

## Дәріс №1. CSS- сайт рәсімдеуде қолданылатын негізгі тәсіл

Егер HTMLді үйдің кірпіштерден тұратын қаңқасына теңесек, онда CSS ті сол кірпіштерді үйлесімді түрде қалап, оларды біріктіріп тұрған цемент қоспасына теңеуге болады. Екеуінің бірі болмаса сайт та болмайды.

Сонымен, CSS деген не және CSS деген HTML мен қалай байланысады.

CSS дегеніміз ағылшынша Cascading Style Sheets , яғни қазақша стилдік каскад таблицалары деп аталады.



Біз HTML ге арналған сабақтан аздап болса да web парақшаны қалай жасауға болатынын көрдік. Ол көрген тәсіліміз тек бер жағы ғана. Бізге ол парақшаның ешкімнің сайт парақшасына ұқсамайтын өзгеше болғанын қалаймыз, яғни оны әрлендіруіміз керек. Міне осы әрлендіруді жасайтын сол CSS.

HTML ді кірпішпен қаланып, төрт қабырғасы тұрғызылып және төбесі жабылған үйге ұқсатуға болады. Ал сол үйдің ішін сырлап, еденін төсеп, мебельдерін қойып дегендей т.б. әрекеттер жасайтын осы CSS.

### HTML ГЕ ҚАРАҒАНДА CSS ТІҢ СИНТАКСИСІ КҮРДЕЛІРЕК.

Себебі, HTML дің бірнеше тегтерінің қалай пайдалану керектігін ұғып алған соң еш қиналмастан парақша бетін жаза беруіңізге болады. Ал CSS стилді белгілеу деп аталады яғни нюанстары көбірек, оның әр сипаттамасын біліп қана қоймай оны қай жерде қалай қолдануға болатынын да үйреніп алу керек.

### БҰЛ СТИЛЬДІ БЕЛГІ ТІЛІ ЕКІ БӨЛІКТЕН ҚҰРАЛАДЫ:

1. Селектор-меткалардан.
2. Осы селекторларға қолданылатын ережелерден.



Яғни бұл ереже бойынша сіздің документіңіздің (парақшаңыздың) ішіндегі p абзацының әріптері қызыл түске, ал артқы жағы (фоны) жасыл түске боялуы керек.

Сонымен, HTML тілімен бірге CSS ті қолданудың үш түрі бар.

Енді осылардың әр қайсысына жеке-жеке тоқталып өтейін. Ескерту:

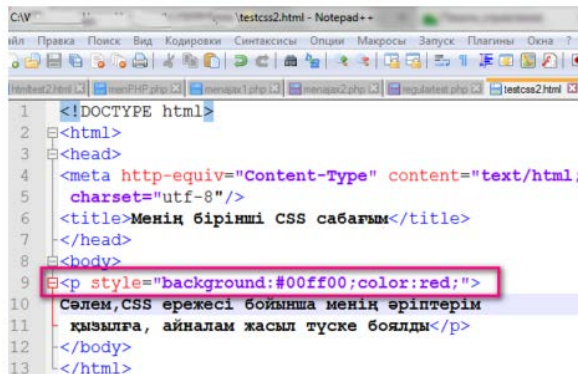
Төмендегі кодтардың бәрі Notepad++ редакторында жазылған, оны компьютерге орнатудың түк қиындығы жоқ. Мен оны қалай орнату керектігіне тоқталып өткенмін, ол туралы барлық мағлұматты сілтеме арқылы таба аласыз.

## 1. ТІКЕЛЕЙ HTML ТЕГІНІҢ ІШІНДЕ "STYLE" АТРИБУТЫН ОРНАЛАСТЫРУ АРҚЫЛЫ:

Редактор бетіне төмендегі HTML парақшасының кодын көшіріп алыңдар.

Ішінде p тегінде CSS ережесі орнатылған.

Түсінікті болу үшін айтып өтейін: background-фон дегенді білдіреді, #00ff00-жасыл түстің коды, color-бояу түсі, red-қызыл түс.

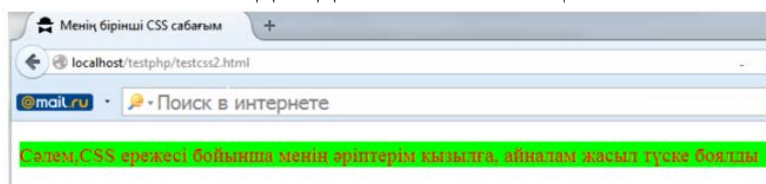


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html;
5 charset="utf-8"/>
6 <title>Менің бірінші CSS сабағым</title>
7 </head>
8 <body>
9 <p style="background:#00ff00;color:red;">
10 Сәлем, CSS ережесі бойынша менің әріптерім
11 қызылға, айналам жасыл түске боялды</p>
12 </body>
13 </html>
```

Енді осы файлға "testcss2.html" деген ат беріп, оны локальдық серверіміз Денверге арналған "\\servers\home\localhost\www\testphp" папкаға сақтап болған соң, оны қалаған браузеріңіздің адрес теретін жолында localhost/testphp/testcss2.html деп шақырамыз .

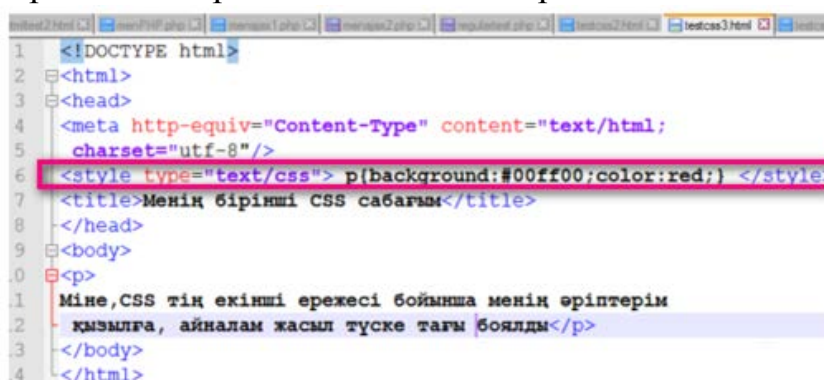
Ол html файлды Денвер арқылы емес, жай түртіп жіберіп те нәтижесін көруге болады. Бірақ, сіз егер PHP скрипт тілін үйренгіңіз келсе, онда Денверде жұмыс істеу сізге дағдыға айналуы керек. Сондықтан, тест файлдарды осы локальдық серверде шақырғанымыз дұрыс деп санаймын. Егер Денверді пайдаланғыңыз келмесе, онда ол файлды папкасында екі рет түртіп жіберіп қарай беріңіз.

Нәтижесі төмендегідей болып шықты.



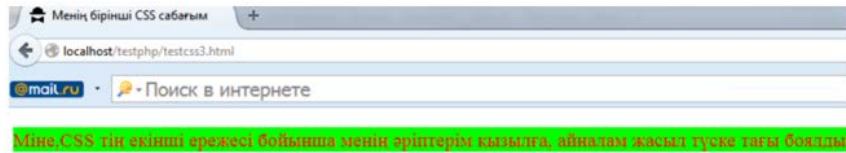
## 2. WEB ПАРАҚШАНЫҢ ШАПКАСЫНДА, ЯҒНИ <HEAD>АРАСЫНА</HEAD> "STYLE" ЕЛЕМЕНТІ АРҚЫЛЫ:

CSS ті бейнелеудің екінші түрі жазылған төмендегі HTML парақшасының кодын тағы да Notepad ке көшіріп жазып алыңыздар.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html;
5 charset="utf-8"/>
6 <style type="text/css"> p{background:#00ff00;color:red;} </style>
7 <title>Менің бірінші CSS сабағым</title>
8 </head>
9 <body>
10 <p>
11 Міне, CSS тің екінші ережесі бойынша менің әріптерім
12 қызылға, айналам жасыл түске тағы боялды</p>
13 </body>
14 </html>
```

Ол файлды шақырғанда шыққан нәтиже төмендегідей:



### 3. CSS СТИЛЬДЕР КОДЫН ОСЫ ПАРАҚ ШАПКАСЫНДА <LINK> АРҚЫЛЫ СИПАТТАЛЫП, ЖҮКТЕЛЕТІН БӨЛЕК ФАЙЛҒА ЖАЗЫП ҚОЮ АРҚЫЛЫ:

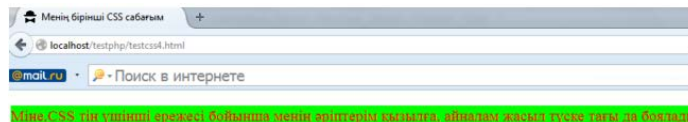
CSS ті қолданудың үшінші түрінде CSS ережелерін басқа файлға жазамыз. Мен оны testcss2.css деп жаздым да сол қалған файлдар салынған папкаға сақтап қойдым. Сіз де солай жасаңыз.

```
1 p{background:#00ff00;color:red;}
```

Енді оны HTML парақшамның жоғары жағындағы <head>аралығына link арқылы сипаттап жазамын.</head>

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html;
5 charset=utf-8"/>
6 <link href="testcss2.css" rel="stylesheet" type="text/css">
7 <title>Менің бірінші CSS сабағым</title>
8 </head>
9 <body>
10 <p>
11 Міне, CSS тің үшінші ережесі бойынша менің әріптерім
12 қызылға, айналам жасыл түске тағы да боялды</p>
13 </body>
14 </html>
```

Енді осы html файлды ашамын, сонда мынадай нәтижесі бар HTML парақшам шығады.



Бұл жерде айта кететін жағдай былай:

CSS ті HTML де қолданудың алдыңғы екі түрі көп қолданылмайды, бірақ білген абзал-олар көбіне сайтыңызды тексеру кезінде кейде почтаны қолдану кезінде қолдана аласыз. Ал сайттарда қолданылатыны, негізінде үшінші түрі.

Енді CSS селекторлар туралы келесі мақаламды қарап шығыңыз.

## Дәріс №2. CSS Селекторларының кең қолданылатын түрлері

CSS туралы алдыңғы мақалада айтып кеткенімдей, оның ережелері селекторлар арқылы белгіленіп HTML кодына енгізіледі. Яғни, HTML тегі осы ережелер арқылы қандай да бір қасиетке ие болады. Сол арқылы парақша беттеріндегі әр элемент керекті орнына жайғасады.

CSS селекторлары бірнеше түрге бөлінеді, олардың санының нақты қанша екенін айтып қажеті де жоқ. Өйткені CSS әр жаңа нұсқасымен бірге жаңа түрлері пайда бола береді, сондықтан мен тек қазіргі қолданыста жүргендерін және көп қолданылатын түрлерін ғана айтып өтпекпін.

Осы жерде естеріңізге сала кетейін, бұл селекторларды біз бұрын айтып кеткенімдей бөлек CSS файлда сипаттап жазамыз да, HTML кодының <head></head> бөлігінде link арқылы жүктеу үшін жазып кетеміз.

Селекторлардың мынадай түрлері бар:

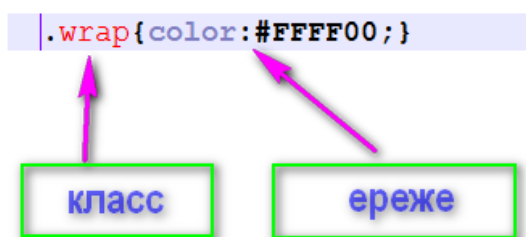
### 1. HTML ТЕГТЕРІ СЕЛЕКТОРЛАР РЕТІНДЕ ҚОЛДАНЫЛАДЫ.

Мен осының алдындағы сабағымда соның бір түрі "P" тегінің селектор ретінде абзац фонын әр түрлі түске бояғанын практика жүзінде көрсеттім.



Яғни кез келген HTML тегін селектор ретінде қолдана аламыз. Бірақ, мұның кемшілігі - ереже тек бір тегке байланып қалады. Мысалы, мен басқа бір абзацтың әріптерін қызылдан басқа түске бояғым келуі мүмкін ғой. Егер сондай қажеттілік туындай қалса көмекке келесі "класс" деп аталатын селекторлар келе алады.

### 2. КЛАСС СЕЛЕКТОРЛАРЫ БЫЛАЙ ЖАЗЫЛАДЫ:



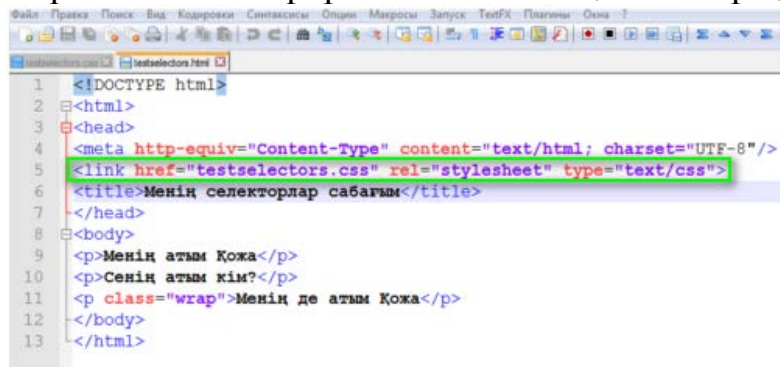
Төменгі суретте екі түрлі селектор көрсетіліп тұр.

```
1 p{color:#00FF00;}
2 p.wrap{color:#FFFF00;}
```

Көріп отырғаныңыздай P тегі екі түрдегі ережемен беріліп тұр. Бұлар не деп тұр екен, талдау жасайық.

Бірінші ереже, яғни бұл алдында айтып өткен тег селекторы ережесі бойынша, барлық абзацтарымыз жасыл түспен жазылуы керек. Бірақ, олардың ішінен қалаған абзацтарды өзгерте аламыз. Қалай дейсіздер ғой, жауабын табу

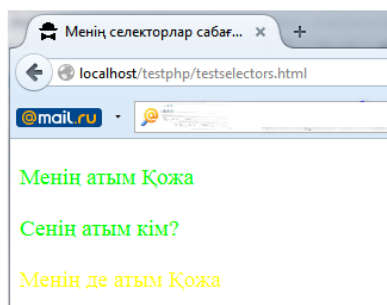
үшін жоғарыдағы CSS стильдік таблицаны "testselectors.css" деп, ал төмендегі HTML кодын "testselectors.html" деп өзіміз жақсы білетін Notepad редакторына көшіріп жазып "testphp" папкасына сақтап жіберіңіздер.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset="UTF-8"/>
5 <link href="testselectors.css" rel="stylesheet" type="text/css">
6 <title>Менің селекторлар сабағым</title>
7 </head>
8 <body>
9 <p>Менің атым Қожа</p>
10 <p>Сенің атым кім?</p>
11 <p class="wrap">Менің де атым Қожа</p>
12 </body>
13 </html>
```

Енді осыны локальдық сервер Денверде ашып қарайық. Ол үшін әр уақытта оған бірінші кірерде ярлық "Start Denwer"ді түртеміз. Сосын кез келген браузердің адрес теретін жолында "localhost/testphp/testselectors.html" деп шақырамыз. Нәтиже төмендегідей болды.

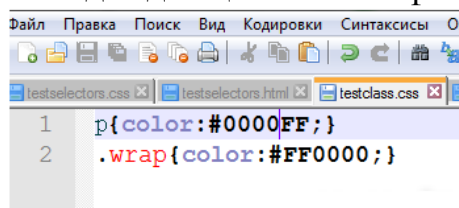
Ол html файлды Денвер арқылы емес, жай түртіп жіберіп те нәтижесін көруге болады. Бірақ, сіз егер PHP скрипт тілін үйренгіңіз келсе, онда Денверде жұмыс істеу сізге дағдыға айналуы керек. Сондықтан, тест файлдарды осы локальдық серверде шақырғанымыз дұрыс деп санаймын. Егер Денверді пайдаланғыңыз келмесе, онда ол файлды папкасында екі рет түртіп жіберіп қарай беріңіз.



Яғни бұл жерде өзіңіз көріп отырғандай бірінші ереже бойынша барлық абзацтар жасылға боялды, бірақ сол абзацтардың кез келгенін белгілі бір өзгеше ережемен сипаттасақ, ол "p"тегімен байланысын үзіп басқа ережеге бағынады. Яғни біздің мысалда ол барлық берілген жасыл түстен басқа сары түске ие болды. Яғни "p.wrap"-класс селекторының дәрежесі "p" - тег селекторы дәрежесінен жоғары болып тұр.

Ал егер бір ережемен бірнеше тегтерді бағындыру керек болса, онда жоғарыда көрсетілген екінші ереже сықылды бірақ HTML тегі көрсетілмеген класс селекторын пайдалану керек.

Олай болса Notepadке төмендегідей қылып CSS файлын жазайық:



```
1 p{color:#0000FF;}
2 .wrap{color:#FF0000;}
```

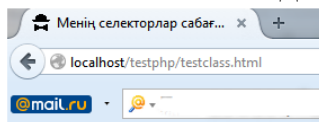
Сосын осы каскадты стильдік кестелер ережесі бекітілетін HTML кодын жазайық:

```
Файл Правка Поиск Вид Кодировки Синтаксисы Опции Макросы Запуск TextFX Плагинь Окна ?
testclass.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset="UTF-8" />
5 <link href="testclass.css" rel="stylesheet" type="text/css">
6 <title>Менің селекторлар сабағым</title>
7 </head>
8 <body>
9 <p>Менің атым Қожа</p>
10 <p class="wrap">Менің атым Қожа</p>
11 <div class="wrap">Сенің атың кім?</div>
12 <p>Менің де атым <span class="wrap">Қожа</span></p>
13 </body>
14 </html>
```

Бұл файлдың атын "testclass.html" деп атап "testphp" папкасына салып жіберелік.

Өздеріңіз көріп отырғандай, бірінші "p" абзацы көк түс кодымен #0000FF , ал екінші "p" абзацы, "div" блог тегі және "span" тегімен қоршалған Қожа сөзі қызыл түс кодымен #FF0000 ережеленген.

Енді осы жазған HTML коды жазылған файлымызды браузерде localhost/testphp/testclass.html деп шақырайық. Менде нәтижесі төмендегідей болды және сізде де солай болғанына сенімдімін.



Менің атым Қожа  
Менің атым Қожа  
Сенің атың кім?  
Менің де атым Қожа

### 3. ҮШІНШІ СЕЛЕКТОРЛАР ТҮРІ - БІРЕГЕЙ (БІРДЕН-БІР) ИДЕНТИФИКАТОР ДЕП АТАЛАДЫ.

Олардың класс селекторларынан айырмашылығы нүктенің орныны # (хеш) қолданылады және бірегей идентификатор парақша бетінде тек бір тегпен бірге бір-ақ рет жазылады.

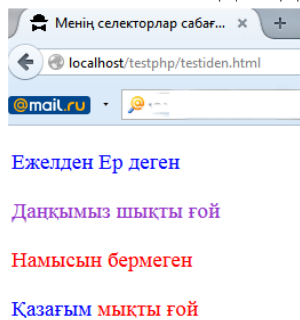
Тағы да Notepad те алдыңғы жазып кеткен "testclass.css" файлын ашып ішіне #abc деген бірегей идентификаторын күлгін түс коды #9A32CD ережесімен қосып жазайық.

```
Файл Правка Поиск Вид Кодировки Синтаксисы
testselectors.css testselectors.html testclass.css
1 p{color:#0000FF;}
2 .wrap{color:#FF0000;}
3 #abc{color:#9A32CD;}
```

Енді осы стильдік кестелер файлын төмендегі көрсетілген HTML кодына link арқылы жазып жіберелік.

```
Файл Правка Поиск Вид Кодировка Синтаксис Опции Макросы Запуск TestFX Планиры Оена ?
localhost/testphp/testiden.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset="UTF-8"/>
5 <link href="testclass.css" rel="stylesheet" type="text/css">
6 <title>Менің селекторлар сабағы</title>
7 </head>
8 <body>
9 <p>Ежелден Ер деген</p>
10 <p id="abc">Даңқымыз шықты ғой</p>
11 <div class="wrap">Намысын бермеген</div>
12 <p>Қазағым <span class="wrap">мықты ғой</span></p>
13 </body>
14 </html>
```

Осы HTML гипертексті тілінің web парақшаға айналған түрін көру үшін тағы да браузер арқылы оны былай деп шақырамыз localhost/testphp/testiden.html. Нәтижесі төмендегідей:



#### 4. СЕЛЕКТОРЛАРДЫҢ ТӨРТІНШІ ТҮРІ ӘМБЕБАП (УНИВЕРСАЛЬНЫЙ) ИДЕНТИФИКАТОР ДЕП АТАЛАДЫ.

Ол былай белгіленеді:

\* {color:red;} -яғни бұл ереже бойынша парақ бетіндегі барлық әріптер қызыл түске ие болады.

#### 5. АТРИБУТТАР СЕЛЕКТОРЛАРЫ ДЕП АТАЛАТЫН СЕЛЕКТОРЛАР HTML ТЕГІНІҢ АТРИБУТТАРЫН КВАДРАТ ЖАҚШАҒА АЛЫП СИПАТТАЛАДЫ.

HTML туралы мақаламда мен атрибуттар туралы айтып өткенмін. Солардың бірі сілтемелерде қолданылатын "href" ті осы атрибуттар селекторына бір мысал ретінде қарастыруымызға болады.

[href^="http"]{color:green;} - яғни бұл ереже бойынша "http" мен басталатын (^-үздіксіз сөйлемшелер ережесі бойынша сөйлемнің ең басы дегенді білдіреді) барлық сыртқы сілтемелер жасыл түске боялады деген сөз.

#### 6. ПСЕВДОКЛАСС СЕЛЕКТОРЛАРЫ HTML ТЕГІНЕН KEЙІН ҚОЙЫЛАТЫН ҚОС НҮКТЕДЕН СОҢ ОРНАЛАСАДЫ.

Олар ешқашан HTML парақшасында көрсетілмейді, тек қана стильдік кестелер CSS файлында сипатталады. Оларға жататындары мыналар:

:link{color:green;} -Яғни мына келтірілген ереже бойынша бұл селектор әлі басылмаған сілтеме әріптерінің бояуын жасыл түске бояйды.

:vизited{color:green;} - ал бұл селектор басылған сілтемелер әріптерінің бояуын жасылға бояйтынын көрсетіп тұр.

Ал келесі псевдокласс селекторлары барлық HTML тегтерінде қолданылады:

a:active{color:red;} - сол жақ мышканы көрсетілген тегті басқанда ғана оның әріптері қызыл түске боялады.

`a:hover{color:blue;}`- бұл селектор ережесі бойынша тек мышка тегтің үстінен жүріп өткенде ғана тег көрсетілген түске боялады.

## 7. ЕНДІ БАЯНДАЛАТЫН СЕЛЕКТОРЛАР ПСЕВДОЭЛЕМЕНТТЕР СЕЛЕКТОРЛАРЫ ДЕП АТАЛАДЫ.

Олар мыналар:

```
p:first-letter{color:blue;}
```

```
p:first-line{color:blue;}
```

```
p:after{content="Осымен бітті.";}
```

```
p:before{conten="Қайырлы күн";}
```

Аттары айтып тұрғандай, абзацтағы бірінші әріп көкке, бірінші қатар көкке, абзац аяқталғанда "Осымен бітті" деген сөйлем қосылады, абзац басталғанда "Қайырлы күн" деп басталады.

Бұл мақалам Осымен бітті.



## Дәріс №3. CSS ережелері - "HEIGHT", "WIDTH" және "OVERFLOW"

Бүгін сіз CSS ережелері: height (биіктік), width (ені), overflow (толып кету) туралы кең мағлұмат аласыз.

мен бұрын айтып өткен HTML ге арналған бірінші мақаламда парақша бетін гипертексті тілмен өрнектегенде html тегі бірінші жазылады, одан соң body тегі дегенмін. Яғни, html тегі парақшаның барлық кеңістігін меншіктеп алады, ал қалған тегтер парақша бетінде контейнерлер секілді біреулері нақты көрсетілген CSS ережелері арқылы, ал кейбіреулері үнсіздік бойынша өз орындарына орналаса алады.

Ал CSS туралы мағлұматтарды қайдан алуға болады, әрине бұрын мен айтып кеткен HTML тегтерінің бірыңғай стандарттары бекітілген W3C консорциумырың сайтынан, ал нақтылап айтқанда CSS спецификациясы деп аталатын парақша бетінен көре аласыз. Оны ашқанда жоғарғы жағында көрсетілген менюден properties дегенге өтіңіз, сонда мына төмендегідей кестелер тізімі сол CSS ережелері болып табылады:

[previous](#) [next](#) [contents](#) [properties](#) [index](#)

Appendix F. Full property table

This appendix is informative, not normative.

Name	Values	Initial value	Applies to (Default: all)	Inherited?	Percentages (Default: N/A)	Media groups
azimuth	<angle>   [[ left-side   far-left   left   center-left   center   center-right   right   far-right   right-side ]   behind ]   leftwards   rightwards   inherit	center		yes		aural
background-attachment	scroll   fixed   inherit	scroll		no		visual
background-color	<color>   transparent   inherit	transparent		no		visual
background-image	<url>   none   inherit	none		no		visual
background-position	[[ <percentage>   <length>   left   center   right ]   <percentage>   <length>   top   center   bottom ]? ]   [[ left   center   right ]   top   center   bottom ]   inherit	0% 0%		no	refer to the size of the box itself	visual
background-repeat	repeat   repeat-x   repeat-y   no-repeat   inherit	repeat		no		visual

Енді осы тізімдегі "height" пен "width" - дерге қарайық:

height	<length>   <percentage>   auto   inherit	auto
width	<length>   <percentage>   auto   inherit	auto

Бұл ережеде олардың үнсіздік (по умолчанию) бойынша мәні "auto" деп көрсетілген, яғни бұл оның мөлшері нақты көрсетілмесе онда браузерлер ол элементерді парақша бетінде өз қалаулары бойынша орналастырады деген сөз.

Яғни, CSS ережелері бойынша height және width олардың биіктігі мен енін анықтайды. Бірақ бұл ережеге тек блок тегтері ғана бағынады, олар туралы мен бұрынғы мақаламда баяндап өткенмін. Ал тармақ тектері ол ережелерді ескермейді.

Сондықтан, height пен width туралы айтқанда тек блок тектері сөз болады.

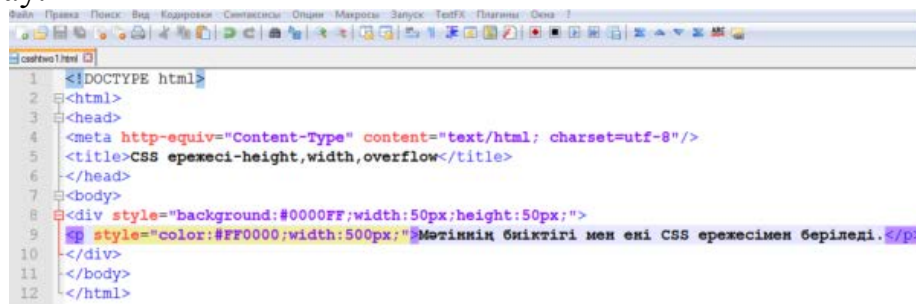
Олар жоғарыда көрсетілгендей былай сипатталады:

width: сандық өлшем | пайыздық өлшем%; height: с. ө. | п. ө. %;

Мысал үшін мына төменде көрсеткендей қылып гипертексті тілмен жазылған скрипт кодын Notepad редакторына салайық.

Ол код тәжірибе ретінде келтіріліп отырғандықтан CSS ты қолданудың бірінші тәсілін пайдаланамыз.

1. "div" контейнері мен оның ішіне салынған абзацты "width" бен "height" арқылы сипаттау.

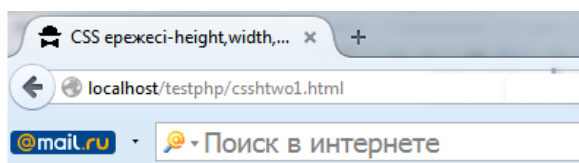


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5 <title>CSS ережесі-height,width,overflow</title>
6 </head>
7 <body>
8 <div style="background:#0000FF;width:50px;height:50px;">
9 <p style="color:#FF0000;width:500px;">Мәтіннің биіктігі мен екі CSS ережесімен беріледі.</p>
10 </div>
11 </body>
12 </html>
```

"div" контейнерінің ені оның ішіне салынған "p" тегі арқылы жазылған абзацтың енінен кіші. Бұл жағдайда абзац "div" контейнерінен шығып тұрады.

Нәтижесін Денверде, яғни локальдық хостымызда көру үшін, егер сіз оған әлі кірмеген болсаңыз, оның ярлығы "Start Denwerді" түртіп жібересіз. Сосын екі DOS тың қара окнолары жүгіріп өткен соң, кез келген браузердің адрес тармағында localhost/testphp/csshtwo1.html деп файлды шақырып, оның кодының нәтижесін көре аласыз.

Ол html файлды Денвер арқылы емес, жай түртіп жіберіп те нәтижесін көруге болады. Бірақ, сіз егер PHP скрипт тілін үйренгіңіз келсе, онда Денверде жұмыс істеу сізге дағдыға айналуы керек. Сондықтан, тест файлдарды осы локальдық серверде шақырғанымыз дұрыс деп санаймын. Егер Денверді пайдаланғыңыз келмесе, онда ол файлды папкасында екі рет түртіп жіберіп қарай беріңіз.



Мәтіннің биіктігі мен ені CSS ережесімен беріледі.

Бұл суреттен өзіңіз көріп отырғандай, абзац өз контейнеріне симай сыртына шығып тұр. Демек, біз парақшамызды жазғанда контейнер құрамы оның аумағынан шықпайтындай етіп үнемі бақылауда ұстап отыруымыз керек.

Меніңше, "width" бен "height" түсінікті сияқты. Енді "overflow" деген не екенін ұғып көрейік.

2. "overflow"- толып кеткен деген мағына білдіреді. Ол да жоғарыда келтірілген W3C стандарттарындағы ережеге сай былай сипатталады:

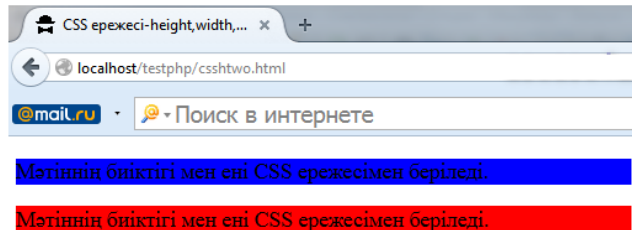
overflow	visible   hidden   scroll   auto   inherit	visible
----------	--	---------

Бұл ережеде оның үнсіздік (по умолчанию) арқылы мәні "visible" (көрсетіледі).

"overflow" қалай жұмыс істейтінін түсінуге келесі мысал көмектеседі. Сүйікті редакторымыз Notepad ке мына HTML кодын жазайық:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5 <title>CSS ережесі-height,width,overflow</title>
6 </head>
7 <body>
8 <p style="background:#0000FF;width:500px;">Мәтіннің биіктігі мен ені CSS ережесімен беріледі.</p>
9 <p style="background:#FF0000;width:500px;">Мәтіннің биіктігі мен ені CSS ережесімен беріледі.</p>
10 </body>
11 </html>
```

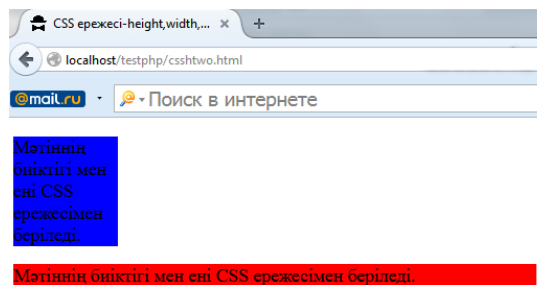
Енді осы жазылған "csshtwo.html" файлын Денвер арқылы тағы да шақырайық:



Көріп отырғаныңыздай екі абзац әп-әдемі боп парақша бетіне жайғаса қалды.Енді олардың тыныштығын бұзайық, ол үшін бірінші абзацтың енін азайтып 100 пиксел қылып өзгертейік.Және бұл өзгерістерді сол "csshtwo.html" файлда жасай берейік.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html;
5 charset=utf-8"/>
6 <title>CSS ережесі-height,width,overflow</title>
7 </head>
8 <body>
9 <p style="background:#0000FF;width:100px;">Мәтіннің биіктігі мен ені CSS ережесімен беріледі.</p>
10 <p style="background:#FF0000;width:500px;">Мәтіннің биіктігі мен ені CSS ережесімен беріледі.</p>
11 </body>
12 </html>
```

Енді осының нәтижесін көру үшін тағы да оны Денвер арқылы шақырып көрейік.

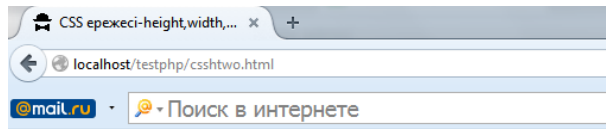


Бірінші абзац ені кішірейгенмен мәтін толық сиып тұр.

Енді осы бірінші абзацтың биіктігін тағы да өзгерту мақсатында "50px" ге дейін кішірейтейік, ол үшін тағы да "csshtwo.html" дің кодын өзгертейік:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html;
5 charset=utf-8"/>
6 <title>CSS ережесі-height,width,overflow</title>
7 </head>
8 <body>
9 <p style="background:#0000FF;width:100px;height:50px;">Мәтіннің биіктігі мен ені CSS ережесімен беріледі.</p>
10 <p style="background:#FF0000;width:500px;">Мәтіннің биіктігі мен ені CSS ережесімен беріледі.</p>
11 </body>
12 </html>
```

Оның нәтижесін көру үшін тағы да браузерде localhost/testphp/csshtwo.html деп шақырамыз:



Мәтіннің биіктігі мен ені CSS

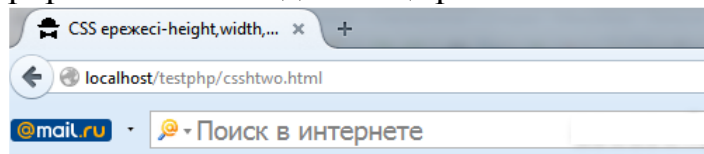
Мәтіннің биіктігі мен ені CSS ережесімен беріледі.

Міне контейнерге сымай толып кетті деген осы болады. Бірінші абзацқа тек жоғарғы екі жол ғана сиып қалғандары сымай сыртқа шығып тұр. Енді осы жерде көмекке "overflow" келеді. Абзацтың сымай қалған бөлігін былай "overflow:hidden" деп жазып, жасырып яғни көрсетпей тастауға болады.

Ол үшін тағы да "cssstwo.html" файлдың кодын былай өзгертеміз:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5 <title>CSS ережесі-height,width,overflow</title>
6 </head>
7 <body>
8 <p style="background:#0000FF;width:100px;height:50px;overflow:hidden;">Мәтіннің биіктігі мен ені
9 CSS ережесімен беріледі.</p>
10 <p style="background:#FF0000;width:500px;">Мәтіннің биіктігі мен ені CSS ережесімен беріледі.</p>
11 </body>
12 </html>
```

Бұл скриптің кодының нәтижесін тағы да браузердің адрес теретін жолында localhost/testphp/cssstwo.html деп шақырамыз.



Мәтіннің биіктігі мен ені

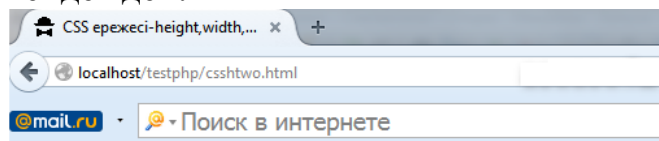
Мәтіннің биіктігі мен ені CSS ережесімен беріледі.

Екі жолдан басқасы көрінбей қалды, бірақ бізге оны толық көрсету керек болса, "overflow"-дің "auto" деген мәнін қолдану керек.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5 <title>CSS ережесі-height,width,overflow</title>
6 </head>
7 <body>
8 <p style="background:#0000FF;width:100px;height:50px;overflow:auto;">Мәтіннің биіктігі мен ені
9 CSS ережесімен беріледі.</p>
10 <p style="background:#FF0000;width:500px;">Мәтіннің биіктігі мен ені CSS ережесімен беріледі.</p>
11 </body>
12 </html>
```

Сонда ол бөлікте оның құрамын шетінде көрсетілген стрелкалар арқылы толық айналдырып көруге арналған аймақша пайда болады.


Оның қалай жұмыс істейтінін көру үшін тағы да оны Денверде шақырамыз. Нәтиже төмендегідей:



Мәтіннің биіктігі мен ені

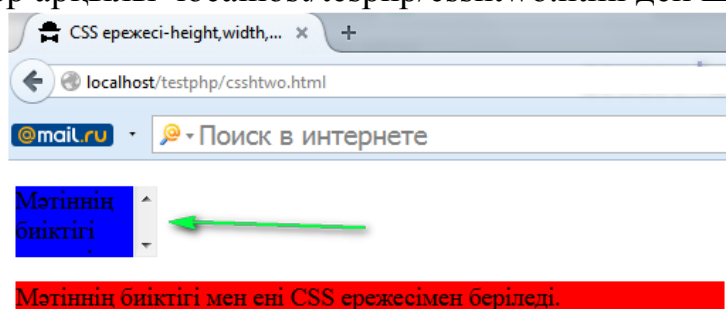
Мәтіннің биіктігі мен ені CSS ережесімен беріледі.

Ал сіз симаған мәтіннің мазмұнын тек бір жағынан ғана, яғни мына жоғарыдағы мысалда, оны тек оң жағын ғана айналдырып көргенді қалауыңыз мүмкін. Онда төменгі айналу аймағын жасыру үшін "overflow-x:hidden;overflow-y:auto"-деп екі рет сипаттап, төмендегідей түрде "csshtwo.html" кодына өзгеріс жасаймыз:



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html;
5 charset=utf-8"/>
6 <title>CSS ережесі-height,width,overflow</title>
7 </head>
8 <body>
9 <p style="background:#0000FF;width:100px;height:50px;overflow-x:hidden;overflow-y:auto;">
10 Мәтіннің биіктігі мен ені CSS ережесімен беріледі.</p>
11 <p style="background:#FF0000;width:500px;">Мәтіннің биіктігі мен ені CSS ережесімен беріледі.</p>
12 </body>
13 </html>
```

Тағы да браузер арқылы localhost/testphp/csshtwo.html деп шақырамыз:



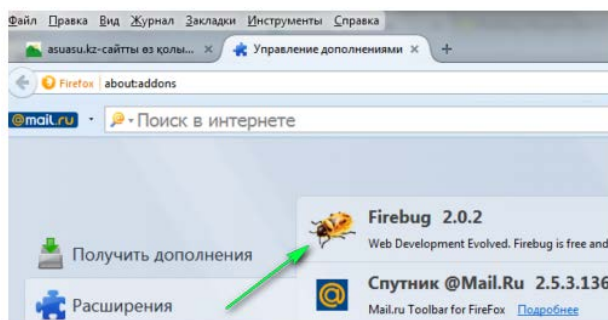
Міне , енді ойдағыдай болып шықты, яғни "overflow" ды өз мақсатымызда пайдалану өте жеңіл болып шықты.

Ал енді ең басында көрсетілген "width" бен "height" -тардың пайыздық өлшеміне тоқталатын болсақ, ол өлшем оның сыртқы контейнеріне қарасты есептелінеді. Мысалы, абзацтың ені 30% деп көрсетіліп, ал ол өзі ені 120px , болатын "div" контейнеріне орналасқан болса, онда "p" абзац "div"-тің төрттен бір бөлігіне орналасады. Және бұл көрсеткіш тек сыртқы контейнер өлшемі болғанда ғана жүзеге асады.

## Дәріс №4. CSS ережелері - "PADDING", "MARGIN" және "BORDER" туралы

Бүгінгі сабақ "padding", "margin" және "border" деген CSS ережелеріне арналады.

Ал бұл сабақты түсіндіру үшін мен сіздерге көптен бері айтқым келіп жүрген Mozilla Firefox браузерінің қосымша кіріспесі Firebug туралы қысқаша айтып кеткім келіп отыр, себебі оны пайдаланбайтын web-сайт жасаушылар жоқ сияқты. Онсыз сайт жасау мүмкін емес, себебі ол сайт жасаудағы ең басты және ең керекті құрал. Оны сіз Мозилланың жоғарыдағы менюінің ішінен Инструменты->Дополнения деген парақша бетіне төменде көрсеткенімдей кіріп, қоңыздың суретімен белгіленген Firebug расширениясын өз компьютеріңізге тегін жүктей аласыз.



Мен өзім 3 браузерді пайдаланамын: негізінде Опера және кейде Google Chrome браузерлерін ғаламторға шығу үшін, ал Мозилла Firefox браузерін сайт парақшаларын жасау үшін.

Егер қаласаңыз, оларды өз атаулары бойынша көрсетілген сілтемелер арқылы компьютеріңізге тегін жүктей аласыз.

Әзірге осы Firebug-тің тек бір қасиетін назарларыңызға ұсынбақпын, себебі ол бүгінгі тақырыпты түсінуге көп көмек бере алады.

Яғни, кез келген сайт бетін ашып, оның кодын болмаса белгілі бір тегінің сипаттамасын көргіңіз келсе, онда сол жерде мышканың оң жақ кнопкасын басыңыз керек. Осылай сіз сайттың көпшілікке көрінетін кодын тамашалай аласыз.

Оның тек осы қасиетін ғана басқа да браузерлер көрсете алады, бірақ программаны дұрыстауды (отладка) олар Firebug сияқты жақсы атқара алмайды.

Мен назарларыңызға өз парақшамдағы екінші абзацтың үстінен мышканың оң жақ кнопкасын басқандағы түсілген суретті ұсынамын:



Енді "Исследовать элемент" арқылы шыққан бетке талдау жасайық:



Бұл бетте алдарыңыздағы кодтың бір бөлігі, оның ішінде көк жолмен боялып тұрған үстінен мышкамен түртілген р абзацы. Жасыл стрелкалармен көрсетіп отырғанымның жоғарыға бағытталғаны сол абзацты көрсетіп, ал төмен қараған стрелка тіктөртбұрыштың ішіндегі сол абзацтың парақша бетіндегі орнын көрсетіп тұр.

Міне, бүгінгі сабақ осы тіктөртбұрышты талдау арқылы бүгінгі тақырып иелері "padding", "margin" және "border" сипаттамаларын түсіну болмақшы.

Яғни парақша бетіндегі кез келген HTML тегінің (тармақ тегтері және блок тегтері) осындай сипаттағы тіктөртбұрышы, демек ол тегтің "padding", "margin" және "border" сипаттамалары болады деген тұжырым туады.

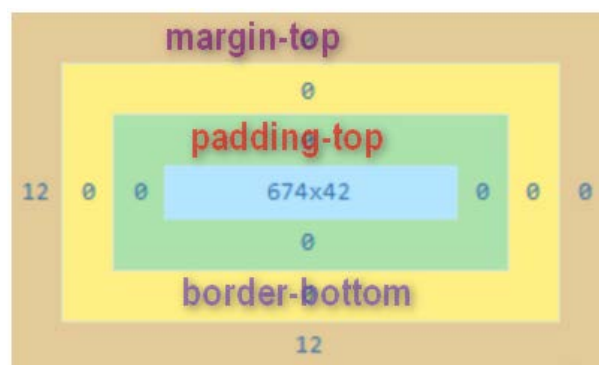
Сонымен, суреттегі тіктөртбұрыштың ең ортасындағы көк түсті бөлігі біздің р абзацымыз, ал оның сыртындағы жасыл түспен көмкерілген аймақ - "padding" (ішкі бос кеңістік) деп аталады.

Ал "padding" сыртын айнала жағалап орналасқан сары түсті аймақ - "border" (шекара) деп аталады.

Сыртқы қоңыр түсті "border"-ді қоршаған аймақ "margin" (сыртқы бос кеңістік) деп аталады.

Мысалы, парақша бетіне мәтін жазғанда болмаса сурет орналастырғанда, оны "div" не "p" контейнерлеріне салғанда ол мәтін не суреттің шекарасы - "border" пайда болады. Оны сіз қалауыңызша оған белгілі бір мән беру арқылы көрсете аласыз, ал егер оның мәнін көрсетпесеңіз ол шекара мүлде көрінбейді.

Ал "padding" мен "margin" сипаттамаларының анықтамасы басқаша, олар ана бір фильмдегі көрінбейтін адам сияқты ешқашан парақша бетінде көрінбейді. Бірақ, парақшаны беттеуде алатын орны ерекше - оларсыз сайт сайт бола алмайды.



Яғни, бұл суреттен өздеріңіз көріп отырғандай "padding-top"-HTML тегінен (мысалы, мәтіннен не суреттен) шекараға (border) дейінгі аралықты, "margin-

top"- шекарадан жоғарғы келесі HTML тегінің "margin-іне" дейінгі аралықты, "border-top" - сол шекараның жоғарыдағы енін көрсететін мөлшемдер.

Ал "padding-right"- оң жақтағы "border-right"-қа дейінгі аралықты, "margin-right" - "border-right" тан келесі көрші HTML тегінің "margin"-іне дейінгі аралықты, "border-right" - оң жақтағы шекараның енін көрсетіп тұр.

Қалғандары осылар тектес сипатталады.

Енді W3C стандарттары бекітілген сайттан олардың CSS ережелерін көшіріп алайық:

1."padding": "padding-top"- "padding-right" "padding-bottom" "padding-left"

'padding-top' 'padding-right' 'padding-bottom' 'padding-left'	<padding-width>   inherit	0
--	---------------------------	---

The properties defined in this section refer to the <padding-width>

<length> **сандық өлшем**

Specifies a fixed width.

<percentage> **пайыздық өлшем**

2."margin":

'margin-right' 'margin-left'	<margin-width>   inherit	0
'margin-top' 'margin-bottom'	<margin-width>   inherit	0

The properties defined in this section refer to the <margin-width> v

<length> **сандық өлшем**

Specifies a fixed width.

<percentage> **пайыздық өлшем**

The percentage is calculated with respect to the *width* of the *parent* block's *width* as well. If the containing block's width depends on the

**auto браузердің еркінде**

See the section on [calculating widths and margins](#) for behavior

3."border":

'border-top' 'border-right' 'border-bottom' 'border-left'	[ <border-width>    <border-style>    'border-top-color' ]   inherit
--	--

The border width properties specify the width of the [border area](#). They can take one of the following values:

**thin жіңішке**

A thin border.

**medium орташа**

A medium border.

**thick жуан**

A thick border.

<length> **сандық өлшем**

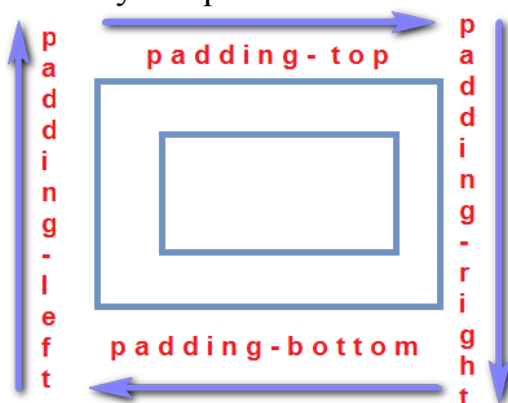
The border's thickness has an explicit value. Explicit border

"PADDING", "MARGIN" МЕН "BORDER" CSS ЕРЕЖЕЛЕРІНІҢ ЖАЗЫЛУЫ.

1. Осы аталған ережелер жазылғанда белгілі бір тәртіпке бағынады, яғни төмендегі суретте көрсетілген стрелка-нұсқағыштар көрсеткен бағыт бойынша



алдымен top-жоғарғысы, сосын right-оң жақ, одан кейін bottom-төменгісі, соңынан left-сол жақтағысы жазылуы керек.



2. Осы аталған CSS ережелерінің 4 түрлі жазылуы бар. Олар төмендегі суретте нақтылап көрсетілген, әр ереженің жанында стрелкамен оның баламасы бейнеленген. Егер сипаттамалардың арасында мәндері ұқсасы кездессе, олар ережелер баламасымен өрнектеле алады. Яғни, бұл балама ережелер сайт парақшаларының кодын жазғанда көп жеңілдік береді.

padding: padding-top, padding-right, padding-bottom, padding-left

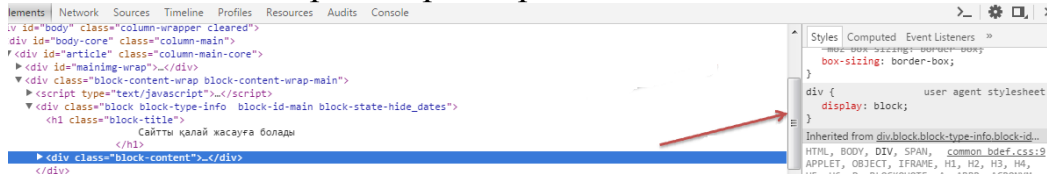
1. padding-top:10px;  
padding-right:20px;  
padding-bottom:30px;  
padding-left:40px; → padding:10px 20px 30px 40px;
2. padding:10px 20px 30px 20px; → padding:10px 20px 30px;
3. padding:10px 20px 10px 20px; → padding:10px 20px;
4. padding:10px 10px 10px 10px; → padding:10px;

## КЕЗ КЕЛГЕН САЙТ ПАРАҚШАСЫНЫҢ КОДЫН КӨРУ МҮМКІНДІГІ

Жоғарыда айтып өткенімдей, сізде Firebug болмаса да қалаған браузеріңізде кез келген сайттың кез келген жерінде тұрып мышканың оң жақ кнопкасын басып HTML тегінің жоғарыда талдап өткен блоктық макетін ғана емес, сол сияқты сайттың кодын да және оның CSS ережелерін де көре аласыз.

Мысалы, Опера браузеріндегі менің парақшамды алайық:

Енді осы окнодағы мен қызыл стрелкамен көрсеткен "Styles" деген вкладканы бассаңыз, стильдік кестелер, яғни осы белгілі бір HTML тегіне қатысты жазылған CSS ережелерін көре аласыз.



## Дәріс №5. CSS ережелері - "FLOAT" және "CLEAR"

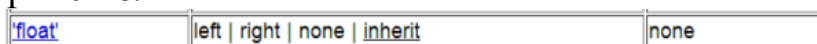
Бүгінгі сабағымда мен сайтты блоктар арқылы беттеуде кең қолданылатын CSS ережеселері - "float" және "clear" туралы мағлұмат бермекпін.

HTML тегтері "div" пен "span" туралы сабағымда "float" туралы мысал келтіріліп өткен. Онда ол "div" контейнерлерін олардың табиғи қозғалыс дағдысынан ажыратып, біреуін сол жаққа, екіншісін оң жаққа орналастырған болатын.

"Div" контейнерлерінің, ия болмаса жалпы HTML тегтерінің табиғи қозғалыс дағдысы деп олардың парақша бетінде бірінің астына бірі орналасуын айтады. Ал олардың осы табиғатын өзгертетін CSS ережелерінің бірі - осы "float".

Осы ереже арқылы жазылған тегтерді қалқитын элементтер деп атайды.

Енді оның ережелерін анықтау үшін әдеттегідей W3C стандарттарына жүгінеміз:

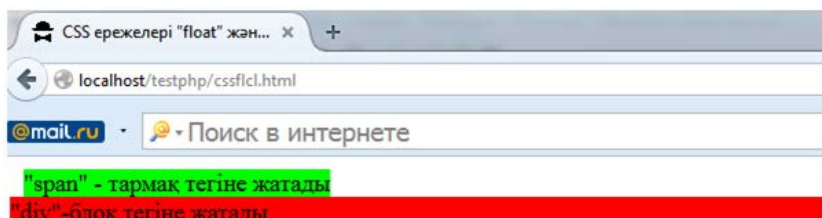


Бұл стандарттар бойынша, егер HTML тегіне CSS ережелері "float:left" болып берілсе ол сол жаққа, "float:right" жазылса оңға орналаса алады. Ал үнсіздік бойынша ол "none" мәніне ие болады.

Енді мысал ретінде жоғарыда сілтемеде көрсетілгендей бір "div" және бір "span" алайық та, былай HTML кодын Notepad ке жазып шығайық.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5 <title>CSS ережелері "float" және "clear"</title>
6 </head>
7 <body>
8 <span style="background:#00ff00;width:200px;margin:10px;">"span" - тармақ тегіне жатады</span>
9 <div style="background:#ff0000;">"div"-блок тегіне жатады </div>
10 </body>
11 </html>
```

Бұл файлды мен "cssflcl.html" деп атадым да өзімнің тәжірибеге арналған файлдарымды сақтайтын папкаға сақтап жібердім. Сіз де солай файлды өз папкаңызға сақтап болған соң оны не Денвер арқылы, не оны жай ғана түртіп жіберіп ашыңыз. Менде нәтиже былай болды, яғни "div" те, "span" да табиғи түрде жылжып, бірінің астына бірі кодта қалай жазылса солай парақ бетінде орналасты.



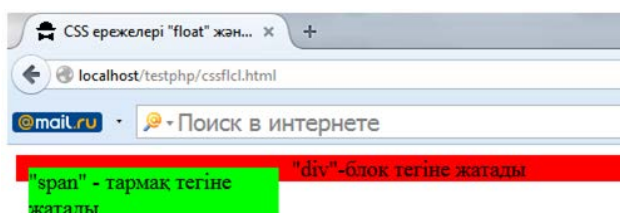
Енді "span" тегіне "float:left" деп жазып, "cssflcl.html" файлын тағы да өзгертейік:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5 <title>CSS ережелері "float" және "clear"</title>
6 </head>
7 <body>
8 <span style="background:#00ff00;width:200px;margin:10px;float:left;">
9 "span" - тармақ тегіне жатады</span>
10 <div style="background:#ff0000;">"div"-блок тегіне жатады </div>
11 </body>
12 </html>

```

Енді ол файлды ашып төмендегі суреттен "span" -ның сол жаққа орын ауыстырғанын көреміз:



Бұл жерде ескеретін мәселе мынада: "margin" тармақ тегтерінде негізінде ескерілмейді, бірақ бұл мысалда "span"- тармақ тегіне "float:left" ережесі берілген соң, ол өзгеріске ұшырап блок тегі құсап әрекет жасап отыр.

Енді жақсы делік, жасыл блок сол жаққа көшті, бірақ оны "div" байқайтын емес, сол күйі міз-бақпай орнын босатпай тұр.

### CSS ЕРЕЖЕСІ - "CLEAR".

"Div" блокты "span" тегіне тиесілі орыннан кетіру үшін басқа бір ереже қолданылады- ол ереже "clear" деп аталады.

Оны анықтау үшін тағы да жоғарыда көрсеткендей W3C стандартына жүгінеміз:

<code>clear</code>	<code>none   left   right   both   inherit</code>	<code>none</code>
--------------------	---	-------------------

Бұл ереже былай дейді және бұл сөз ағылшынша орынды не кеңістікті босату дегенді білдіреді:

1. "clear:left" - ережесі бұл тег үшін сол жақ бос болатынын көрсетіп тұр.
2. "clear:right"- ережесі бұл тег үшін оң жақ босатылатынын айқындап тұр .
3. "clear:both" -ережесі жазылған элемент екі жағын да босата алады.

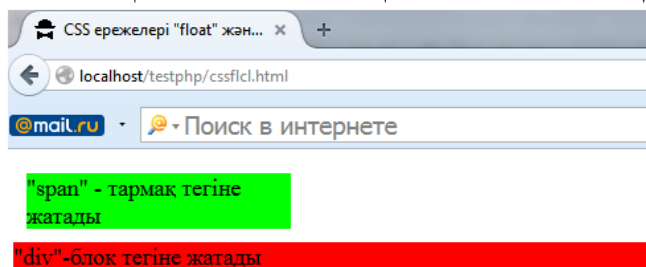
Демек, осы ережені негізге алып жоғарыдағы "span" тегі өз орнын толық меншіктей алады екен.Оған көз жеткізу үшін, жоғарыдағы кодымыздағы "div" тегіне "clear:left" деп жазайық.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5 <title>CSS ережелері "float" және "clear"</title>
6 </head>
7 <body>
8 <span style="background:#00ff00;width:200px;margin:10px;float:left;">
9 "span" - тармақ тегіне жатады</span>
10 <div style="background:#ff0000;clear:left;">"div"-блок тегіне жатады </div>
11 </body>
12 </html>

```

Енді өзгеріске ұшыраған файлымызды қайтадан не Денвер арқылы, не оны өзінің папкасында ашайық. Сонымен оның нәтижесі төмендегідей болды:



Соңында "div" қырсыққанын қойып, өз орнына көшіп тынды, себебі оған жазылған "clear:left" ережесі оған оның оң жағының бос болуын бұйырды.

Сонымен, қорыта айтқанда "float" ережесі осы қасиеттерінің арқасында блок беттеуде кеңінен қолданылады.

## Дәріс №6. CSS тізім стилін сипаттау

HTML сабақтарында мен таңбаланған және нөмірленген тізімдер туралы баяндап өткенмін.

Бірақ, олардың кейбірінің орналасуы код жазу барысында тізімдерді өзіміз қалаған түрге келтіруге қиындық жасайды. Мысалы, мен меню жасағанда `<ul>` `</ul>` ді қолданамын, бірақ мен менюде ешқандай таңбаның да, нөмірдің де болғанын қаламадым. Ал HTML дегі тізім үнсіздік бойынша әр меню тақырыбының алдына дөңгелек немесе араб сандарын жазады. Одан қалай құтылуға болады. Міне, осы кезде CSS көмекке келеді.

Не болмаса, мысалы таңбаның не нөмірлердің орнына сурет болғанын қалайтындар болса ше? Міне, осындай жағдай да CSS тің көмегімен шешіледі.

Тізім стилі оның негізгі қасиеті `list-style` арқылы сипатталады, ал ол өз алдына 3 қасиеттен құралады:

<code>list-style</code>	<code>list-style-type</code>    <code>list-style-position</code>    <code>list-style-image</code>
<code>list-style-type</code>	таңбаның түрін сипаттайды
<code>list-style-position</code>	таңбаның орнын белгілейді
<code>list-style-image</code>	таңба ретінде қолданылатын суретті сипаттайды

Тізім стилінің бұл 3 қасиеттері бірдей қолданылуы міндетті емес, тек қолданылатын қасиеттер қанша болса да осы көрсетілген ретте жазылуы тиіс және аралары пробел арқылы бөлінуі керек.

Төменгі кестеде осы стильдің 3 қасиеттері сипатталып беріліп тұр:

Қасиеті: Мәні	Сипаттамалары
<code>list-style-type: disc</code>	
<code>list-style-type: circle</code>	
<code>list-style-type: square</code>	
<code>list-style-type: decimal</code>	
<code>list-style-type: upper-roman</code>	дөңгелек (үнсіздік бойынша) шеңбер
<code>list-style-type: lower-roman</code>	квадрат араб саны
<code>list-style-type: upper-alpha</code>	жоғарғы регистрдегі рим саны
<code>list-style-type: lower-alpha</code>	төменгі регистрдегі рим саны латынның бас әріптері латынның кіші әріптері
<code>list-style-type: none</code>	ешқандай таңба жазылмайды
<code>list-style-position: outside</code>	таңба тармақтың сыртында орналасады (үнсіздік бойынша)
<code>list-style-position: inside</code>	таңба тармақтың ішінде орналасады
<code>list-style-image: none</code>	ешқандай сурет болмайды
<code>list-style-image: url</code>	сурет қолданылады

Енді мен HTML кодында CSS ті белгілейтін тәсілдің екіншісін пайдаланып, HTML кодын екі түрде төмендегідей жазып шығайын.

Біріншісінде, тізім стилі ондағы таңбаның қасиеттерін бір ғана әдіспен сипаттауға мүмкіндік береді, соны қолданайын.

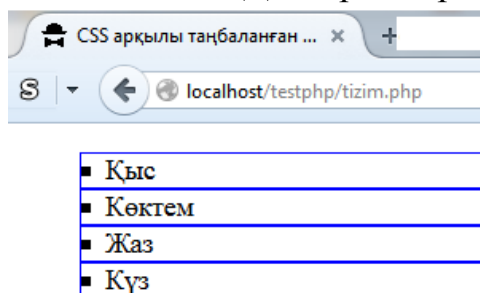
Ал екінші код сол стильдің қасиеттерін бөлек-бөлек сипаттайды.

Екінші қасиет бойынша жазылған "таңба тармақтың сыртында не ішінде орналасады" дегенді түсіну үшін, тармақтың шекарасын CSS арқылы сипаттап тізім стилінің жанына жазып қоялық:

1. Таңба тармақтың ішінде орналасады:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>CSS арқылы таңбаланған тізім</title>
<style type="text/css"> ul { list-style: square inside; /* Таңба квадрат түрінде
тармақтың ішінде орналасады*/ } li { border: 1px solid blue; /* Тармақтың
шекарасын ені 1px көк түспен белгілейік*/ }
</style>
</head>
<body> <ul> <li>Қыс</li> <li>Көктем</li> <li>Жаз</li> <li>Күз</li> </ul>
</body>
</html>
```

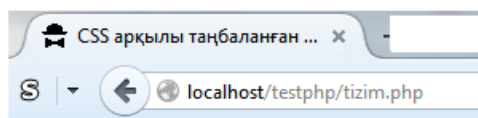
Енді осы HTML кодының нәтижесін Денверде қаралық:



2. Таңба тармақтың сыртында орналасады:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>CSS арқылы таңбаланған тізім</title>
<style type="text/css"> ul { list-style: square; /* Таңба квадрат түрінде
белгіленеді */ list-style-position: outside; /* Осы таңба тармақтың сыртына
орналасады */ } li { border: 1px solid blue; /* Тармақтың шекарасын ені 1px көк
түспен белгілейік*/}
</style>
</head>
<body> <ul> <li>Қыс</li> <li>Көктем</li> <li>Жаз</li> <li>Күз</li>
</ul></body>
</html>
```

Енді осы HTML кодымыздың нәтижесін Денвер арқылы тамашалайық.

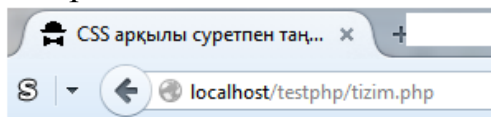






- Қыс
- Көктем
- Жаз
- Күз

Енді CSS тізім стилінің үшінші қасиетін талдайық. Ол үшін HTML кодын жазып, таңба орнына өзіміз қалаған суретті орнаталық. Демек, бізге өте кішкене сурет керек. Мен ол үшін өз сайтымдағы қазақша өрнектелген суретті алып, оның мөлшерін 20x16 ге кішірейтіп алдым. Ол суретті сіздер де қаласаңыздар өздеріңізге көшіріп алыңыздар.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>CSS арқылы суретпен таңбаланған тізім</title>
<style type="text/css"> UL { list-style-image: url(images/cssornament.png); /*
Таңба сурет түрінде белгіленеді */ }
</style>
</head>
<body> <ul> <li>Қыс</li> <li>Көктем</li> <li>Жаз</li> <li>Күз</li> </ul>
</body>
</html>
```

Енді кодтың нәтижесін көрелік:



-  Қыс
-  Көктем
-  Жаз
-  Күз