Раздел: CSS – каскадни набори от стилове

1. <u>Същност. Видове CSS.</u>

CSS е съкращение от Cascading Style Sheets и представлява отделен език, съдържащ множество "инструменти", с които може да се влияе на външния вид на HTML страниците. От определена гледна точка CSS е нещо като "надстройка" на HTML.

CSS предлага големи удобства и улеснения при изграждането на даден HTML документ. В много случаи е по-добре даден елемент от HTML страница да бъде създаден с помощта на CSS, отколкото чрез ползване на добре познатите HTML тагове. Най-голямото улеснение, което предлага CSS е свързано с контрола на голям набор HTML документи, като контролирането на външния вид на страниците става чрез промяна на един единствен файл - CSS файла, без да е нужно да се променя HTML кода във всяка една от HTML страниците.

Например:

Да предположим, че имате сайт, съставен от 50 HTML страници. Ако ползвате тага **** за да задавате вида на шрифта, големината и цвета му, ще трябва във всяка една html страница да пишете кодове:

някакъв текст Също така ще се наложи за всяка една страница да дефинирате например цвета на фона чрез атрибута bgcolor на тага body: < body bgcolor="#000000">

< body bgcolor="#000000"; и т.н...

Ако някога решите да промените някои от тези параметри ще ви се наложи да отворите сорсовете на всичките 50 страници и да въведете промените във всеки един от тях. Цялото това усилие може да си го спестите като направите един CSS файл и разположите в него всички нужни параметри. След това трябва само да сложите по един линк към CSS файла във всяка от примерните 50 HTML страници, за да бъдат валидни зададените параметри за всяка една от тях.

Има три начина на използване на CSS:

Първи - външен CSS файл. В този случай всичкият CSS код се пише във външен файл, към който по-късно се дава линк във HTML страницата. Този начин обикновено се ползва, когато са налице по-дълги и по-обемни CSS кодове.

Пример 1: външни CSS

Отворете някакъв текстов редактор, например Notepad и напишете в него следния код:

body {
background-color: #000000;
font-family: arial, sans-serif, helvetica;
font-size: 12px;
color: #cccccc
}

След това съхранете файла като му зададете разширение css - например style.css

Cera остава във всяка HTML страница в секцията **head** да сложите следния код: <link rel=stylesheet type="text/css" href="style.css" />

```
<html>
<head>
<title>CSS Test</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
I'm testing remotely linked cascading style sheets!
</body>
</html>
```

Предимството на този начин е, че можете да използвате един CSS файл за няколко (или множество) HTML файлове.

Когато се наложи да редактирате CSS кода вие ще го редактирате само веднъж!

Втори - вътрешни CSS кодове, които да бъдат поставени в HTML файловете ви. Това означава, че всяка страница има свой собствен CSS код, който се намира в нея.

Пример 2: вътрешни CSS

```
<html>
<head>
<title>My First CSS site</title>
<style type="text/css">
body {
font-family: verdana, arial, serif;
font-size: 11px;
color: #000000;
}
</style>
</head>
<body>
Text
</body>
</html>
```

Ако пробвате ще видите, че текстът в BODY секцията ще бъде 11px Verdana. Така се ползва CSS директно в HTML файла.

Забележете, че синтаксисът е абсолютно същия!

Трети - in-line CSS - директно на местата където ви трябва, при останалия HTML код.

Пример 3: in-line CSS

```
<html>
<head>
 <title>Testing in-line styles</title>
</head>
<body style="background-color: #D5D5D5;">
  I'm testing in-line styles!
  <span style="color: #0000CC;">
  This text is blue!
  </span><br />
  This text is not!
  <span style="color: #CC0000;">
  This text is red!
  </span><br />
  Hey look! This text isn't!
  </body>
</html>
```

Въведеният CSS код променя определените части от HTML файла.

Трите начина могат да бъдат комбинирани.

2. <u>Структура на правилата на CSS</u>

- а. <u>Селектор (selector)</u>
- b. <u>Атрибут (property)</u>
- с. <u>Стойност (value)</u>

Структурата на CSS файла включва **названието на елемента**, за който ще се задават параметрите - в примера това беше **body**, и след това самите **параметри**, които се ограждат в големи скоби - **{ }**. Практически всеки таг от HTML - например table, p, h1, h2 и т.н. - може да бъде на мястото на body и да му се зададат някакви параметри. HTML таговете, за които се задават параметрите, се наричат в CSS **селектори**, а кода, поставен в големите скоби, се състои от **атрибути** и зададените им **стойности**. Т.е. структура на CSS кода е

селектор {атрибут:стойност}

В скобите може да се затвори един атрибут или няколко атрибута с техните стойности, като в такъв случай те се отделят с точка и запетая.

Например:

```
body {
background-color: #000000;
font-family: arial, sans-serif, helvetica;
font-size: 12px;
color: #cccccc
}
```

В този случай имаме един **селектор** - **body**. Кода, заключен между двете скоби, е съставен от атрибутите background-color, font-family, font-size и color, а на всеки атрибут е зададена някаква стойност -в случая например стойността на атрибута color е #сссссс и т.н. Тъй като имаме зададени няколко атрибута с техните стойности, те са отделени един от друг чрез точка и запетая.

Тъй като в този пример селектора е body, то всички зададени параметри ще важат за секцията body на HTML страницата. Ако на мястото на body сложим например **селектора table**, тогава декларираните атрибути и техните стойности ще важат за всяка таблица, която имаме в HTML документите:

```
table {
background-color: #000000;
font-family: arial, sans-serif, helvetica;
font-size: 12px;
color: #cccccc
}
```

Специално що се отнася до декларирането на цветовете има и още едно улеснение. В случая на атрибута background-color е зададена стойност #000000 (черен цвят) и на атрибута color е зададена стойност #сссссс (бледосив цвят). Те биха могли да се декларират и така:

background-color: #000; color: #ccc

Като по този начин ще се спести малко пространство ,което може да се окаже и голямо, ако се пише дълъг документ с много код за цветове. CSS позволява двойките еднакви букви или цифри от 16-тичната бройна система да се заменят с едната от тях, т.е. кода #ff0000 може да се напише #f00,кода #334455 може да бъде #345 и т.н.

Пример 4:

- 1. Препишете в тетрадките си HTMI текста.
- 2. Оградете селекторите, подчертайте с плътна черта атрибутите, а с прекъсната стойностите им.
- 3. Отворете Notepad и въведете тескта.
- 4. Съхранете файла с разширение mycss.css
- 5. Отворете нов Notepad, сложете в него втория HTMI код и съхрането под име 4.html

```
body {
margin-top: 5px;
margin-left: 0px;
margin-right: 0px;
background-color: #efefef;
font-family: times new roman, timoku;
font-size: 18px;
color: #000000
}
h1 {
font-family: impact;
font-size: 32px;
color: #bb0000;
text-align: center
}
a {
color: #0000aa;
font-size: 16px;
font-family: arial, helvetica, verdana, sans-serif;
font-weight: bold
}
a:link {
text-decoration: none
}
a:visited {
text-decoration: none;
color: #5500ff
}
a:hover {
```

```
text-decoration: underline;
color: #aa0000
}
a:active {
text-decoration: none;
color: #77aaCC
}
td {
color: #222222;
font-size: 12px;
font-family: arial, helvetica, verdana, sans-serif;
padding-left: 5px;
padding-right: 5px;
padding-top: 20px;
padding-bottom: 20px
}
< html>
< head>
< title>CSS Example</title>
< meta http-equiv=Content-Type content="text/html; charset=windows-1251" />
< link rel=stylesheet type="text/css" href="mycss.css" />
</head>
< body>
Това е текст в секцията body, извън таблицата, със шрифт Times new roman.
< h1>Това е центрирано червено заглавие със шрифт Impact</h1>
Това е текст в 1-ва клетка със шрифт Arial
Това е текст във 2-ра клетка, а това е <a href="http://www.uroci.com">Връзка към
uroci.com</a>, която
е с по-голям размер на шрифта и има следните ефекти: цвета и е тъмносин, без
подчертаване,
при слагане на курсора върху нея цвета се променя на червен и се появява долна черта.
Това отново е текст извън таблицата.
```

```
</body>
```

</html>

3. Стойности на свойствата

1. <u>Цвят</u>

2. Размери

Задача: От текста в пример 4 открийте и препишете в таблицата използваните стойности за цвят и размер.

<u>Стойности за цвят</u>	<u>Стойности за размер</u>

4. Стилове за шрифт

Контрола на текста вCSS се извършва чрез следните атрибути и съответните им стойности:

1. font-family - Задава шрифт на текста. Възможните му стойности са названието на един или няколко шрифта, отделени със запетаи (например font-family: arial, sans-serif, helvetica, verdana).

2. font-size - Задава големина на текста. Възможните му стойности са цифрови - в пиксели (px) или пойнтове (pt) (например font-size: 14px).

3. font-weight - Задава удебеляване или "изтъняване" на текста. Може да приема следните стойности:

normal- показва нормален текст bold - удебелява текста bolder - удебелява текста повече от bold lighter - изтънява текста 100 - изтънява текста (същия ефект като lighter) 200 300 400 - прави текста какъвто е подразбиране(същия ефект като normal) 500 600 700 - удебелява текста (същия ефект като bold) 800 900 - удебелява максимално текста (същия ефект катоbolder)

4. text-align - Задава позиционирането на текста в страницата. Възможните му стойности са:

left - подравнява текста вляво(същата е стойността по подразбиране)
 right - подравнява текста вдясно
 center - центрира текста
 justify - подравнява текста едновременно вляво и вдясно

5. color - Задава цвят на текста. Възможните му стойности са цветове, описани чрез 16тичната бройна система, чрез английските им названия или чрез задаване на RGB стойност. Например бял цвят ще се зададе - чрез название: color: white

- чрез 16-тична стойност: color: #ffffff

- чрез RGB стойност: color: rgb(255,255,255)

6. background-color - Задава цвят за фон на текста. Възможните му стойности са цветове, описани чрез 16-тичната бройна система, чрез английските им названия или чрез RGB стойност, подобно на атрибута color.

7. text-decoration - Задава допълнителен ефект (украса) на текста. Възможните му стойности са:

underline - подчертава текста с долна черта overline - подчертава текста с горна черта line-through - зачертава текста blink - кара текста да премигва none - показва нормален текст, като премахва всички ефекти от текста (например премахва подчертаването на връзките)

8. letter-spacing - Задава разстояние между буквите на текста. Възможните му стойности са normal или зададено с цифри разстояние между буквите. Цифровите стойности могат да бъдат указани в пиксели (например letter-spacing: 5px) или сантиметри (например letter-spacing: 0.3cm). Тези стойности могат да бъдат и с отрицателен знак - в този случай буквите от текста се сбиват, тъй като разстоянието между тях се намалява (например letter-spacing: -2px).

9. word-spacing - Задава разстояние между думите в текста. Възможните му стойности са **normal** или **зададено с цифри** разстояние между думите. Цифровите стойности могат да бъдат указани в пиксели или сантиметри и да приемат отрицателен знак, подобно на стойностите на атрибута letter-spacing.

10. white-space - Задава дали текста да бъде показан точно както е написан в текстовия редактор, т.е. дали да се покажат всички "спейсове" и "ентери" - разстоянията между буквите, думите и редовете,както са написани например в Notepad или браузъра да игнорира оставените бели пространства в текста и да го покаже компактен. Възможните му стойности са:

normal - показва текста като игнорира оставените бели пространства pre - показва текста точно както е написан в текстовия редактор, какъвто е ефекта от HTML тага

nowrap - не позволява текста да се пренесе на по-долен ред докато не се постави тага за прекъсване
. Ако текста е достатъчно дълъг в долната част на браузъра ще се появи лента за скролиране.

11. text-indent - Използва се при започване на нов абзац в текста. Оставя разстояние между лявата страна на страницата и началото на текста от първия ред на абзаца. Възможните му стойности са цифрови - в пиксели (px), сантиметри (сm) или проценти (%). 12. text-transform - Задава на текста главни или малки букви. Възможните му стойности ca:

uppercase - задава на целия текст главни букви lowercase - задава на целия текст малки букви capitalize - задава като главна началната буква на всяка дума от текста none - показва текста във вид по подразбиране

13. direction - Задава посоката, в която тече текста.Възможните му стойности са:

Itr - текста тече от ляво на дясно rtl -текста тече от дясно на ляво

Пример 5:

1. Отворете Notepad и въведете тескта.

```
h1
{
font-family: arial, helvetica, sans-serif;
font-size: 36px;
color: #ff0000;
text-align: center;
text-transform: capitalize;
text-decoration: overline;
letter-spacing:-3px;
word-spacing: 1cm
}
```

- 2. Съхранете файла с разширение myfile.css
- 3. Отворете нов Notepad, сложете в него втория HTMI код и съхрането под име 5.html

<html> < head>

- < title>CSS Example</title>
- < meta http-equiv=Content-Type content="text/html; charset=windows-1251" />
- < link rel=stylesheet type="text/css" href="myfile.css" />
- </head>
- < body>
- < h1>това е примерно заглавие</h1>
- </body>
- </html>

- 4. Създайте ваши стилове на заглавия h2, h3 и h4. Съхранете файла с разширение **myfile1.css.**
- 5. Като използвате пример 5 демонстрирайте ги. Новия код съхранете като 6.html.
- 6. Като използвате кода от пример 7, създайте ваши стилове.

Пример 7:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>Text CSS</title>
<style>
p.style1 {
      font-family:Arial, Helvetica, sans-serif;
      font-size:12pt;
      font-style:italic;
      line-height:200%;
      text-align:justify;
      text-indent:60px;
}
p.style2 {
      letter-spacing:5px;
      text-transform:capitalize;
}
p.style3 {
      font-weight:900;
}
p.style4 {
      font-weight:bold;
}
p.style5 {
      text-decoration:underline;
}
</style>
</head>
<body>
 Този текст е с размер 12pt, наклонен, разстоянието
между редовете е 150%, тоест 24pt, двустранно подравнен. Първият ред е отместен с 60
пиксела.</р>
Този текст е с разстояние между буквите 5 пиксела
и всяка първа буква на всяка дума е голяма.
Този текст е средно почернен
Този текст е доста почернен
```

Този текст е подчертан </body> </html>

5. Блокове със съдържание. Позициониране

Най- често използваният метод за обединяване на съдържание в блокове е да поставим различни елементи от документа между таговете **<div>** и **</div>** Можем да присвояваме на тага div набори от стилове, които да управляват позиционирането на този блок, както и други негови характеристики.

Пример 8:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>DIV таг</title>
<style>
div.one {
         position:absolute;
         padding-top:25px;
         padding-bottom:25px;
         padding-right:50px;
         padding-left:50px;
         background-color:#FFCC00;
         left: 43px;
         top: 20px;
}
div.one p {
         font-weight:bold;
}
div.two {
         position:absolute;
         padding:25px 50px 25px 50px;
         background-color:#993366;
         left: 281px;
         top: 20px;
}
div.two p{
         font-style:italic;
         color:#FFFFF;
}
div.three {
         position:absolute;
         padding:30px 0 0 60px;
         background-color:#99CCCC;
         left: 43px;
         top: 200px;
```

```
width:395px;
}
div.three p {
        font-variant:small-caps;
        line-height:250%;
}
</style>
</head>
<body>
<div cla class="one">
Първи параграф
<img src="../images/e-mail.jpg" width="35" height="35" alt="Плик" />
</div>
<div class="two">
Втори параграф
<img src="../images/e-mail.jpg" width="35" height="35" alt="Плик" />
</div>
<div class="three">
Трети параграф
Щтракнете тук, за да отворите <a href="http://google.com/" target=" blank">google.com</a>
</div>
</body>
```

</html>

Позициониране на блокове с елементи

Разгледайте наборите от стилове, които използвахме, за да позиционираме блоковете с елементи, както и да форматираме текста в тях.Първият от тях гласи:

div.one {

position:absolute; /* позцията на този блок е относителна спрямо съдържашия го блок, ако няма съдържащ блок, то спрямо <html> блока. */

padding-top:25px; /* празното място от горната граница на блока до съдържанието му е 23px */

padding-bottom:25px;

/* празното място от долната граница на блока до съдържанието му е 23px */

padding-right:50px; /* празното място от дясната граница на блока до съдържанието му е 23px */

padding-left:50px; /* празното място от лявата граница на блока до съдържанието му е 23px */
background-color:#FFCC00; /* зададен е фоновият цвят в блока */
left: 43px; /*разстоянието на блока от левият ръб на прозореца или външния съдържащ го блок е 43px */
top: 20px; /*разстоянието на блока от горният ръб на прозореца или външния съдържащ го блок е 43px */

Забележка: Обърнете внимание, че в класове two и three сме описали съкратено четирите padding свойства по този начин:

padding:25px 50px 25px 50px;

Това значи:

}

- padding-top 25px
- padding-right 50px
- padding-bottom 25px
- padding-left 50px

Разрешени са и записи от вида:

```
padding:25px 50px;
/* горната и долната граница са по 25px, а лявата и дясната по 50px */
```

padding:25px; /* всички граници са по 25px */

padding:25px 30px 50px; /* зададени са стойности съответно на горна, дясна и долна граница */

Със следващият клас описваме как ще изглежда параграф в блока с име опе

div.one p {
 font-weight:bold;
 /* във всеки параграф в блок с клас **опе** текста ще е удебелен */
}

Рамка

CSS Свойства на рамка	Описание					
Основни свойства						
border-color	 Цвят на рамката, приема вид цвят зададен по един от начините: име на цвят - например border-color: "red"; RGB задаване на цвят- например border-color:"rgb(255,0,0)"; шестнадесетична стойност на цвят- например border- color:"#ff0000" 					
border-width	Ширина на рамката, приема стойности в пиксели (px), например:					
	border-width:5px;					
border-style						

Съвет: Можете да зададете и трите свойства на рамката: цвят, дебелина и стил заедно, на един ред, ето така: **border:3px solid red;** - първата стойност е за **border-width**, втората за **border-style**, третата за **border-color.**

Можете да задавате трите свойства по отделно на горната, долната, лявата и дясната рамка, ето така:

border-top:3px double red; border-bottom:3px dotted green; border-left:3px dashed #FF6600; border-right:3px double #CC00CC;

Последователността отново е: ширина на рамката, стил на рамката, цвят на рамката

Разгледайте следващият пример:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>Border</title>
<style>
p.b1
{
border-style:double;
border-width:5px;
}
```

```
p.b2
{
border-top:3px double red;
border-bottom:3px dotted green;
border-left:3px dashed #FF6600;
border-right:3px double #CC00CC;
}
table td, th {
border: 3px dotted #9900CC;
}
table th.total {
border-top:5px double #FF0000;
border-bottom: 3px dotted #9900CC;
border-left: 3px dotted #9900CC;
border-right: 3px dotted #9900CC;
}
</style>
</head>
<body>
Първи параграф
Bmopu параграф<br />Bmopu парграф
Koлoнa 1
 Koлoнa 2
 Koлoнa 3
100
 200
 300
100
 200
 300
200
 400
```

600	

6. Разположение на таблици. Форматиране

HTML код на style.css

```
table.colored_table td{
        color:#000000;
        background-color:#FFFFFF;
        padding:10px;
        border:double;
        border-color:#006633;
}
table.colored_table th {
        color:#FFFFF;
        background-color:#FF9900;
        padding:10px;
        border:double;
        border.color:#006633;
}
```

HTML код за таблицата

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
k href="style.css" rel="stylesheet" type="text/css" />
<title>HTML Шарена таблица</title>
</head>
<body>
     Фирма
         Разходи
        Януари
         Февруари
         Mapt
        "Едно" ООД 
         500
         800
         900
        "Две" АД 
         900
         1000
         750
```

</body> </html>

Резултат:

A urano	Разходи		
Фирма	Януари	Февруари	Март
"Едно" ООД	500	800	900
"Две" АД	900	1000	750

7. Форматиране на списъци и фон

1. CSS списъци

<u>Списъците в HTML</u> биват два вида - подредени (номерирани) и неподредени (без номерация). CSS Ви дава възможност да променяте символите, числата или буквите, които се използват за отбелязване на отделните елементи на списъците. Можете дори да задавате свои картинки за символ на списък.

1. Вид на символ за списък

Вид на символ за списъчен елемент се задава със свойството list-style-type, а възможните стойности са следните:

Неподредени списъци

- попе без символ
- disc запълнен кръг (това е символа по подразбиране)
- circle незапълнен кръг
- square квадрат

Подредени списъци

- decimal числа с арабски цифри: 1, 2, 3...
- decimal-leading-zero като decimal, но с допълваща нула за числата от 1 до 9, напр. 01, 02, 03
- lower-latin малки латински букви:a, b, c, d, e...
- **lower-greek** малки гръцки букви: α, β, γ, δ...
- lower-roman числа с латински цифри, малки: i, ii, iii, iv, v...
- upper-latin главни латински букви: A, B, C, D, E...
- upper-roman числа с латински цифри, главни: I, II, III, IV, V...

2. Позициониране на списък

Позицията на списъка може да се управлява допълнително със свойството list-style-position. Възможните стойности са:

- outside стойността по подразбиране
- inside ще зададе допълнителен отстъп за списъка

3. Използване на собствено изображение

Вместо няколкото символа, които Ви предлагат браузърите, можете да използвате собствено изображение за отбелязване на списъчни елементи. Това се прави по следния начин:

ul { list-style-image:url(kartinka.gif) }

На мястото на kartinka.gif сложете името на файла, който ще използвате.

Зададена по този начин, картинката ще се показва с малко по-различна позиция в зависимост от използвания браузър. За да постигнете еднакъв стил, независим от браузъра, който ползват посетителите на страниците Ви, трябва да укажете фоновото изображение не за списъка като цяло, а за отделните му елементи:

ul {
list-style-type:none;
padding:0px;
margin:0px;
}
li {
background-image:url(kartinka.gif);
background-repeat:no-repeat;
background-position:0px 5px;
padding-left:17px;
}

За списъка като цяло не се указва изображение (list-style-type:none), а свойствата padding и margin са със стойност 0, за да няма различия в резултата между браузърите.

За отделните списъчни елементи (li) е указано изображението, което разбира се не трябва да се повтаря (background-repeat:no-repeat). C background-position свойството точно можете да позиционирате изображението, а padding-left е необходимо, за да се отмести текста вдясно, иначе ще се получи наслагване на текст и изображение.

4. Съкратен запис на стил за списъци

Стиловете за списъците могат да бъдат задавани и със съкратен запис:

ol {list-style: lower-latin inside:}

ul {list-style: inside image-url(kartinka.gif);}

Стойностите на list-style се отделят помежду си само с интервал и се задават в реда type, position, image, но не е нужно всичките 3 да бъдат указвани едновременно.

CSS фон (background)

Със свойството фон (background) може да се задава фонов цвят, фоново изображение, както и да се управлява изгледа на този фон - повторение по хоризонтала или вертикала, позиция и поведение при скролиране на страницата.

1. цвят

Фонов цвят се задава със свойството background-color и стойности:

- transparent (прозрачен);
- име на цвят, напр. chocolate;

- RGB стойност примерно rgb(120, 230, 180);
- HEX стойност #FF7F50;

Например можем да зададем за заглавие h1 фонов цвят тюркоазено синьо по следния начин:

h1 {background-color: #40E0D0}

2. Фоново изображение

Страниците могат да бъдат разнообразени и с използване на подходящо фоново изображение. За да сложите снимка за фон на страницата, използвайте свойството background-image със стойност url и името на снимката:

body {background-image: url(smile.gif)}

Не забравяйте, че в интернет се използват три вида изображения: GIF, PNG, JPEG. Също така, внимавайте фоновото изображение да не вземе превес над текста - твърде ярки цветове може да затруднят четенето.

3. Повторение на фоново изображение

Голямо фоново изображене, което да покрива цялата страница рядко се използва - това би направило страницата твърде тежка. Като резултат малко хора биха дочакали да се заредят няколкото мегабайта и вместо в предимство, фоновото изображение ще се окаже недостатък. Вместо цели снимки, често се използва хоризонтално и/или вертикално повторение на по-малки изображения. За целта се използва свойството background-repeat и някоя от следните стойности:

- repeat повторение по хоризонтала и по вертикала до запълване на страницата
- repeat-x повторение само по хоризонтала
- repeat-y повторение само по вертикала
- no-repeat без повторение

4. Позициониране на фоново изображение

Със свойството background-position имате контрол над мястото на страницата, където ще се появи фоновото изображение. Можете да избирате измежду следните стойности за позициониране на фоново изображение:

• две числови стойности - първата за разстояние от левия край на страницата, а втората - за разстояние от горния край, напр.

body {background-position: 200px 20px}

- проценти (%) аналогично на числовите стойности се използват две числа -проценти част от широчината и от височината на страницата
- left top положение горе вляво, еквивалентно на 0% 0%
- center top горе в средата, същия резултат има и 50% 0%
- right top горе вдясно, може да се използва вместо 100% 0%
- left center вляво по хоризонтала и в средата по вертикала (0% 50%)

- center center точно в средата на страницата (50% 50%)
- right center вдясно по хоризонтала и в средата по вертикала (100% 50%)
- left bottom долу вляво (0% 100%)
- center bottom долу в средата (50% 100%)
- right bottom долу вдясно (100% 100%)

5. Фиксиране на фоново изображение

Можете да застопорите фоновото изображение, така че дори да превъртате страницата, то да си стои на едно и също място на екрана. Това фиксиране на фоново изображение се постига със следната команда:

background-attachement: fixed;

6. Съкратен запис на фоновите свойства

Изброените до тук фонови свойства може да се задават по съкратен начин, използвайки свойството background. Ето един пример, който задава фонов цвят (близък до оранжево), фоново изображение на име grass.gif, забрана за повторение на изображението, позициониране горе вдясно и фиксиране на изображението:

body {background: #F4A460 url(grass.gif) no-repeat right top fixed;}

8. Създаване на динамични ефекти

Създайте файл с име style1.css отворете го и напишете следния код:

#zagl {position: absolute; top: 10px; left: 185; width: 500px; text-align: center; color: black; font-family: comic sans MS}

#pix {position: absolute; top: 0px; left: 0px}

#pix2 {position: absolute; top: 0px; left: 700px}

dt {margin: 10px}

img {border: none; width: 115px}

#pix a img.skr {width: 500px; position: absolute; top: 60px; left: 165px; visibility: hidden}

#pix a:active {background: gray}

#pix a:active img.skr {visibility: visible}

#pix2 a img.skr {width: 500px; position: absolute; top: 60px; left: -515px; visibility: hidden}

#pix2 a:active {background: gray}

#pix2 a:active img.skr {visibility: visible}

После създайте файл например, gallery.html и напишете този код вътре:

```
<html>
< link href="style2.css" rel="stylesheet" type="text/css">
< head>
< title>Галерия</title>
< meta http-equiv="Content-Type" content="text/html; charset="windows-1251">
</head>
< body leftmargin="0" topmargin="0">
< h3 id="zagl">Галерия.</h3>
< dl id="pix">
< dt><a href="http://uroci.net/#"><img src="img1.jpg">
< img class="skr" src="img1.jpg">
</a>
< /dt>
< dt><a href="http://uroci.net/#"><img src="img2.jpg">
< img class="skr" src="img2.jpg">
< /a>
< /dt>
< dt>
```

```
< a href="http://uroci.net/#"><img src="img3.jpg">
< img class="skr" src="img3.jpg">
</a>
</dt>
< dt>
< a href="http://uroci.net/#"><img src="img4.jpg">
< img class="skr" src="img4.jpg">
</a>
< /dt>
< dt>
< a href="http://uroci.net/#"><img src="img5.jpg">
< img class="skr" src="img5.jpg">
</a>
</dt>
< dt>
< a href="http://uroci.net/#"><img src="img6.jpg">
< img class="skr" src="img6.jpg">
</a>
< /dt>
</dl>
< dl id="pix2">
< dt>
< a href="http://uroci.net/#"><img src="img7.jpg">
< img class="skr" src="img7.jpg">
</a>
</dt>
< dt>
< a href="http://uroci.net/#"><img src="img8.jpg">
< img class="skr" src="img8.jpg">
</a>
</dt>
< dt>
< a href="http://uroci.net/#"><img src="img9.jpg">
< img class="skr" src="img9.jpg">
< /a>
< /dt>
< dt>
< a href="http://uroci.net/#"><img src="img10.jpg">
< img class="skr" src="img10.jpg">
</a>
</dt>
< dt>
< a href="http://uroci.net/#"><img src="img11.jpg">
< img class="skr" src="img11.jpg">
</a>
</dt>
< dt>
< a href="http://uroci.net/#"><img src="img12.jpg">
< img class="skr" src="img12.jpg">
</a>
```

< /dt> < /dl> < /body> < /html>

9. ПРАКТИЧЕСКИ ТЕСТ ВЪРХУ РАЗДЕЛ 2