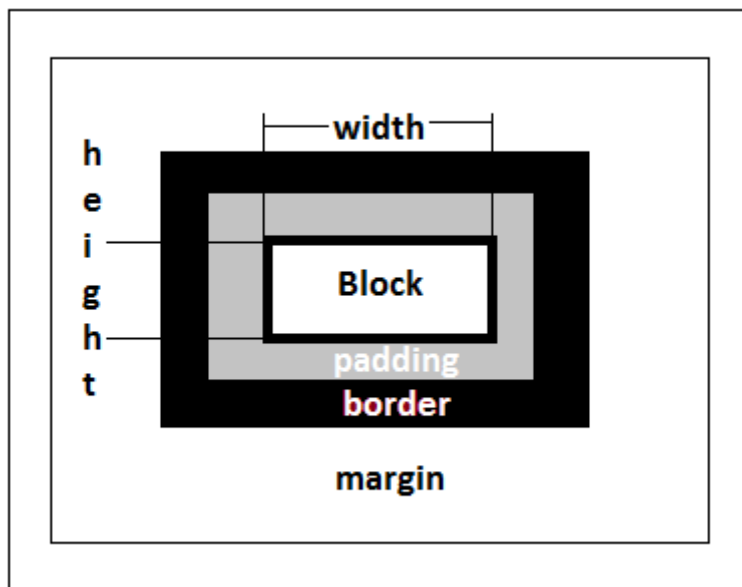
Web sohasida, blok tushunchasi juda kerakli termin hisoblanadi, saytning barcha brauzerlarda bir xilda ochilishi, rasm va yozuvlarni o'ylaganingizdek namoyon bo'lishi aynan blok holatiga bog'liq. Bu maqoladan boshlab, blokga tegishli stillar xususiyatini ko'rib chiqamiz. Dastlab, umumiy blok nimaligini bilib olamiz.

Blok bu - ma'lum bir chegaraga ega bo'lgan va alohida qism sifatida qaraladigan ma'lum bir axborotni o'zida saqlaydigan joydir.

HTML gipermatn tilini oladigan bo'lsak, u yerda ham blok va qator elementlari tushunchasi bor. Blokli elementlar, alohida strukturaga ega bo'lib, to'rtburchak chegaraga bilan o'ralgan bo'ladi. Ikkita blokli elementni bir qatorga joylashtirib bo'lmaydi. HTML gipermatn tilida blokli elementga misol qilib, "**H1...H6**", "**P**" va "**DIV**" teglarini olish mumkin. Har safar bu teglarni ishlatganda, brauzer har biri uchun alohida qator ajratib beradi.

Qatorli elementlarga esa, "**strong**", "**em**", "**i**" kabi teglarni misol qilish mumkin. Bu teglardan bir qatorda bimalol foydalansa bo'ladi.

CSS stillar texnologiyasida ham blok tushunchasi mavjud va u ham o'z strukturasi va to'rtburchak chegarasiga ega. Strukturali blok ichida qatorli blok ham bo'lishi mumkin(blok ichidagi yozuvlar), qatorli blok yozuv ko'rinishida ham bo'lishi mumkin, to'g'riroq etadigan bo'lsak, strukturali blok ichida ob'yektli blok bo'lishi mumkin. YUqoridagi nazariy ma'lumotlarni quyida bitta rasm orqali tushunish mumkin.



Bloklarning o'z xususiyatlari bor, misol uchun, blok uzunligi(width), blok balandligi(height), chegarasi(border),...

Har bir blok qandaydir tashkil etuvchilardan iborat bo'lishi shart, ya'ni yozuv, rasm yoki boshqa axborot. Blokning bu qismi **kontent** deb yuritiladi. Misol uchun "**P**" tegi uchun kontent yozuv hisoblanadi.

Blok atrofida qandaydir boshqa blok yoki ob'yektlar bo'lishi mumkin, blokdan ulargacha bo'lgan masofaga "**margin**" kalit so'zi javob beradi.

Blok ajralib turishi uchun, uning chegarasi bo'lishi lozim, bu chegaraga "**border**" javob beradi.

YUqoridagilardan tashqari blokning ichki bo'sh oralig'i bo'lishi mumkin, ya'ni blokdan blok chegarasigacha bo'lgan masofa, bu masofaga "**padding**" kalit so'zi javob beradi.

Keyingi maqolalarda blokning shu xususiyatlarini birma-bir ko'rib chiqamiz, bu maqolani blok tushunchasiga ta'rif sifatida qabul qilasiz.

№10 CSS dars - Blok chegaralari: **BORDER**, **BORDER-***, **OUTLINE**, **OUTLINE-***

Bloklar haqida oldingi maqolada gaplashib o'tdik, endi shu blok xususiyatlarini ko'rib chiqamiz. Bu maqolamiz blok chegaralariga bag'ishlanadi, blok bo'lgandan keyin uning albatta chegarasi ham bo'lishi kerakku.

Chegaralar blokni boshqa ob'yektlardan ajratib turadi va veb sahifada chiroyli namoyon bo'lishini ta'minlaydi. Demak, maqolada blok chegaralariga oid quyidagi CSS xususiyatlarini ko'rib chiqamiz.

- **border-width;**
- **border-color;**
- **border-style;**
- **border;**
- **outline;**
- **outline-color;**
- **outline-style;**
- **outline-width;**

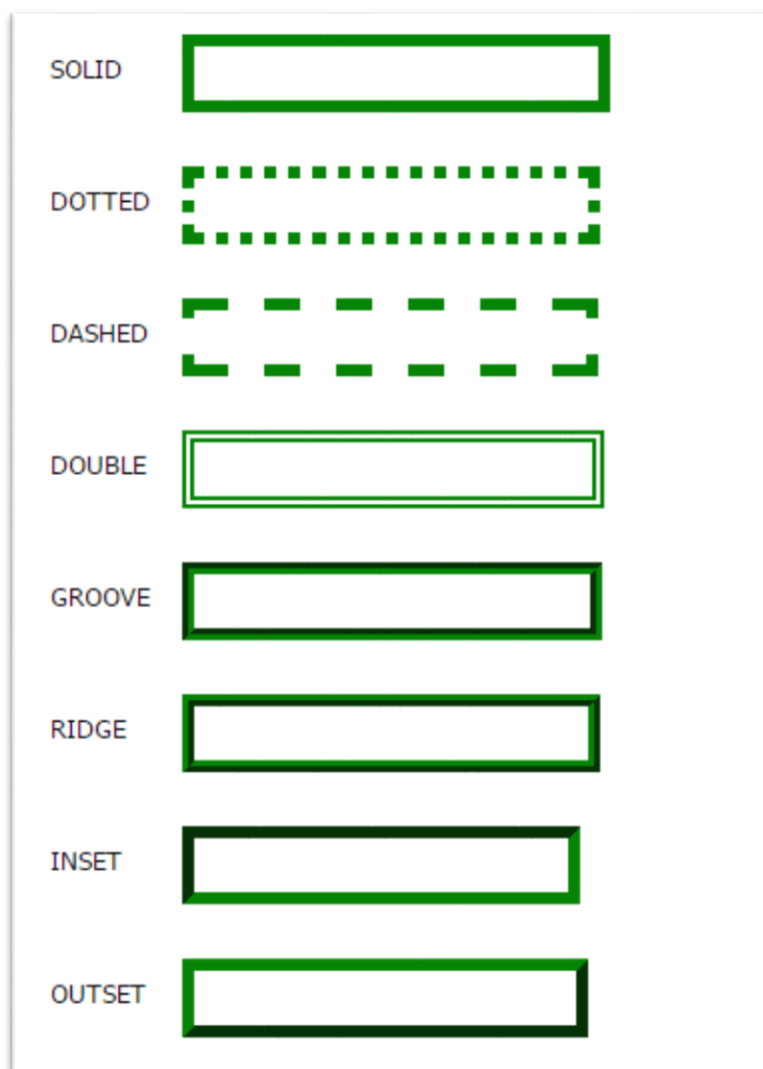
Ko'rib turganingizdek, unchalik ko'p emas, keling hammasiga batafsil to'xtaylik.

Border-width - bu xususiyat ramka(chegara) qalinligini o'rnatish uchun ishlatiladi. Odatda qiymat pikselda beriladi, undan tashqari kalit so'zlar orqali ham berish mumkin, **thin**(2px), **medium**(4px) yoki **thick**(6px). Bu o'lchamlar qanchaligini quyidagi rasmdan ko'rish mumkin:



Border-color - nomida aytib turibdiki, ramkaga rang beradi, rang nomi yoki rangning 16 lik sanoq tizimidagi kodi orqali berish mumkin.

Border-style - bu xususiyat ramka qanday stilda bo'lishini belgilaydi. Quyida ramka stillarini ko'rishingiz mumkin.



Asosiy ramka turlarini ko'rib chiqdik, bitta narsa, "**Double**" stil ramkasi uchun ramka qalinligi minimal **3px** bo'lishi lozim, aks holda kutgan natijangizni ololmasligingiz mumkin.

YUqoridagi 3 ta stil xususiyatiga misol ko'ramiz.

1 <html>

```

2   <head>
3     <title>Border</title>
4     <style>
5   h1 {
6     border-width: 5px;
7     border-style: inset;
8     border-color: green;
9   }
10  h2 {
11    border-width: 18px;
12    border-style: double;
13    border-color: red;
14  }
15  p {
16    border-width: 3px;
17    border-style: solid;
18    border-color: blue;
19  }
20  </style>
21  </head>
22  <body>
23    <p>Manchester United Football Club</p>
24    <h1>Manchester United Football Club</h1>
25    <h2>Manchester United Football Club</h2>
26  </body>
27  </html>

```

Manchester United Football Club

Manchester United Football Club

Manchester United Football Club

Border - yuqoridagi ramka xususiyatlarini bitta shu kalit soʻz orqali qisqa yozish mumkin. Bu asosan professionallar uchundir, boshlovchilarga har birini alohida yozishni maslahat beraman.

Yuqoridagi misolni, bitta "**border**" orqali yozamiz.

```

1  h1 {border: 5px inset green;
2  }
3  h2 {border 18px double red;

```



```

4  }
5  p {border 3px solidblue;
6  }

```

Barcha ramkaga oid xususiyatlarga quyidagi kalit soʻzlarni qoʻshish mumkin:

top - yuqori qism(border-top);

right - oʻng tomon(border-right);

bottom - quyi qism(border-bottom);

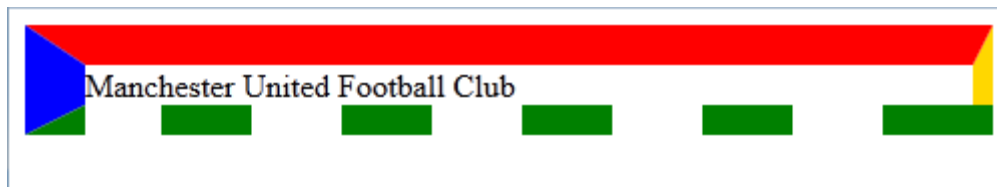
left - chap tomon(border-left).

Bu kalit soʻzlar orqali ramkaning istalgan tomonini oʻzgartirish mumkin, misol koʻramiz.

```

1
2  <html>
3    <head>
4      <title>Border</title>
5      <style>
6  p {
7    border-top-width: 20px;
8    border-top-style:solid ;
9    border-top-color: red;
10   border-right-width: 10px;
11   border-right-style:dashed ;
12   border-right-color: gold;
13   border-bottom-width: 15px;
14   border-bottom-style:dashed;
15   border-bottom-color: green;
16   border-left-width: 30px;
17   border-left-style:solid ;
18   border-left-color: blue;
19 }
20 </style>
21 </head>
22 <body>
23 <p>Manchester United Football Club</p>
24 </body>
25 </html>

```



Qisqartirilgan holda,

```

1  p {
2    border-top: 20px solid red;
3    border-right: 10px dashed gold;
4    border-bottom: 15px dashed green;
5    border-left: 30px solid blue;

```

6 }

YUqoridagi ramka berishlardan tashqari, yana tashqi ramka degan tushuncha majud, ya'ni blokdan tashqari tomonda ramka beradi, buning uchun CSSNing "**outline**" xususiyatidan foydalaniladi.

Outline-color - blokning tashqi tomoni(tashqi ramka)ga rang berish.

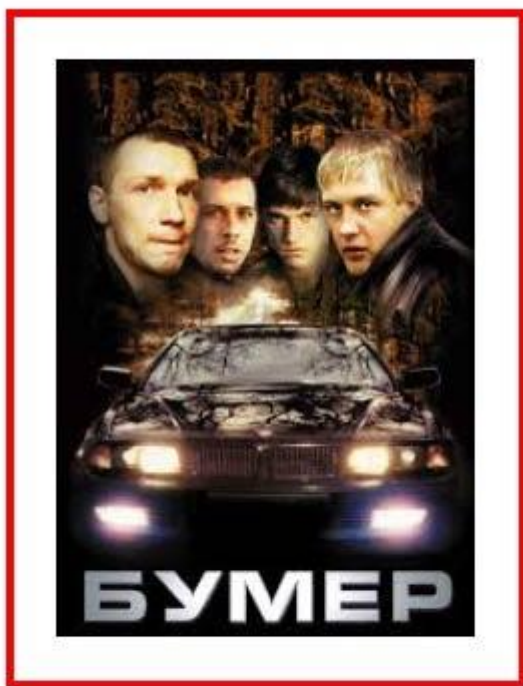
Outline-style - tashqi ramkaga stil berish. Stil turlari "**border-style**" stil turlari bilan bir xil.

Outline-width - tashqi ramka qalinligini o'rnatish. "**border-width**" kabi qiymatlarga ega(**thin**, **medium**, **thick** yoki piksellarda).

Shu uchchala stil xususiyatlariga misol ko'ramiz.

```
1
2   <html>
3     <head>
4       <title>Outline</title>
5       <style>
6   .out {
7     padding:20px;
8     color:red;
9     margin:10px;
10    outline-color:red;
11    outline-width:4px;
12    outline-style:solid;
13  }
14    </style>
15  </head>
16  <body>
17    
18  </body>
19  </html>
```

Natija, rasm atrofida 4 piksel qalinlikdagi, qizil rangli oddiy ramkadan iborat.



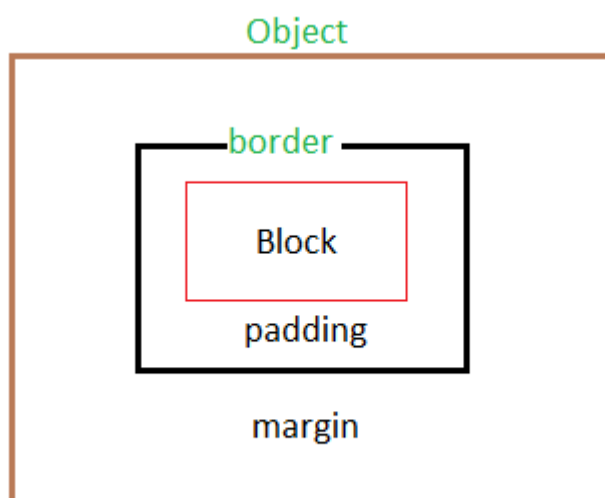
YUqoridagi kodni "**outline**" orqali bir qatorda yozish mumkin:

```
1 .out{outline:4px solid red;}
```

natija o'sha holda qoladi.

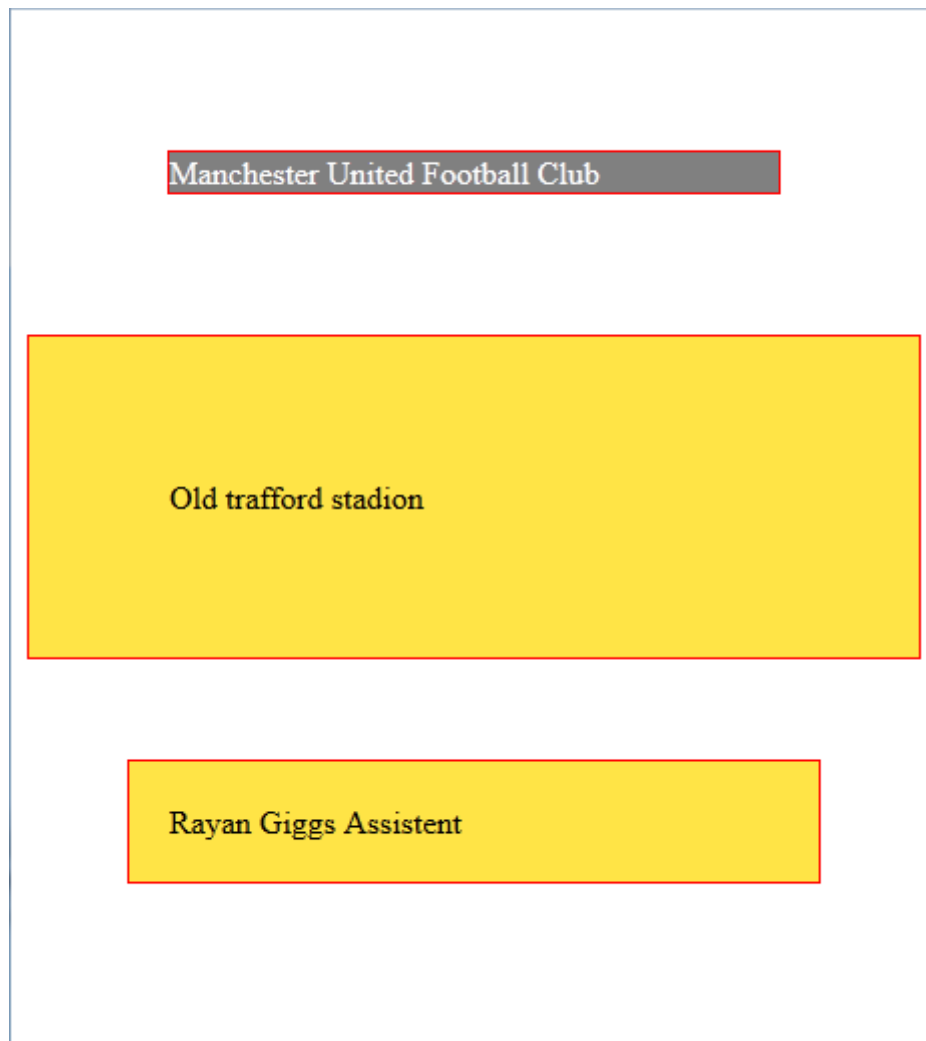
№11 CSS dars - Blokning oraliq masofalari: MARGIN, PADDING

Bu maqolamiz blokning ichki va tashqi oraliq masofalariga bag'ishlanadi, juda kerakli va muhim dars deb hisoblasak bo'ladi. Blokning o'z ramkasi(chegarasi)gacha bo'lgan oraliq masofa va ramkasidan atrofidagi biror ob'yektgacha bo'lgan masofa kabi xususiyati bor. Bu ikki oraliq masofani tushinish uchun quyidagi rasmni keltirib o'taman. Bu yerda blok sifatida biror blokli teg(p, div,...) bo'lishi mumkin.



Rasmdan ko‘rinib turibdiki, blokdan o‘z ramkasigacha bo‘lgan masofa "**padding**" deb, blok ramkasidan biror ob'yekt(element) yoki element mavjud bo‘lmasa hujjat oxirigacha bo‘lgan masofa "**margin**" deb yuritilar ekan. Bu ikki xususiyatni kod ichida ishlatish, blokni veb sahifada aniq va tushunarli bo‘lishini ta'minlaydi. Misol ko‘ramiz va hammasi tushunarli bo‘ladi

```
1  <html>
2    <head>
3      <title>Margin Padding</title>
4      <style>
5  .p1 {
6    background-color : gray;
7    border:1px solid red;
8    margin:70px;
9    color:white;
10   }
11  .p2 {
12    background-color : #FFE446;
13    border:1px solid red;
14    padding:70px;
15   }
16  .p3 {
17    background-color : #FFE446;
18    border:1px solid red;
19    margin:50px;
20    padding:20px;
21   }
22   </style>
23   </head>
24   <body>
25   <p class="p1">Manchester United Football
26   Club</p>
27   <p>Old trafford stadion</p>
28   <p class="p3">Rayan Giggs Assistent</p>
29   </body>
30   </html>
```



Oldingi maqolada ishlatilgan quyidagi kalit soʻzlar ham bu xususiyat uchun ishlaydi:

-left - chap tomondan joy qoldirish(**margin-left, padding-left**)

-right - oʻng tomondan joy qoldirish(**margin-right, padding-right**)

-top - yuqori qismdan joy qoldirish(**margin-top, padding-top**)

-bottom - pastki qismdan joy qoldirish(**margin-bottom, padding-bottom**).

Endi, kalit soʻzlar orqali boʻsh joy qoldirishga oid misolni koʻramiz.

```
1 <html>
2 <head>
3   <title>Margin Padding</title>
4   <style>
5   .d1 {
6     background-color : gray;
7     border:1px solid red;
8     margin-top:70px;
9     margin-bottom:10px;
10    color:white;
11  }
12  .d2 {
13    background-color : #FFE446;
14    border:1px solid red;
15    padding-left:70px;
```


```

16 padding-right:10px;
17 }
18 </style>
19 </head>
20 <body>
21 <p>Eric Contana</p>
22 <p>Roy Kean</p>
23 </body>
24 </html>

```



Eric Contana



Roy Kean

YUqoridagi CSS kodni qisqartirib quyidagicha yozish mumkin:

```

1 .dl{margin: 70px 0 10px 0;}
2 .dl{padding: 0 10px 0 70px;}

```

Ko'rinib turibdiki yozilish ketma-ketligi soat yo'nalishi bo'yichadir, ya'ni: yuqori qism(**top**), o'ng tomon(**right**), pastki qism(**bottom**), chap tomon(**left**).

Qaysi paytda qaysi birini(margin, padding) ishlatishni, o'zingiz hal qilishingiz kerak, faqat shuni aytishim mumkinki, "**padding**" blok ichida bo'ladi, "**margin**" blok tashqarisida.

№12 CSS dars - Blok o'lchamlari: WIDTH, HEIGHT

Blokga oid xususiyatlarni ko'rib chiqishni davom ettiramiz, bu safar yana bir asosiy xususiyatlardan biri, blokning balandligi(**height**) va uning uzunligi(**width**) haqida gaplashamiz. Agar sezgan bo'lsangiz, blokning uzunligi va balandligi avtomat aniqlashtiriladi, yozuv qancha katta bo'lsa(yoki blokni boshqa tashkil etuvchi), shuncha blokning uzunlik va balandlik katta bo'ladi.

Shunday bo'lsa ham, biz CSS orqali blokning uzunlik va balandligini o'zimiz o'rnatishimiz mumkin. Demak bu ish uchun biz quyidagi ikkita xususiyatdan foydalanamiz:

- **height** - blokning balandligi;
- **width** - blokning uzunligi.

Odatda, CSS da blok siftida "**div**" tegi ishlatiladi, lekin paragraf(p tegi), ro'yxatlarni ham blok sifatida ishlatish mumkin.

Nazariy ma'lumotlar bilan tanishib chiqdik, endi misol ko'ramiz.

```

1 <html>
2 <head>
3 <title>WH</title>
4 <style>
5 .mul {
6 width: 100px;

```

```
7  border: 1px solid red;
8  background: #FFE446;
9  }
10 .mu2 {
11  height: 70px;
12  border: 1px solid red;
13  background: #FFE446;
14  }
15 .mu3 {
16  width: 75px;
17  height: 120px;
18  border: 1px solid red;
19  background: #FFE446;
20  }
21 .mu4 {
22  width: 30px;
23  height: 30px;
24  border: 1px solid red;
25  background: #FFE446;
26  }
27  </style>
28  </head>
29  <body>
30  <p class="mu1">England champion football club - Manchester
31  United football club!</p>
32  <p class="mu2">Eric Contana</p>
33  <p class="mu3">Wayne Rooney</p>
34  <p class="mu4">Phile John</p>
35  </body>
    </html>
```

England
champion
football club -
Manchester
United football
club!

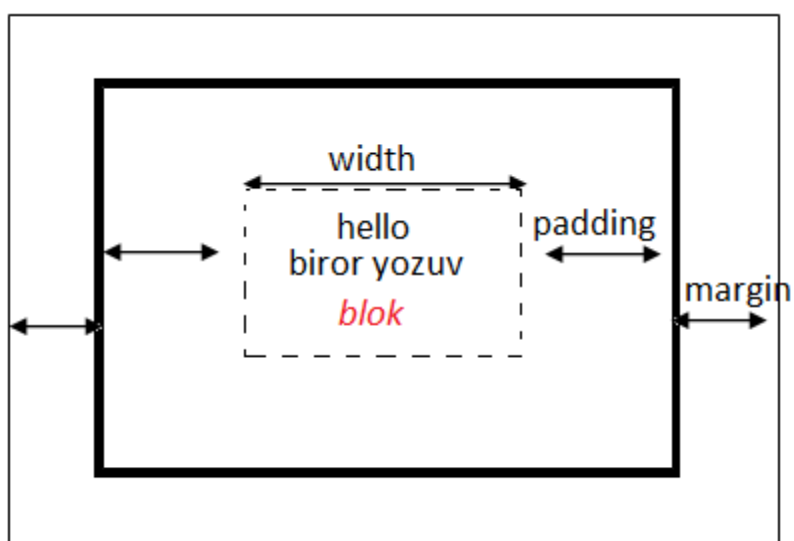
Eric Contana

Wayne
Rooney

Phile
John

Har xil o'lchamdagi bloklar hosil qildik, agar biz yozgan yozuv blok ichiga sig'masa, ko'rib turganingizdek blok tashqarisida ham yozuvning davomi yozilaverar ekan. Shuning uchun bloklar hosil qilishda hamma narsalarni hisobga oling.

Blok uzunligi va balandligini berish bilan siz, faqat yozuv uchun ajratilgan joyni aniqlashtirgan bo'lasiz, oraliq masofalarni belgilovchi CSS xususiyatlar(**margin,padding**) bu o'lchamlarga kirmaydi. Siz kiritgan o'lchamdan keyin, o'rnatgan oraliq masofa qiymatichalik joy tashlab ketiladi, rasmda ko'radigan bo'lsak,



YUqoridagi o'lcham beruvchi CSS xususiyatlardan tashqari yana quyidagi o'lchov beruvchi xususiyatlar mavjud:

- max-width - blokning maksimal uzunligi;

- min-width - blokning minimal uzunligi;
- max-height - blokning maksimal balandligi;
- min-height - blokning minimal balandligi.

Bu xususiyatlar blokning maksimal va minimal o'lchovlarini aniqlashtirib beradi, biror kod orqali bu maksimal va minimal o'lchovlardan chetga chiqib bo'lmaydi. Beriladigan o'lchovlar faqat shu oraliqga tushishi lozim. Misol,

```
1 .size {
2   height:80px;
3   width:600px;
4   max-width: 300px;
5   border:1px solid red;
6   background-color:orange;
7 }
```

Bu holda blok uzunligi 300 piksel bo'ladi, biz o'rnatgan 600 piksel ishlamaydi. Menimcha hammasi tushunarli, tushunarsiz narsalar pochta yoki komentariyada ko'rib chiqiladi.

№13 CSS дарс - Блокларни саҳифа бўйлаб жойлаштириш: POSITION

CSS бизга исталаган объектни, браузернинг исталган жойида чиқариш имконини беради. Бунинг учун шу блок координаталарини кўрсатишимиз керак ҳолос. Блок деганда "**div**" тегидан ташқари рўйхатлар, параграфлар, заголовкалар, расмлар, умумий қилиб айтганда, блокли модел кўринишидаги барча теглар тушунилади. Координаталарни ўрнатиб, блокларни жойлаштириш **позициялаш** (жойлашиниш) дейилади. Мақола айнан блокларни жойлашувига бағишланади. Жойлаштириш 2 хил кўринишда бўлади ва бу хил кўринишлар учун қуйидаги CSS хусусиятларини ишлатамиз.

1. Абсолют жойлаштириш(position:absolute;);
2. Бирор нарсага нисбатан жойлаштириш(position:relative;).

Абсолют жойлаштириш. Жойлаштиришда, одатда ушбу тур жойлаштириш ишлатилади. CSSда жойлаштирув вазифасини "**Position**" калит сўзи амалга ошириб беради, ушбу жойлаштирувни абсолют қилиш учун эса "**absolute**" киймати берилади. Ундан сўнг қуйидаги калит сўзлар орқали координаталар берилади:

- top;
- left;
- right;
- bottom.

Бу калит сўзлар, браузерда қандай намоён бўлишини қуйидаги расмда кўриш мумкин.



Абсолют жойлаштирувнинг яхши томони шундаки, блок айнан сиз берган координаталарда жойлашади, ўзи олдиндан аниклаштириб қўйилган жой(html кодда берилган жой) эса автомат ўчади. Унинг ўрнида бошқа нарсалар жойлашиши мумкин. Мисол кўрамиз, чунки сўз билан тушунтириш бироз кийинроқ. Дастлаб, оддий расм қўйиш мисолини кўрамиз.

```
1 <html>
2 <head>
3 <title>Position</title>
4 </head>
5 <body>
6 
7 <h1>Bumer I filmidagi aktyorlar va
8 mashina!</h1>
9 </body>
</html>
```



Bumer I filmidagi aktyorlar va mashina!

Энди шу мисолдаги расмни координата бўйича жойлаштирамиз.

```
1  <html>
2  <head>
3  <title>Position</title>
4  <style>
5  .smile{
6  border: 1px solid red;
7  position:absolute;
8  top:100px;
9  right:100px;
10 }
11 </style>
12 </head>
13 <body>
14 
15 <h1>Bumer I filmidagi aktyorlar va
16 mashina!</h1>
17 </body>
18 </html>
```

Bumer I filmidagi aktyorlar va mashina!



Эътибор берган бўлсангиз, расм ўрнатилган координатага кўчди ва расм ўрнида биз ёзган ёзув пайдо бўлди. Расм координатаси, тепадан 100 пиксел ва ўнг томондан 100 пиксел координатада жойлашди.

Одатда координата фақат иккита қийматда берилади:

```
1 .mu{
2   position:absolute;
3   bottom:300px;
4   left:100px;
5 }
```

ёки

```
1 .mu{
2   position:absolute;
3   top:100px;
4   right:100px;
5 }
```

Блок браузернинг бирор бурчагига нисбатан олинади, шу сабабли иккита координата қиймати берилади.

Агар, блок браузерга ёпишиб туришини хоҳласангиз, яъни браузер катталиги ўзгарганда айлантиргич(прокрутка) чиқса ҳам, блок айланмасдан қотиб туриш керак бўлса, "**absolute**" қиймат ўрнига "**fixed**" қиймат бериш лозим.

```
1 .mu{
2   position:fixed;
3   top:100px;
4   right:100px;
5 }
```

Бу икки қийматни мисоллар орқали, ишлатиб кўрин.

2. Бирор нарсага нисбатан жойлаштириш. Бу тур жойлаштиришда, блокнинг дастлабки ҳолатидан бошлаб, координата ҳисобланади(абсолют жойлаштириш каби браузернинг юқори қисмидан эмас). Ундан ташқари блокнинг дастлабки ҳолати ўзгаришсиз сақлаб қолинади, одатда бўш жой бўлиб кўриниб туради. Бу жойлаштиришни амалга ошириш учун, "**position**" хусусиятига "**relative**" қиймати берилади. Юқоридаги мисолни "**relative**" орқали ёзамиз.

```
1 <html>
2 <head>
3 <title>Position</title>
4 <style>
5 .smile{
6 border: 1px solid red;
7 position:relative;
8 top:100px;
9 right:10px;
10 }
11 </style>
12 </head>
13 <body>
14 
15 <h1>Bumer I filmidagi aktyorlar!</h1>
16 </body>
17 </html>
```



Мисолдан кўриниб турибдики, расм теги биринчи ёзилган, демак расм браузернинг чап юқори қисмида жойлашиши керак, унинг тегида эса ёзув. Биз координатани берганимиздан сўнг, расм бошланғич ҳолатидан берилган қиймат бўйича сурилди ва дастлабки ҳолати бўш бўлиб қолди. Ёзув ҳам унинг жойига ўтмади, абсолют жойлаштириш бўлганида, ёзув расмнинг олдинги ҳолатига ўтар эди.

№14 CSS дарс - Блокларни жойлаштириш(2 - қисм): FLOAT, CLEAR

Браузерда блокларни тўғри жойлаштириш, жуда муҳим саналади, чунки қулай жойлаштирилган блоклар фойдаланувчиларга сайтдан фойдаланишга қулайликлар туғдиради. Нимага фақат блоклар ҳақида гаплашиоқдамиз? чунки ҳозирги кунда кўп сайтларнинг асоси айнан блоклардир, яъни сайтларнинг структуралари блоклар орқали яратилмоқда. Бундай структурали сайтларга рақобат қилиб, жадвал асосида яратилган сайтларни келтириш мумкин. Уларнинг қайси бири яхшилиги хусусида кўп тортишилар бўлган ва бўлмоқда ҳам. Бу масалани бошқа мақолаларда кўриб ўтамиз.

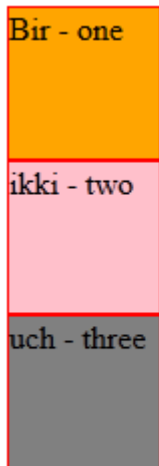
Демак, блокларни веб саҳифа бўйлаб жойлаштириш(бундай блокларни яна сузувчи блоклар(плавающие блоки) деб ҳам аташади), CSSнинг **"float"** хусусияти асосида амалга оширилади. Бу хусусият уч хил қийматни қабул қилиши мумкин:

- **left** - структурали блок чап томонга сурилади.
- **right** - структурали блок ўнг томонга сурилади.
- **none** - блок сурилмайди.

Келинг дастлаб сурилишларсиз оддий мисол кўрамиз, яъни **"float"** ишлатмаймиз.

```
1  <html>
2  <head>
3    <title>Float</title>
4    <style>
5  .box1 {
6    width:75px;
7    height:75px;
8    border:1px solid red;
9    background-color:orange;
10 }
11 .box2 {
12 width:75px;
13 height:75px;
14 border:1px solid red;
15 background-color:pink;
16 }
17 .box3 {
18 width:75px;
19 height:75px;
20 border:1px solid red;
21 background-color:gray;
22 }
23 </style>
24 </head>
25 <body>
26 <div class="box1">Bir - one</div>
27 <div class="box2">ikki - two</div>
28 <div class="box3">uch - three</div>
29 </body>
```

30 </html>



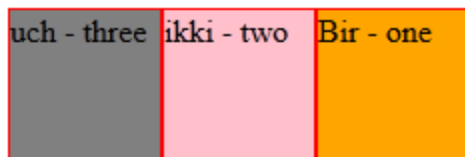
3 та блок одатий ҳолда жойлашди, яъни чап томондан бошлаб, ҳар бир блок пастка қараб кетма-кет ўрнатилди. Агар "**float**" бермасангиз шу кўринишда жойлашув амалга ошади. Энди "**float**" орқали ўнг томондан бошлаб жойлаштирамиз.

```
1   <html>
2   <head>
3     <title>Fload</title>
4     <style>
5     .box1 {
6     width:75px;
7     height:75px;
8     border:1px solid red;
9     background-color:orange;
10    float:right;
11  }
12  .box2 {
13  width:75px;
14  height:75px;
15  border:1px solid red;
16  background-color:pink;
17  float:right;
18  }
19  .box3 {
20  width:75px;
21  height:75px;
22  border:1px solid red;
23  background-color:gray;
24  float:right;
25  }
26  </style>
27  </head>
28  <body>
```

```

29 <div class="box1">Bir - one</div>
30 <div class="box2">ikki - two</div>
31 <div class="box3">uch - three</div>
32 </body>
33 </html>

```

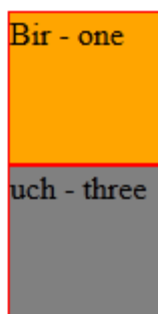
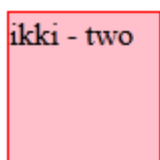


Агар, 2-блокдан сурилишни олиб ташласак, у алохида бўлиб қолади ва ўзининг доимий жойлашувида жойлашади. Юқоридаги код, фақат 2 блок учун **"float:right"** қаторини олиб ташлаймиз.

```

1  .box3 {
2    width:75px;
3    height:75px;
4    border:1px solid red;
5    background-color:gray;
6  }

```



Бу усул расмларга таъриф беришда фойдалидир, яъни расм, унинг ёнида бу расмга таъриф.

```

1  <html>
2    <head>
3      <title>Float</title>
4      <style>
5    .box1 {
6      width:150px;
7      height:150px;
8      color:black bold;

```

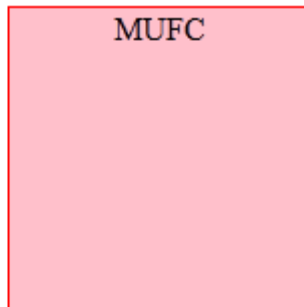


```

9   text-align:center;
10  border:1px solid red;
11  background-color:pink;
12  float:right;
13  }
14  </style>
15  </head>
16  <body>
17  <div class="box1">MUFC</div>
18  <div>Manchester United Football klubi 1878 yilda tashkil topgan bo'l
19  MANCHESTER UNITED nomini olgan</div>
20  </body>
    </html>

```

Manchester United Football
klubi 1878 yilda tashkil
topgan bo'lib, 1902 yilda
MANCHESTER UNITED
nomini olgan



"float" хакида гапирилганда, албатта **"clear"** хусусиятини ҳам эслаб ўтиш лозим. Бу хусусият берилган қиймат асосида, қуйи қисм орқали жойлаштиришни амалга оширади, биров тушунарсиз, лекин мисолда тушуниб олишингиз мумкин, қийматлари:

- left - блок барча чап томондаги блоklarнинг қуйи қисмида жойлашади;
- right - блок барча ўнг томондаги блоklarнинг қуйи қисмида жойлашади;
- both - блок барча блоklarнинг қуйи қисмида жойлашади;
- none - қуйи қисмда жойлаштириш олиб ташланади.

Дастлаб, **"clear"** хусусиятисиз мисол қўрамыз.

```

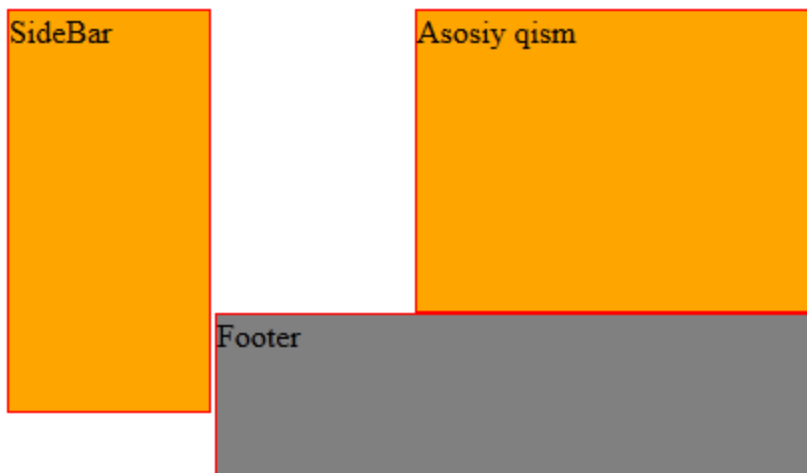
1   <html>
2   <head>
3   <title>Float</title>
4   <style>
5   .box1 {
6   width:100px;
7   height:200px;
8   border:1px solid red;
9   background-color:orange;
10  float:left;
11  }
12  .box2 {
13  width:200px;
14  height:150px;
    border:1px solid red;

```

```

15 background-color:orange;
16 float:right;
17 }
18 .box3 {
19 width:300px;
20 height:80px;
21 border:1px solid red;
22 background-color:gray;
23 float:right;
24 }
25 </style>
26 </head>
27 <body>
28 <div class="box1">SideBar</div>
29 <div class="box2">Asosiy qism</div>
30 <div class="box3">Footer</div>
31 </body>
32 </html>
33

```



"Footer" ёзувли блокга эътибор беринг, бу блок энг пастда ҳамма блокларнинг тегида жойлашиши керак эди, буни **"clear"** билан амалга оширамиз. Унинг учун **"box3"** стилини ўзгартирамиз ҳолос.

```

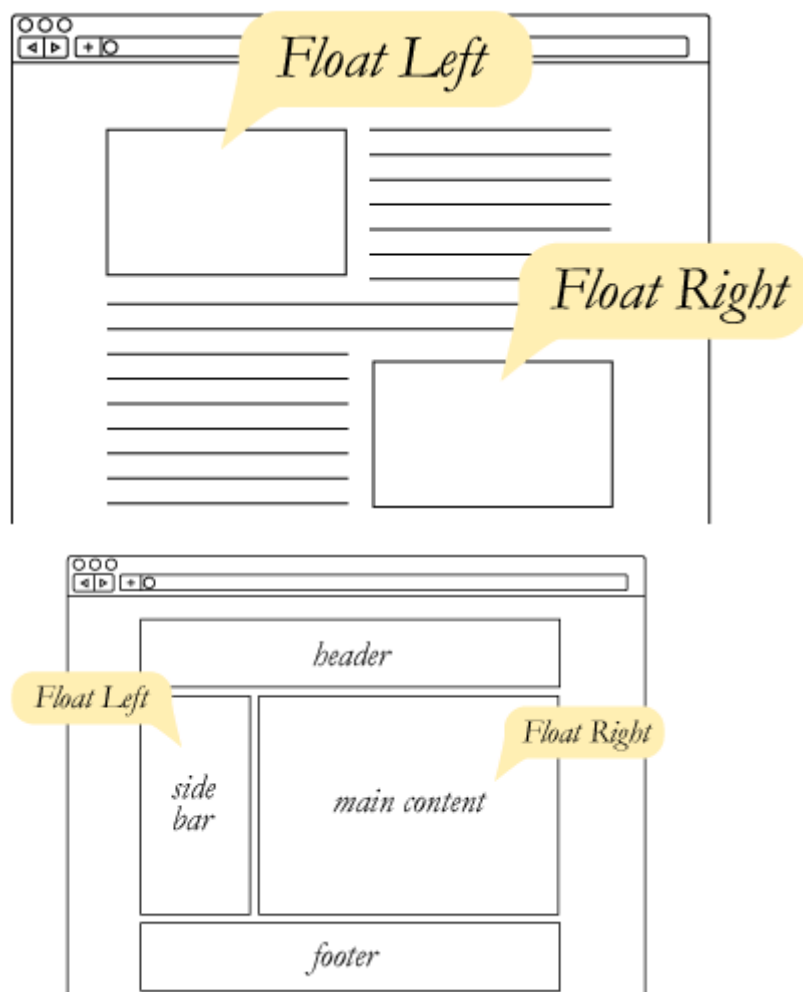
1 .box3 {
2 width:300px;
3 height:80px;
4 border:1px solid red;
5 background-color:gray;
6 clear:both;
7 }

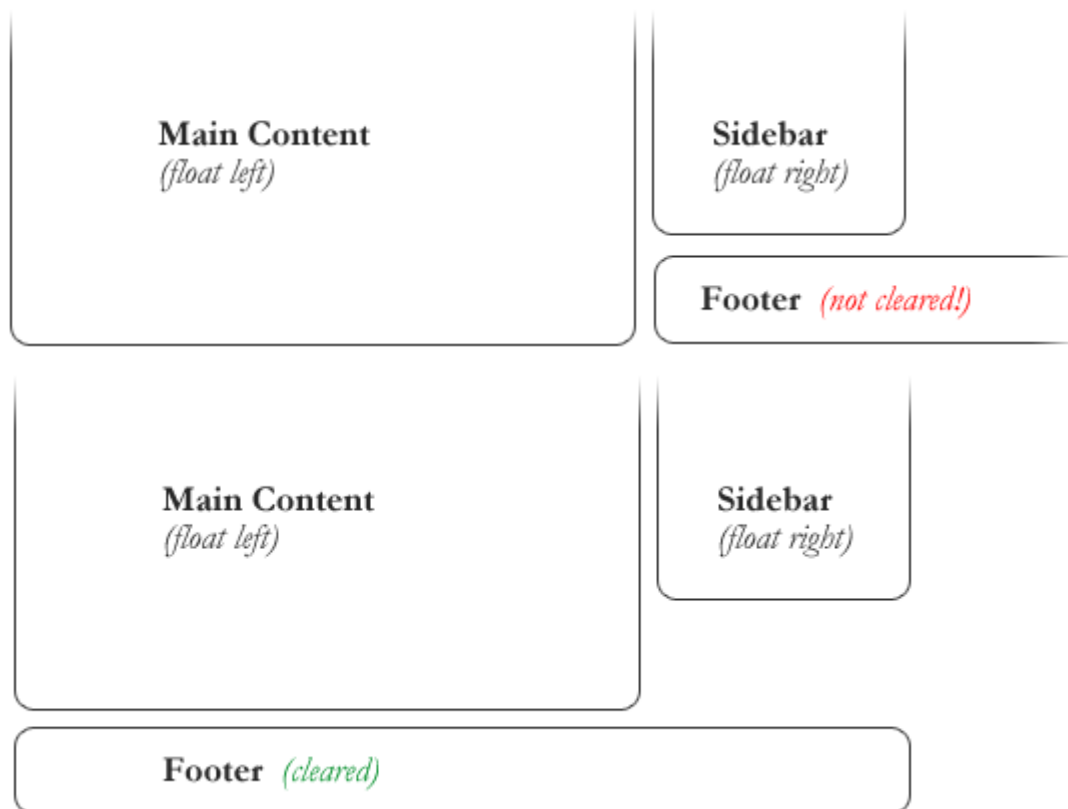
```

Натижа эса, биз кутган кўринишни берди.



Бу мавзу бироз тушунарсиз бўлганлиги сабабли, яхшироқ тушуниб олишингиз учун **css-tricks.ru** сайтидан олинган куйидаги расмларни кўрсатаман. Бу расмларда сайтнинг структурасини **"float"** орқали ҳосил қилиш кўрсатилган.





№15 CSS дарс - Блок ва контенларни жойлаштириш(3 - қисм): CLIP, VISIBILITY, OVERFLOW, VERTICAL-ALIGN

Блоклар жойлашуви мавзусини давом эттираман, навбатдаги CSS хусусиятимиз **"clip"** деб номланади ва бу хусусият блок ичидаги контентни, қанча қисми намоён бўлишини аниқлаштириб беради. Бу хусусият ичига координата нуқталари берилади ва шу нуқталар орасида ётган контент қисми кўринади ҳолос.

Контент фақатгина тўртбурчак кўринишда кесиб олиниши мумкин ва бериладиган координата ўқлари қуйидаги кетма-кетликда берилади: **Y1, X1, Y2, X2**. Бу координата ўқлари олдидан **"rect"** деган калит сўз ишлатилади. Тушунарли бўлиши учун мисол кўрамыз.

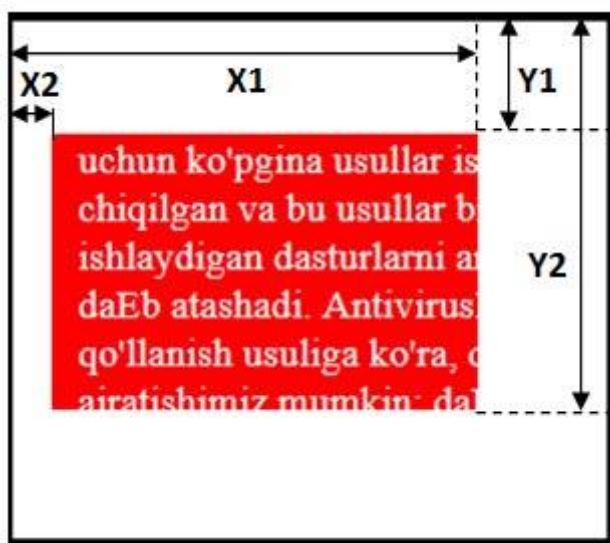
```
1 <html>
2   <head>
3     <title>Clip</title>
4     <style>
5       .clip{
6         position: absolute;
7         clip: rect(40px, 180px, 150px, 10px);
8         width: 250px;
9         color: white;
10        background: red;
11        border: 1px solid blue;
12        padding: 20px;
13      }
14    </style>
```

```

14 </head>
15 <body>
16 <div class="clip"> Hozirgi vaqtda viruslarni
17 yo'qotish
18 uchun ko'pgina usullar ishlab chiqilgan va
19 bu usullar bilan ishlaydigan dasturlarni
20 antiviruslar deb atashadi.
21 Antiviruslarni, qo'llanish usuliga ko'ra,
22 quyidagilarga ajratishimiz
23 mumkin: detektorlar, faglar, vaktsinalar,
    privivkalar, revizorlar.</div>
    </body>
    </html>

```

Натижа тушунарли бўлиши учун, координаталарни кўрсатиб ўтаман.



Навбатдаги CSS хусусиятимиз блокларни кўринарли ва кўринмас ҳолга келтиради, яъни **"visibility"** хусусияти. Бу амални **"display"** орқали ҳам амалга ошириш мумкин, лекин бу иккисини фарқи бор. **"Visibility"** орқали блокни кўринмас ҳолга олиб келса, блок учун ажратилган жойни ҳеч ким ишлата олмайди, яъни блок жойи сақланиб қолади, **"display"** хусусиятида эса блок жойи сақланмайди (display хақида кейинроқ). Бу хусусиятни қуйидаги асосий қийматлари мвжуд:

visible - блокни кўринарли ҳолга олиб келади;

hidden - блокни кўринмас ҳолга олиб келади;

Мисол кўрамыз,

```

1 <html>
2 <head>
3   <title>Visibility</title>
4   <style>
5     .vis{
6       height:150px;
7       width:200px;
8       background:black;
        color:white;


```

```

9     border:1px solid gray;
10    visibility:hidden;
11    }
12    </style>
13    </head>
14    <body>
15    <div class="vis">Boxing Day!</div>
16    <div>Boxing Day!</div>
17    </body>
18    </html>
19


```

Натижада, "**vis**" класи билан ёзилган ёзув кўринмайди, лекин **150x200** ўлчамдаги жой бўш қолдирилади ва ундан кейин кейинги элементлар чиқарилади.



Boxing Day!

Агар, "**visibility:visible**" қилиб блокни кўринтирсак, қуйидаги кўринишда бўлади.



Boxing Day!

Boxing Day!

Давом этамиз, блокларни ҳосил қилдик, уни ичини тўлатдик ҳам дейлик, агар блок ичидаги контент блок ичига сиғмаса нима бўлади? бу ҳолни ҳам CSS ўйлаб қўйган. Айнан контент блок ичига сиғмаса нима қилиш кераклигини "**overflow**" хусусияти орқали бериб қўйишимиз мумкин, унинг қуйидаши қийматлари мавжуд:

visible - контент блок ичига сиғмаса ҳам тўлиқ кўрсатиш;

hidden - контентнинг фақат блок ичига сиғган қисми кўринади ҳолос;

scroll - ҳар доим ўтказгич(прокрутка) кўрсатиш;

auto - ўтказгич фақат керак бўлганда пайдо бўлади.

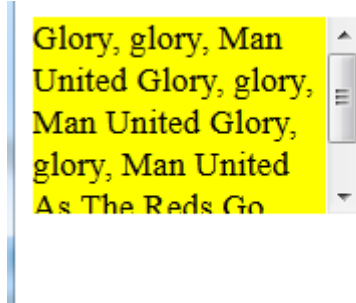
Энг яхшиси, "**auto**" қиймати бўлса керак, яна ўзиз биласиз, демак мисол кўрамиз:

```

1  <html>
2    <head>
3      <title>Over</title>
4      <style>
5        .over{
6            height:90px;
7            width:150px;
8            background:yellow;
9            overflow:auto;
10       }
11    </style>
12  </head>
13  <body>
14    <div class="over">
15    Glory, glory, Man United
16    Glory, glory, Man United
17    Glory, glory, Man United
18    As The Reds Go Marching On On On
19  </div>
20  </body>
21  </html>

```

Қийматдан кўриниб турибдики(auto), керак бўлса ўтказгич пайдо бўлади, бизнинг мисолда керак.



Кейинги CSS хусусиятимиз, вертикал бўйлаб жойлаштириш деб номланади, яъни **"vertical-align"**. Элементни вертикал бўйлаб текислаш ёки шу элементни исталганча юқори ва пастга ўрнатиш мумкин. Қуйидаги қийматларни қабул қилади:

baseline - элементни вертикал бўйлаб асосий чизикга ўрнатиш(одатий ҳол);

bottom - энг қуйида турган элемент билан бир чизикга жойлаштириш;

middle - элементни вертикал бўйлаб асосий чизик марказидан бошлаб ўрнатиш;

sub - қуйи индекс кўринишида жойлаштириш;

super - юқори ендекс кўринишида жойлаштириш;

text-bottom - элементнинг энг қуйи чиқиғи бўйлаб жойлаштириш;

text-top - элементнинг энг юқори чизиги бўйлаб жойлаштириш;

top - энг юқори турган элемент билан бир чизикга жойлаштириш;

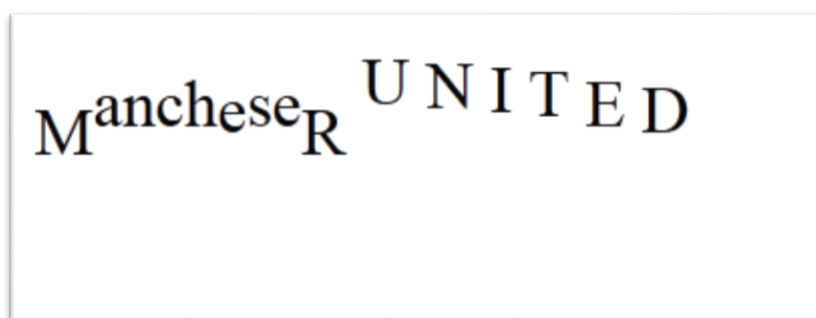
Бу қиматлардан ташқари, фоизлар, пикселлар кўринишида ҳам қиймат бериш мумкин, менинг фикримча пикселлар орқали берганингиз қулайроқ. Мисол кўрамиз,

```

1 <html>
2   <head>
3     <title>ver_align</title>
4   </head>
5   <body>
6     <div style="font-size:30px;">
7       <span style="vertical-align:sub">M</span>anch<span style="vertical-align:middle;">e</span>se<span style="vertical-align:bottom;">R</span>
8       <span style="vertical-align:10px">U</span>
9       <span style="vertical-align:8px">N</span>
10      <span style="vertical-align:6px">I</span>
11      <span style="vertical-align:4px">T</span>
12      <span style="vertical-align:2">E</span>
13      <span style="vertical-align:middle">D</span>
14    </div>
15  </body>
16 </html>

```

Натижаси чиройли кўринишга эга бўлади:



№16 CSS дарс - Слойлар билан ишлаш: Z-INDEX

Бу мақолада **CSSда слой** тушунчасини кўриб чиқамиз. Агар Photoshop билан ишлаган бўлсангиз, слой нималигини яхши билишингиз мумкин. **Слой** бу - қобик, яъни алоҳида алоҳида блокдир. Агар 3 та блок бўлса ва уларнинг ўлчамлари бир-хил бўлса ва сиз уларни устма-уст жойласангиз, энг қуйи слой(блок) кўринмайди. Бунда, устма-уст жойлашув бўлиб қолади, лекин слой мавжуд бўлиб тураверади.

Блокларни жойлаштиришда, биз аниқ координаталарни берамиз, кўп ҳолларда "x" ва "y" нуқталарни, ёки бошланғич ҳолатдан сурилишларни(**position**). Бу мақолада яна бир нуқта - "z" ни киритиб, ҳар бир блокни устма-уст жойлашувини аниқлаштирамиз. Слойларни бошқаришни CSSнинг "**z-index**" хусусияти орқали амалга оширамиз.

Блокнинг координаталари билан бирга уларнинг жойлашувини рақамлаймиз. Шу номер орқали қайси слой энг юқорида ва қайсиниси энг қуйида бўлишини кўрсатамиз. Мисол кўрамиз, дастлаб блокларни рақамламаймиз ва натижани кўрамиз.

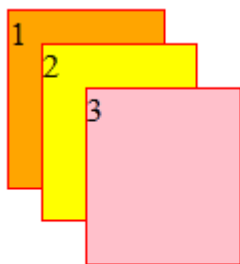
```

1 <html>
2   <head>

```

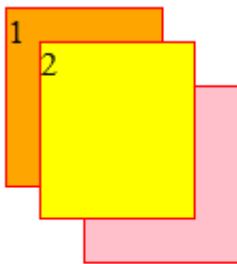


```
3     <title>Sloy</title>
4     <style>
5     .sloy1 {
6     position:absolute;
7     left:15px;
8     top:15px;
9     height:80px;
10    width:70px;
11    border:1px solid red;
12    background-color:orange;
13    }
14    .sloy2 {
15    position:absolute;
16    left:30px;
17    top:30px;
18    height:80px;
19    width:70px;
20    border:1px solid red;
21    background-color:yellow;
22    }
23    .sloy3 {
24    position:absolute;
25    left:50px;
26    top:50px;
27    height:80px;
28    width:70px;
29    border:1px solid red;
30    background-color:pink;
31    }
32    </style>
33    </head>
34    <body>
35    <div class="sloy1">1</div>
36    <div class="sloy2">2</div>
37    <div class="sloy3">3</div>
38    </body>
39    </html>
```



Код натижасидан кўриниб турибдики, қайси слой охирида ёзилса, ўша блок энг юқорида намоён бўлади. Энди, юқоридаги коднинг CSS қисмини ўзгартирамиз, яъни қўшамиз ва қуйидагига эга бўламиз.

```
1  .sloy1 {
2  position:absolute;
3  left:15px;
4  top:15px;
5  height:80px;
6  width:70px;
7  border:1px solid red;
8  background-color:orange;
9  z-index:2;
10 }
11 .sloy2 {
12 position:absolute;
13 left:30px;
14 top:30px;
15 height:80px;
16 width:70px;
17 border:1px solid red;
18 background-color:yellow;
19 z-index:3;
20 }
21 .sloy3 {
22 position:absolute;
23 left:50px;
24 top:50px;
25 height:80px;
26 width:70px;
27 border:1px solid red;
28 background-color:pink;
29 z-index:1;
30 }
```



Қайси блокнинг рақами(z-index:3)катта бўлса, ўша блок энг юқорида кўринади, мисолда иккинчи блокнинг рақами энг катта.

Яхшилаб ўйлаб кўрилса, бу хусусият билан анча чиройли ишларни амалга ошириш мумкин бўлади, чиройли саҳифалар, соя шаклидаги расмлар, тушувчи менюлар ва ҳаказо.

№17 CSS дарс - Динамик контентлар ҳосил қилиш ва курсор турлари: CONTENT, COUNTER-*, QUOTES, CURSOR

Агар сиз барча CSS мақолаларни ўқиб бораётган бўлсангиз, ҳозирга келиб анча нарсаларни ўрганиб олган бўласиз. Бу мақола ҳам олдингиларидан қолишмаган ҳолда муҳим саналиб, контентларга бағишланади.**Контент** бу - бирор блокнинг ташкил этувчиси ёки бир мавзунини ўзида йиғган бирор матндир, умумий ҳолда айтадигна бўлсак, сайтнинг ташкил этувчиларидир.

Қуйида контентлар билан ишлаш учун мўлжалланган CSS хусусиятлари билан танишишингиз мумкин:

- **quotes;**
- **counter-increment;**
- **counter-reset;**
- **content.**

Кўриб турганингиздек, унчалик кўп эмас, лекин уларнинг қийматлари кам ҳам эмас :). Бирма-бир кўриб чиқамиз:

Quotes - матн(контент) ичида қўштирноқ турини бериш. Ҳар хил стандартларда, матн ичида келадиган қўштирноқларга маълум бир талаблар қўйилади, айнан қўштирноқ тури "**quotes**" орқали веб саҳифага чиқарилади. Қўштирноққа олиниш керак бўлган ёзув "**q**"(quotes нинг қисқартирилаган фойрмаси)теги ичига ёзилади ва бу орқали олдиндан аниқлаштирилган қўштирноқ саҳифага чиқарилади, бу хусусиятнинг қуйидаги қийматлари бор:

Ko'rinish	HTML kod	Unicode
"	"	\0022
'	'	\0027
«	« или «	\00ab
»	» или »	\00bb
‘	‘	\2018
,’	’	\2019
“	“	\201c
”	”	\201d
„	„	\201e

Агар ҳам очиладиган, ҳам ёпиладиган қўштирноқ керак бўлса, иккалови кетма-кет берилади. Шу қийматлардан фойдаланиб мисол кўрамыз,

```

1  <html>
2  <head>
3  <style>
4  q{
5  font-size:18px;
6  quotes: "\00ab" "\00bb";
7  }
8  </style>
9  </head>
10 <body>
11 <p>United <q>Man United</q></p>
12     </body>
13 </html>
```

Counter-increment, counter-reset - бу хусусиятлар ҳисоблагичлар яратишда ишлатилади, яъни рўйхатлар яратишда кетма-кет рақамлашларни амалга ошириб беради. **Counter-reset** орқали, ҳисоблагич ҳисобини сақлаш учун бирор ном(идентификатор) берилади ва **counter-increment** орқали бу идентификаторни ўзгартириб борилади.

Ҳисоблагич идентификатори, **Counter-reset** орқали қуйидагича эълон қилинади(асосан body теги учун яратилади):

```
1 <style>
2 body{
3 counter-reset:akmx;
4 }
5 </style>
```

Бу ҳолда ҳисоб "1" дан бошланади, агар "0" дан бошлаш керак бўлса,

```
1 counter-reset:akmx -1;
```

Агар, ҳисоблагич маълум бир рақамдан бошланиш керак бўлса(мисол учун 5 дан),

```
1 counter-reset:akmx 5;
```

Идентификаторни эълон қилишни кўриб чиқдик, қолган комбинацияларни ўзингиз бироз ўйлаб амалга ошира оласиз.

Идентификаторни биттага ошириш учун, **counter-increment** хусусиятидан фойдаланилади:

```
1 counter-increment:akmx;
```

Агар фақат жуфт сонлар кўринишида ошиш керак бўлса,

```
1 counter-increment:akmx 2;
```

Бу хусусиятган мисол кўриш учун, **"content"** хусусиятидан хабардор бўлиш лозим, ҳисоблагич қийматларини браузерга чиқариш учун, **"content"**нинг **"counter"** қиймати керак бўлади. Бироз олдида ўтган ҳолда айтишим мумкинки, **"counter"** орқали керакли жойга ҳисоблагич қийматини чиқариш мумкин. Бундан ташқари, контентнинг қаеридан, ҳисоблагич қийматини чиқаришни аниқлаштириш учун, **"content"** хусусиятининг **"before"**(контентдан олдин) ёки **"after"**(контентдан кейин) қийматлари ҳам керак бўлади. Мисол кўрамиз,


```
1 <html>
2 <head>
3 <style>
4 body{
5 counter-reset:akmx;
6 }
7 h3:after{
8 counter-increment:akmx;
9 content: counter(akmx) ". ";
10 }
</style>
```

```

11 </head>
12 <body>
13 <h3>Eric Contana</h3>
14 <h3>Roy Kean</h3>
15 <h3>Rio Ferdinand</h3>
16     </body>
17 </html>
18

```

Мисолни таҳлил қилайдган бўлсак, **"akmx"** деган ҳисоблагич идентификатори яратилди(1 дан бошланади), унинг қиймати ҳар сафар биттага оширилиб борилади, **"h3:before"** яъни, **"h3"** тегидан олдин ҳисоблагич қиймати қўйиб борилади(ҳисоблагич қийматидан сўнг нуқта қўйилади), натижа



```

1. Eric Contana
2. Roy Kean
3. Rio Ferdinand

```

Мисолдан ҳаммаси тушунарли бўлди деб ўйлайман. Энди, **"content"** хусусиятини тўлиқ кўриб чиқамиз.

Content - веб саҳифа матнида мавжуд бўлмаган маълум бир қийматларни, шу CSS хусусияти орқали генерация қилиниб, матнга киритилади. Мисол қилиб, юқоридаги кодни кўрсатишимиз мумкин, тартиб рақам матнда мавжуд эмас эди, код орқали киритилди. Бу хусусиятни қуйидаги қийматлари мавжуд:

- open-quote - **"quotes"** хусусияти орқали очилувчи қўштирнокни ўрнатиш;
- close-quote - **"quotes"** хусусияти орқали ёпилувчи қўштирнокни ўрнатиш;
- no-open-quote - очилувчи қўштирнокни ўрнатишни бекор қилиш;
- no-close-quote - ёпилувчи қўштирнокни ўрнатишни бекор қилиш;
- counter - ҳисоблагич қийматини броузерга чиқариш;
- normal - матннинг қаерида генерация қилинган қийматларни чиқариш кераклигини аниқлаштириш, иккита псевдокласс қиймати бор: **":before"**(матндан олдин), **":after"**(матндан кейин).

Мисол кўрамиз,

```

1 <html>
2 <head>
3 <style>
4   .cont {
5     color: green;
6     quotes: "<" ">";
7   }
8   .cont:before {
9     content: open-quote;

```

```

10     }
11     .cont:after {
12         content: close-quote;
13     }</style>
14 </head>
15 <body>
16 <p>Manchester United o'yinchilari uchun</br> <b><span class="cont">2
17 2015</span> </b>yil og'ir o'tmoqda</p>
18     </body>
19 </html>

```

Ажратилган контент учун, ўрнатилган қўштирноқ контент боши ва охирида намоён бўлмоқда.

Manchester United o'yinchilari uchun
<2014-2015> yil og'ir o'tmoqda

Контент ичига исталган матн(сўз, бирор ёзув)ни ҳам киритишингиз мумкин:

```

















1 <html>
2 <head>
3 <style>
4     .kk:after{
5         content: " Best of the Best";
6     }
7 </style>
8 </head>
9 <body>
10 <p> <span class="kk">Manchester
11 United</span></p>
12 </body>
13 </html>

```

Энг асосийси, контент дан кейин ёки олдин(before, after) керакли ёзувни чиқаришни ўрнатиш.

Manchester United Best of the Best

Мақола охирида, яна бир CSS хусусиятини кўрсатиб ўтмоқчиман, унинг номи "**cursor**" деб номланади. Бу хусусият, сичқонча белгиланган объект устига олиб борилганда унинг кўрсаткичини ўзгартириб беради. Қуйидаги расмда бу хусусиятнинг қийматлари расм кўринишида келтирилган:

	Qiymat		CSS kodi
	default		P {cursor: default}
	crosshair		P {cursor: crosshair}
	help		P {cursor: help}
	move		P {cursor: move}
	pointer		P {cursor: pointer}
	progress		P {cursor: progress}
	text		P {cursor: text}
	wait		P {cursor: wait}
	n-resize		P {cursor: n-resize}
	ne-resize		P {cursor: ne-resize}
	e-resize		P {cursor: e-resize}
	se-resize		P {cursor: se-resize}
	s-resize		P {cursor: s-resize}
	sw-resize		P {cursor: sw-resize}
	w-resize		P {cursor: w-resize}
	nw-resize		P {cursor: nw-resize}

Юқоридаги қийматлардан ташқари, ўзингиз бирор расмни сичқонча кўрсаткичи сифатида ўрнатишингиз ҳам мумкин. Бирор мисол кўрамиз,

```

1 <html>
2 <head>
3 </head>
4 <body>
5 <p>Har xil ko'rinishdagi sichqoncha
6 ko'rsatkichlari:</p>
7 <a href="#" style="cursor:help">HELP</a> </br>
8 <a href="#"
9 style="cursor:pointer">POINTER</a></br>
10<a href="#" style="cursor:move">MOVE</a></br>
11<a href="#" style="cursor:text">TEXT</a></br>
```



```

12<a href="#"
   style="cursor:url('owncursor.jpg')">MY_CURSOR</a>
</body>
</html>

```

Бу мисол орқали ҳар хил турдаги сичқонча кўрсаткичларини кўришингиз мумикин, ҳаттоки ўзингиз ўрнатган расм кўринишида ҳам(`cursor:url('owncursor.jpg')`).

№18 CSS дарс - Жадвал ячейкаларига стил бериш: BORDER-*, EMPTY-CELLS, TABLE-LAYOUT

Сайт яратишда жадваллар жуда катта аҳамиятга эга, шу сабабли CSS ҳам жадваллар учун алоҳида стиллар ишлаб чиқган, мақола айнан жадвалларни ташқи кўринишини чиройли қилиб берадиган стих хусусиятларига бағишланади. Агар, сиз HTML гиперматн тилида жадвалларни ҳосил қилишни билмасангиз, "HTML да жадваллар" номли мақолани ўқиб чиқинг, сўнг бу мақолага қайтиб стилларни ўрганишни давом эттиринг.

Жадвалга стил бериш учун CSSда қуйидаги стил хусусиятлари мавжуд:

- **border-collapse;**
- **border-spacing;**
- **border-side;**
- **empty-cells;**
- **table-layout.**

Булар асосан ячейкалар билан ишлайди, келинг бу стил хусусиятларини бирма-бир кўриб чиқамиз.

Border-collapse - жадвал ячейкалари чегараларини чизиш. Сиз ячейкаларга рамка ўрнатсангиз, ячейка бирлашган жойида иккита чегара бўлиб қолади. Бу калит сўз орқали сиз, шу иккита рамкани битта қилиш имконига эгасиз. Қийматлари:

- collapse- ячейкалар орасида битта умумий рамка бўлади.
- separate - ҳар бир ячейкада ўз рамкаси бўлади, чегарада эса иккита рамка бўлади.

Бу хусусиятсиз мисол кўрамиз:

```

1    <html>
2    <head>
3    <title>border-spacing</title>
4    <style>
5    table {
6    border:2px solid red;
7    }
8    th{
9    border:2px solid green;
10   }
11   td{
12   border:2px solid green;
13   }
14   </style>
15   </head>

```

```

15 <body>
16 <table>
17 <tr>
18 <th>MU</th><th>MU2012</th>
19 <th>MU2013</th><th>MU2014</th>
20 </tr>
21 <tr>
22 <td>Trener</td><td>Ferguson</td>
23 <td>Moyes</td><td>Van Gal</td>
24 </tr>
25 <tr>
26 <td>Kapitan</td><td>Vidic</td>
27 <td>Vidic</td><td>Rooney</td>
28 </tr>
29 <tr>
30 <td>Star</td><td>Van Persi</td>
31 <td>Rooney</td><td>Di Maria</td>
32 </tr>
33 </table>
34 </body>
35 </html>
36

```

MU	MU2012	MU2013	MU2014
Trener	Ferguson	Moyes	Van Gal
Kapitan	Vidic	Vidic	Rooney
Star	Van Persi	Rooney	Di Maria

Энди, CSS хусусиятини ишлатамиз, фақат **"table"** стилини ўзгартирамиз:

```

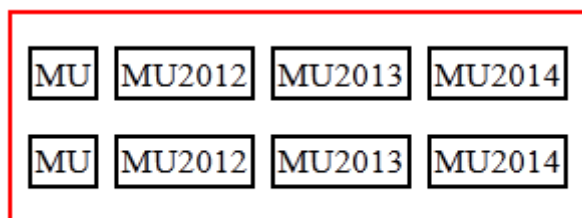
1 table {
2   border:2px solid red;
3   border-collapse:collapse;
4 }

```

MU	MU2012	MU2013	MU2014
Trener	Ferguson	Moyes	Van Gal
Kapitan	Vidic	Vidic	Rooney
Star	Van Persi	Rooney	Di Maria

Border-spacing - Жадвалдаги ячейкалар орасидаги масофани ўрнатади. Агар "**border-collapse**" ўрнатилган бўлса, бу стил хусусияти ишламайди. Иккита қиймат қабул қилиши мумкин, қийматлар пикселда берилади, биринчи қиймат горизонтал ҳол учун, иккинчиси вертикал. Агар, битта қиймат бўлса, икки ҳолат учун бир хил бўлади.

```
1
2   <html>
3   <head>
4   <title>border-spacing</title>
5   <style>
6   table {
7   border:2px solid red;
8   border-spacing: 8px 15px;
9   }
10  td{
11  border:2px solid black;
12  }
13  </style>
14  </head>
15  <body>
16  <table>
17  <tr>
18  <td>MU</td><td>MU2012</td>
19  <td>MU2013</td><td>MU2014</td>
20  </tr>
21  <tr>
22  <td>MU</td><td>MU2012</td>
23  <td>MU2013</td><td>MU2014</td>
24  </tr>
25  </table>
26  </body>
27  </html>
```



MU	MU2012	MU2013	MU2014
MU	MU2012	MU2013	MU2014

Caption-side - HTML гиперматн тилининг "**Caption**" атрибути орқали ҳосил қилинган жадвал номини(caption атрибути ҳақида), керакли жойга ўрнатиш. Қийматлари:

- top - жадвал юқорисига ўрнатиш;
- bottom - жадвалнинг қуйи қисмига ўрнатиш;

- left - жадвалнинг чап томонига ўрнатиш(манимча фақат Firefox браузерда ишлайди);
- right- жадвалнинг ўнг томонига ўрнатиш(манимча фақат Firefox браузерда ишлайди);

```

1
2   <html>
3   <head>
4   <title>Margin Padding</title>
5   <style>
6   table {
7   border-collapse:collapse;
8   }
9   caption{
10  caption-side:bottom;
11  }
12  td{
13  border:2px solid red;
14  border-collapse:collapse;
15  }
16  </style>
17  </head>
18  <body>
19  <table>
20  <caption>Jamoat</caption>
21  <tr>
22  <td>MU</td><td>MU2012</td>
23  <td>MU2013</td><td>MU2014</td>
24  </tr>
25  <tr>
26  <td>MU</td><td>MU2012</td>
27  <td>MU2013</td><td>MU2014</td>
28  </tr>
29  </table>
30  </body>
31  </html>

```

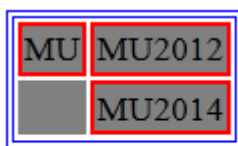
MU	MU2012	MU2013	MU2014
MU	MU2012	MU2013	MU2014

Jamoat

Empty-cells - агар жадвал ячейкаси бўш бўлса(ҳеч қандай маълумот бўлмаса, пробел бўлса, кейинги қаторга ўтиш белгиси бўлса, **visibility=hide** бўлса, яъни кўринмас бўлса) бу стил хусусияти, шу ячейкани худда маълумот бордагидек кўриниш ҳосил қилиб беради(рамкаларини кўрсатади). Қийматлари:

- show - маълумот бордигидек рамкаларини кўрсатиб беради.
 - hide - умуман стилларини олиб ташлайди, ҳатто фон ранги ҳам бўлмайди.
- Бу стил хусусиятисиз мисол кўрамиз.

```
1
2   <html>
3   <head>
4   <title>Empty</title>
5   <style>
6   table {
7   border:4px double blue;
8   }
9   td{
10  background:gray;
11  border:2px solid red;
12  }
13 </style>
14 </head>
15 <body>
16 <table>
17 <tr>
18 <td>MU</td><td>MU2012</td>
19 </tr>
20 <tr>
21 <td></td><td>MU2014</td>
22 </tr>
23 </table>
24 </body>
25 </html>
```



MU	MU2012
	MU2014

Энди CSS стилга "**empty-cells**"ни қўшамиз ва қуйидагига эга бўламиз:

```
1   td{
2   background:gray;
3   border:2px solid red;
4   empty-cells:show;
5   }
```

MU	MU2012
	MU2014

Чегара ҳосил бўлди, агар "empty-cells:hide;" қилсак умуман стил йўқолади.

MU	MU2012
	MU2014

Table-layout - жадвал ячейкасини узунлигини ўрнатиш, қийматлари:

- auto - браузер жадвални экранга чиқаради, анализ қилиб, кейин ўлчамини ўзгартириб қўяди;
- fixed - аниқ ўрнатилган ўлчам, бу қийматни бериш учун жадвал узунлиги ўрнатилган бўлиши лозим.

№19 CSS дарс - Веб саҳифаларни чоп этишга тайёрлаш: WIDOWS, ORPHANS, PAGE-BREAK-*

Баъзи пайтларда веб саҳифаларни қоғозга чоп қилиш керак бўлади, веб саҳифанинг қоғоздаги варианты биров ноқулайроқдир. Чунки веб саҳифа яратувчиси, чоп қилишни эътиборга олмаган бўлади. CSS хусусиятлари орқали веб саҳифаларни чоп қилишни созлаш мумкин. Бу хусусиятлардан кўпчилик фойдаланмайди, лекин тўғри ишлатилса, саҳифанинг қоғоздаги варианты ҳам чиройли бўлади. Чоп этиш хусусиятлари фақат блоки(p, div)элементларда ишлайди.

Қуйида чоп этишга оид CSS хусусиятлари рўйхати берилган:

- widows;
- orphans;
- page-break-after;
- page-break-before;
- page-break-inside.

Бу хусусиятларни бирма-бир кўрсатаман, лекин мисолларни расм шаклида кўрсатмайман, чунки бунинг учун камида бир варрок тўла матн бўлиши керак. Шундай бўлса ҳам, тушунарли ёзишга ҳаракат қиламан. Чоп этиш хусусиятлари фақат саҳифа чоп қилинганда ёки чоп қилишдан олдин кўришни (**предварительный просмотр**) амалга оширганда ишлайди, код натижасини кўришда чоп этиш хусусиятлари ишламайди.

1. Widows - бу хусусият бирор варрок тўлганда, кейингисига ўтиш керак бўлган жойда ишлайди, яъни блок ичидаги матннинг минимал нечта қатори кейинги варрокка ўтиш кераклигини аниқлаштиради. қиймат сифатида ракам берилади, мисол учун:

```

1      <html>
2      <head>
3      <style>
```

```

4      @media print {
5          p{ widows:3; }
6      }
7  </style>
8  </head>
9  <body>
10 <p>Katta hajmdagi matn bo'lishi mumkin</p>
11 </body>
12 </html>

```

Бу мисолдан бирор натижа ололмайсиз, чунки матн бир варрокни тўла эгалламаган, мисолда фақат бу хусусиятни қандай ишлатишни кўрсатдим. "@" белгиси веб саҳифага бирор ташқи нарсани боғланаётганини кўрсатади, бизнинг ҳолда бу принтер.

2. Orphans - веб саҳифани чоп қилишда, матннинг минимал нечта қатори дастлабки варрокда қолиш кераклигини кўрсатади. Қиймат рақамда берилади. Агар сиз берган қиймат, яъни қаторлар сони дастлабки варрокқа сиғмаса, бутун блок матнини кейинги қаторга ўтказиб юборади. **Orphans** билан **widows** хусусиятлари бир - бирига қарама қарши қийматларни қабул қилса, **widows** хусусияти қиймати ишлайди, приоритет **widows** да кўпроқ.

3. Page-break-after - блокларни варрок бўйлаб бўлишни амалга ошириб беради. Саҳифада бир неча блоклар бўлса, маълум бир блокдан кейин веб саҳифани бўлиб, қолганини кейинги варрокқа ўтказиб юборади. Қуйидаги қийматларга эга:
always - ҳар доим шу элементдан кейин қолган қисмни кейинги варрокга ўтказиш;
auto - агар зарур бўлса қолган қисмни кейинги варрокга ўтказади, ўзи аниқлайди;
avoid - элементдан кейин кейинги варрокга ўтишни тақиклайди;
left - элемендан кейин варрокни бўлиб, қолган элементларни битта ёки иккита бўш варрок ташаб кейингисига ёзади, бу орқали матнларни фақат жуфт варроқларга ёзади, тоқ варроқлар бўш бўлади(chrome, safari, firefoxда ишламайди, Operaда ишлайди);

right - элемендан кейин варрокни бўлиб, қолган элементларни битта ёки иккита бўш варрок ташаб кейингисига ёзади, бу орқали матнларни фақат тоқ варроқларга ёзади, жуфт варроқлар бўш бўлади(chrome, safari, firefoxда ишламайди, Operaда ишлайди);

```

1  <html>
2  <head>
3  <title>page </title>
4  <style>
5  @media print {
6  .p {
7  page-break-after: always;
8  }
9  }
10 </style>
11 </head>
12 <body>

```

```

12     <h2>Hello World</h2>
13     <p>Bu qism birinchi varroqda bo'ladi</p>
14     <span>bu esa keyingisiga o'tadi</span>
15 </body>
16 </html>
17

```

4. Page-break-before - блокларни варроқ бўйлаб бўлишни амалга ошириб беради. Саҳифада бир неча блоклар бўлса, маълум бир блокдан аввал веб саҳифани бўлиб, шу жойдан бошлаб, қолганини кейинги варроққа ўтказиб юборади. Қийматлари **"page-break-after"** билан бир хил.

5. Page-break-inside - бу хусусият блок ичи билан ишлайди ва берилган қиймат асосида бўлишни амалга ошириб беради. Қийматлари:

auto - керак бўлганда блок ичидаги матнни бўлиши мумкин, матн қисмларга ажралиб, бошқа-бошқа варроқларда чоп этилади;

avoid - блок ичидаги матнни бўлмайди, блок бутунлай у ёки бу варроқда чоп этилади.

```

1     <html>
2     <head>
3     <title>Inside</title>
4     <style>
5     @media print {
6     p {
7         page-break-inside: avoid;
8     }
9     }
10    </style>
11    </head>
12    <body>
13    <p>Bu matn bitta varroqda namoyon bo'ladi, matnning bir qismi
14        bir varroqda ikkinchisi boshqasida bo'lmaydi.</p>
15    </body>
16    </html>

```

№20 CSS якунловчи дарс - Блоклар жойлашуви DISPLAY ҳақида

Бу мақола билан **CSS2** стилига тегишли барча стил хусусиятларни кўриб чиқган бўламиз. Кейин, кўпчилик учун қизиқ бўлган **CSS3** ҳақида ёзишни бошлашим мумкин бўлади. Охирги хусусият **"display"** деб аталади.

Display - блокли элементларни веб саҳифада қай кўринишда чиқариш ва уни жойлашувини аниқлаштириб беради, бу хусусият орқали олдин кўриб ўтган бир неча хусусиятларнинг вазифасини бажарса бўлади, унинг учун бироз ўйлаб бу хусусиятни ишлатиш лозим.

Бу хусусиятни бир неча қийматлари бор, уларни бирма-бир кўриб чиқамиз.

block - бу қиймат қаторли элементлар(span, hx,...)ни ҳам блокли(div, p) қилиб кўрсатади, яъни қаторли элементларни кетма-кет ёзсангиз ҳам алоҳида қаторда чиқариб беради(қаторли элементлар бир қаторда чиқиши лозим);

inline - бу қиймат юқоридаги қийматга тесқари, яъни блокли элементларни қаторли қилиб беради, яъни "**div**", "**p**" теглар бир қаторда ёзилиши мумкин.

Юқоридаги икки қийматни битта қилиб, мисол кўрамиз.

```
1    <html>
2    <head>
3    <title>Display_1</title>
4    </head>
5    <body>
6    <span>Qatorli element </span>
7    <span><b>DISPLAY</b> ishlatilmadi</span>
8    <span style="display:block; color:red;"><b>Display</b> ishlatilib,
9    <span style="color:red">span blokli element bo'lib qoldi</span><br>
10   <span>-----</span>
11   <p>Bu blokli element</p>
12   <p>shunig uchun ikkinchi blok pastga tushib qoldi</p>
13   <p style="display:inline; color:red;">Buham blokli element, </p>
14   <p style="display:inline; color:red;">lekin <b>display</b> orqali
15   </body>
16   </html>
```

Мисолда, қаторли(span)элементларни блокли, блокли(p)элементлар қаторли қилинди.

Qatorli element **DISPLAY** ishlatilmadi

Display ishlatilib,
span blokli element bo'lib qoldi

Bu blokli element

shunig uchun ikkinchi blok pastga tushib qoldi

Bu ham blokli element, lekin **display** orqali qatorli element bo'lib qoldi

inline-block - бу қиймат бизга икки хил нарсани аңлатади, яъни киритилган контент ички томондан блок, контент ичи блок, лекин, ташқи томондан қаторли тип, бошқа объектлар учун қатор теги критилган ҳисобланади. Бироз ғалати, ички томони блок, ташқи томони қаторли контент, ҳали ҳам тушунмаган бўлсангиз мисолда кўрсатаман.

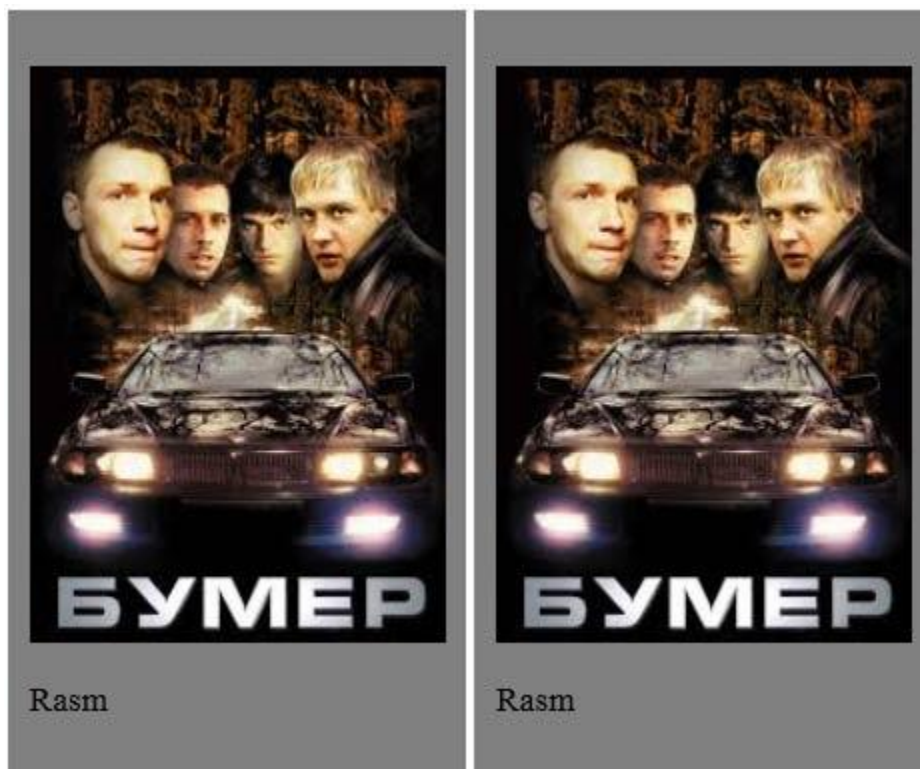
```
1    <html>
2    <head>
3    <title>Display_2</title>
4    <style>
5    .rasm{
6    background-color:gray;
7    padding:10px;
8    width:190px;
```

```

9     display: inline-block;
10  }
11  </style>
12  </head>
13  <body>
14    <div class="rasm">
15      <p></p>
16      <p>Rasm</p>
17    </div>
18    <div class="rasm">
19      <p></p>
20      <p>Rasm</p>
21    </div>
22  </body>
23  </html>

```

Иккита расм саҳифага чиқарилди, бу расм блокли элемент(`div`)орқали чиқарилди, агар **"display"** хусусиятини киритмасангиз, расмлар ёнма-ён эмас, қаторма-қатор чиқади, расм, унинг тегида иккинчи расм.



"Display" нинг **"inline-block"** қиймати, расмни блок шаклида қолдириб, кейинги расм(кейинги объект) учун бу тег(`div`)ни қаторли тег деб таништирмоқда. Шунинг учун, расмлар кетма-кет чиқмоқда. Иккинчи расм учун, биринчи расм қатор типига тегишли, лекин биринчи расм ўзи учун блокли.

`inline-table` - юқоридаги қиймат билан бир хил, бунда блок-қатор ролини жадвал ўйнайди, мисолда кўрсатаман.

```

1    <html>
2    <head>
3    <title>Display_2</title>

```

```

4     </head>
5     <body>
6     <table border=1>
7     <tr><td>2000-2001</td> <td>2001-2002</td></tr>
8     <tr><td>2003-2004</td> <td>2004-2005</td></tr>
9     <table>
10    Bu jadval oqali hamma har xil narsani tushunishi mumkin, odatda bu
11    <font color="red">-----
12    <table border=1 style="display:inline-table">
13    <tr><td>1900-1901</td> <td>1901-1902</td></tr>
14    <tr><td>1903-1904</td> <td>1904-1905</td></tr>
15    <table>
16    Hali ham nimaligini tushunmadingizmi?
17    </body>
18    </html>

```

Дастлабки ҳолда, жадвал блок кўринишида, шунинг учун ундан кейинги матн жадвалдан кейин чиқмоқда

2000-2001	2001-2002
2003-2004	2004-2005

Bu jadval oqali hamma har xil narsani tushunishi mumkin, odatda bu yillar...

1900-1901	1901-1902
1903-1904	1904-1905

Hali ham nimaligini tushunmadingizmi?

Кейинги ҳолда эса, **"display" "inline-table"** қиймати билан ишлатилмоқда, яъни жадвалдан кейинги объект(ёзув) жадвални қаторли тип деб ўйлайди ва ёнидан чиқади.

none - шу қийматга эга бўлган блок кўринмас ҳолга келади. Ҳаттоки у олган жой ҳам сақланмайди.

```

1     <html>
2     <head>
3     <title>Display_2</title>
4     </head>
5     <body>
6     <div>Uzbekistan, Tashkent</div>
7     <div style="display:none">England, London</div>
8     <div>Russia, Moscow</div>
9     </body>
10    </html>

```

Uzbekistan, Tashkent

Russia, Moscow

манимча бу қиймат жуда яхши тушунарли.

list-item - блоklar рўйхатдан иборат эканлигини ўрнатади, шунда маркерлар ҳам ҳосил бўлади.

```
1 <html>
2 <head>
3 <title>Display_5</title>
4 <style>
5 div{
6 display:list-item;
7 }
8 </style>
9 </head>
10 <body><ol>
11 <div>Uzbekistan, Tashkent</div>
12 <div>England, London</div>
13 <div>Russia, Moscow</div>
14 </ol>
15 </body>
```

маркер турларини **"list-style"** орқали бериш мумкин.

1. Uzbekistan, Tashkent
2. England, London
3. Russia, Moscow

"Display"нинг қолган қийматлари жадваллар билан боғлиқ бўлиб, html гипермат тили жадвал теглари билан бир хил(HTML да жадваллар), яъни

CSS стил хусусиятилари	HTML гипермат тили
Table	<table>
Table-caption	<caption>
Table-cell	<td>, <th>
Table-column	<col>
Table-column-group	<colgroup>
Table-footer-group	<tfooter>
Table-header-group	<thead>
Table-row-group	<tbody>
Table-row	<tr>