

Уеб системи и технологии

Тема 4:

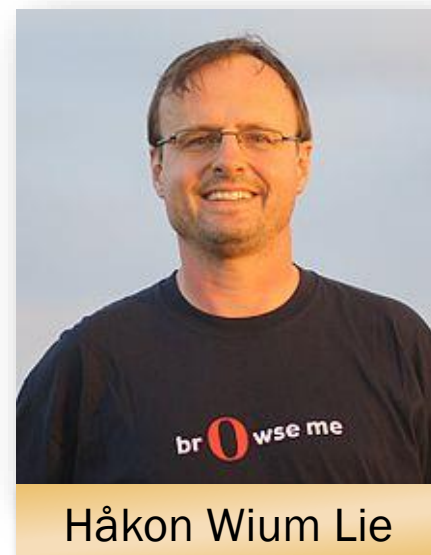
Работа с каскадни стилови шаблони
CSS. Начини на използване на CSS в
HTML документи.



доц. д-р Радослава Кралева

1. Въведение в CSS

- ✎ За първи път стиловете са въведени в HTML 4, тъй като тогава се е появила необходимостта от допълнително форматиране на уеб документите.
- ✎ Този нов дескриптивен език за описание на набори от стилове, наречен CSS (Cascading Style Sheets), е предложен от норвежеца Håkon Wium Lie (Хокон Уиум Ли).
- ✎ По времето на създаването на CSS, Lie работи в CERN заедно с Tim Berners-Lee (създателят на спецификациите HTML, URIs, HTTP и директор на W3C) и Robert Cailliau (създателят на World Wide Web).
- ✎ Първите стилови набори стандартът CSS 1 (CSS level 1 Recommendation) се появява през 1996 г.
- ✎ Дълго време втората версия на стандарта CSS 2, заема основна позиция в средите на уеб разработването, днес обаче тя е заменена с новата и по-модерна версия - CSS 3.



1. Въведение в CSS

- ✎ CSS е съкращение на Cascading Style Sheets.
- ✎ На български се превежда като
 - „каскадни набори от стилове“,
 - „таблица със стилове“ или
 - „каскадни стилови шаблони“.
- ✎ Всеки стилев набор съдържа описание за начина на представяне на един HTML елемент (таг).
- ✎ С други думи каскадните набори от стилове определят как да се покажат HTML елементите в браузърите.

1. Въведение в CSS

∞ **Връзката** между HTML елемента и стила може да се зададе **явно** или **неявно**:

- При **явната връзка** чрез дефиниране на уникално име се указва точно към кой елемент от HTML документа се отнася стила;
- При **неявната връзка** стилът се отнася до всички HTML елементи, които са указани със съответния таг.

1. Въведение в CSS

☞ Съществуват различни начина на употреба на стилови набори:

- **Вътрешни стилове:**

- **Вградени (in-line) стилове:** Добавят се като свойство на атрибута `style` на всеки отделен HTML елемент;
- **Вътрешни (embed) стилове:** Добавят се в заглавната част на HTML документа, вътре в тага `<style>...</style>`.

- **Външни (external) стилове:** Всички необходими стилове се описват и съхраняват във външен файл, който е с разширение `*.css`.

- Външен стилев файл: Този файл се включва към HTML документа, чрез тага

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

- Външен стилев файл, който се включва към друг стилев файл, с помощта на командата:

```
@import "nameOfCSSFile.css";
```

2. Синтаксис на CSS

- ✎ **Стиловите набори** са съвкупност от стилове.
- ✎ Описанието на всеки стил се състои от **селектор** (selector) и **декларации** (declaration).
 - **Селекторът** се използва за свързване на стила с HTML елемента, който ще бъде променен.
 - Една декларация се състои от **свойство** (property) и **стойност** (value), която е допустима за това свойство.
 - **Свойството** е името на стиловия атрибут, който трябва да се промени. Всяко свойство има стойност.

Селектор



Декларация

Декларация



Свойство

Стойност

Свойство

Стойност

2. Синтаксис на CSS

☞ Неявно свързване на стил:

- Формалното описание при неявното свързване на стилове с конкретен HTML елемент, изглежда по следния начин:

```
предефиниран_селектор{  
    свойство1: стойност;  
    свойство2: стойност; }
```

Пример:

```
img  
{  
    margin: 5px;  
    border: 1px solid;  
    border-color: blue;  
}
```

- В примера по-горе се дефинира стил на всеки таг `` за добавяна на изображение, към HTML документа. Използваните свойства са:
 - `margin: 5px;` - Определя разстояние от 5px от рамката на обекта, до останалите елементи в документа;
 - `border: 1px solid;` - Определя рамка, която е с дебелина 1px и е плътна.
 - `border-color: blue;` - Определя цвета на рамката. Стойността може да се зададе и като RGB формат.

2. Синтаксис на CSS

Явно свързване на стил:

- При явното свързване на стил с HTML елемент се използват така наречените **класове**.
- За селектор се указва не името на тага, а потребителски дефинирано наименование, което се задава като стойност на атрибута `class`, в съответния таг.
- Формалното описание на явното свързване на стилове, с конкретен HTML елемент, изглежда по следния начин:

```
.селектор_клас{  
    свойство1: стойност;  
    свойство2: стойност; }  
...  
<таг class="селектор_клас" />
```

Пример:

```
.pic  
{  
    float: left;  
    border: 1px solid;  
    border-color: blue;  
}  
...  
  

```

- В примера по-горе се дефинира стил с наименование `pic`, който ще се свърже с тага ``. Използваните свойства са:
 - `float: left;` - Определя плаваща позиция на изображението спрямо останалите обекти. В случай е ляво подравнено;
 - `border: 1px solid;` - Определя рамка, която е с дебелина 1px и е плътна.
 - `border-color: blue;` - Определя цвета на рамката. Стойността може да се зададе и като RGB формат.

2. Синтаксис на CSS

∞ Явно свързване на стил:

- Може да се прилагат няколко стила върху един HTML елемент, отново чрез използването на класове:

Пример:

```
.warning
{
    color: red; /*може и чрез шестнайсетичното представяне #FF0000*/
    font-weight: bold;
}
.big
{
    font-size: large;
}
...
<p class="warning big">Внимание!</p>
```

2. Синтаксис на CSS

∞ Явно свързване на стил:

- За явно свързване на стил с HTML елемент се използва и така наречения **именуван стил**. За определяне на именувания стил се използва знака #.
- За селектор се указва отново уникално име, което се задава като стойност на атрибута id в съответния таг.
- Характерното при именувания стил е, че в един HTML документ атрибута id, трябва да има *уникална стойност*. Не може два или повече тага да използват едно и също наименование.
- Формалното описание на именувания стил и явното му свързване с конкретен HTML елемент, изглежда по следния начин:

```
#селектор_именуван_стил{
    свойство1: стойност;
    свойство2: стойност; }
...
<таг id="селектор_именуван_стил" />
```

Пример:

```
#smallText
{
    font-size: small;
}
...
<p id="smallText">
    Текстът тук ще бъде малък!
</p>
```

2. Синтаксис на CSS

∞ Комбинирани стилове:

- CSS позволява да се създават стилове с няколко селектора, това са така наречените **комбинирани стилове**.
- Те служат за свързване към тагове, които трябва да отговарят едновременно на няколко условия.
- Правилата за създаване на комбинирани стилове са:
 1. За селектор може да се използва името на тага, името на класа или името на именувания стил.
 2. Селекторите се изброяват отляво на дясно, като по този начин се определя нивото на вложеност. Таговете, които стоят от дясно, трябва да са вложени в тези отляво.

```
p strong { color: red; }
```

```
. . .
```

```
<p> Текстът тук ще бъде <strong> червен и удебелен </strong>! </p>
```

2. Синтаксис на CSS

∞ Комбинирани стилове:

- Правилата за създаване на комбинирани стилове са:

3. Ако комбинираният стил съдържа първо името на *tag* (например <p>) и след това името на *клас / именуван стил* (например #smallText), тогава на този *tag* трябва да бъде зададен указания *клас / именувания стил*:

```
p #smallText { font-size: small; }
```

. . .

```
<p id="smallText"> Текстът тук ще бъде малък! </p>
```

4. Селекторите се разделят с интервали.
5. Стилът се свързва с тага, който е обозначен от най-десния селектор.

3. Начини на използване на стилите набори

∞ Вградени (in-line) стилове:

- Вградените стилове се описват непосредствено в HTML тага, за който се отнасят.
- Те се използват сравнително рядко, главно за тестове, тъй като не отговаря на изискванията на **Web 2.0 концепцията** (кодът на страницата да се съхранява отделно от нейното оформление).
- Тук липсва селектор, тъй като той не се използва.
- Липсват фигурни скоби.

```
<таг style="свойство1: стойност;  
        свойство2: стойност; ">
```

Пример:

```
<p style="text-align: center;  
        color: Green;  
        font: Arial">
```

Текстът е центриран, цветът е зелен, а използвания шрифт е Arial.

```
</p>
```

3. Начини на използване на стилите набори

∞ Върешни (embed) стилове:

- Върешните стилове се описват в заглавната част на страницата (<head> ... <style>...</style> </head>).
- Използват се, когато един документ има уникален стил.

Пример:

```
<head>
  <meta charset="utf-8">
  <title>My first web page</title>
  <style type="text/css">
    h1{
      font-family: Arial;
      font-size: 20pt;
      color: white;
      padding: 5px;
      text-align: center;
      width: 500px;
    }
    body {
      background: aqua;
    }
  </style>
</head>
```

3. Начини на използване на стилите набори

∞ Външни (external) стилове:

- Външният стил е идеален, когато е необходимо да се използва един и същи стил, на няколко страници.
- По този начин се осъществява лесно ревизиране на използваните стилове.
- За използването на наличния стил е необходимо към всяка страница да се добави таг с име <link>

Пример:

```
<head>  
  <meta charset="utf-8">  
  <title>My first web page</title>  
  <link rel="stylesheet" type="text/css" href="myStyle.css">  
</head>
```

4. Приоритет на стиловете

- ∞ Кой стил ще се използва за форматиране на даден HTML таг, когато има повече от няколко начина на задавена на стилове, се определя от техния приоритет.
- ∞ Приоритетите на стиловете от най-високия към най-ниския са следните:
 - Вградени (in-line) стиловете – с най-висок приоритет;
 - Вътрешни стилове;
 - Външни стилове;
 - Настройките на браузъра – с най-нисък приоритет.
- ∞ Ако връзката към *външния набор от стилове* се постави **след** *вътрешния набор от стилове* в заглавната част на документа, то външния стилове ще заменят вътрешните.

5. Специални стилове

- ✎ В HTML има още един вид селектори, които се наричат **специални селектори**.
- ✎ Характерното за специалните селектори е, че начинът на свързване на стила със съответния HTML елемент може да бъде специфициран.
- ✎ Това означава, че може да се укаже към коя част от параграф, изображение или таблица ще бъде приложено съответното форматиране.

5.1 Комбинатори

- ✎ **Комбинаторите** са специални стилове, които свързват стила със съответния HTML елемент в зависимост от неговото местоположение спрямо друг(и) HTML елементи.
- ✎ Има няколко различни вида комбинатори в CSS:
 - **Наследяващ селектор** (descendant selector): Използва се **интервала (шпацията)** за специален символ между двата селектора;
 - **Дъщерен селектор** (child selector): Между двата селектора се използва специалния символ „>“;
 - **Вложен съседен селектор** (adjacent sibling selector): Между двата селектора се използва специалния символ „+“;
 - **Общ селектор** (general sibling selector): Между двата селектора се използва специалния символ „~“;

Наследяващ селектор

- При **наследяващия селектор** се използва за специален символ само **интервала** между два селектора: **наследника** (**селектор2**) и **родителския** (**селектор1**).
- Характерното за него е че, **форматирането ще бъде приложено върху всички елементи определени от наследника** (**селектор2**), намиращи се в тага, за който се отнася родителския елемент (**селектор1**).
- Тук нивото на влагане не е от значение.

Синтаксис

```
<селектор1> <селектор2>
{
    свойство1: стойност;
    свойство2: стойност;
    свойство3: стойност;
    ...
}
```

Пример

CSS

```
div p { background-color: yellow; }
```

HTML

```
<p>Параграф без форматиране</p>
<div>
    <p>Параграф с форматиране</p>
    <h1> Заглавие 1 </h1>
    <p>Параграф с форматиране</p>
</div>
<p>Параграф без форматиране</p>
```

Дъщерен селектор

- При **дъщерния селектор** за специален символ се използва „>“.
- Характерното за него е, че **форматирането се прилага само върху непосредствения наследник (селектор2)**, намиращ се в тага, използващ **селектор1**.
- Ако в **селектор2** има друг таг, в който е вложен **селектор2**, върху него няма да бъде прилагано посоченото форматиране.

Синтаксис

```
<селектор1> > <селектор2>
{
    свойство1: стойност;
    свойство2: стойност;
    свойство3: стойност;
    ...
}
```

Пример

```
CSS
blockquote > p {
    font-style: italic;
}
HTML
<blockquote>
    <p>Параграф с форматиране</p>
    <div>
        <p>Параграф без форматиране</p>
    </div>
    <p>Параграф с форматиране</p>
</blockquote>
```

Вложен съседен селектор

- При **вложения съседен селектор** се използва специалният символ „+“, който **указва непосредствено след кой елемент трябва да се намира HTML елемента, върху който да бъде приложен съответния стил.**
- Трябва да се отбележи, че и двата елемента е необходимо да имат един и същи родителски елемент.

Синтаксис

```
<селектор1> + <селектор2>
{
    СВОЙСТВО1: СТОЙНОСТ;
    СВОЙСТВО2: СТОЙНОСТ;
    СВОЙСТВО3: СТОЙНОСТ;
    ...
}
```

Пример

```
CSS
h1 + p {
    font-family: Arial;
    font-size: 12pt;
}

HTML
<h1> Заглавие 1 </h1>
<p> Параграф в първа секция. </p>
```

Общ селектор

- При **общия селектор** се използва символът „~“, за да определи връзката между двата свързани селектора.
- Идеята тук е, че стилът ще бъде **приложен върху всички дъщерни селектори** с име **селектор2**, намиращи се след първия **селектор1**, независимо дали между тях няма други HTML елементи.

Синтаксис

```
<селектор1> ~ <селектор2>
{
    свойство1: стойност;
    свойство2: стойност;
    свойство3: стойност;
    ...
}
```

Пример

CSS

```
h2 ~ p {
    background-color: antiquewhite;
}
```

HTML

```
<p> Параграф без фонов цвят </p>
<h2>Заглавие от 2 ниво </h2>
<p> Параграф с фонов цвят. </p>
<h3>Заглавие от 3 ниво</h3>
<p> Параграф с фонов цвят. </p>
```

Селектори по атрибут на таг

- ☞ Селекторите по атрибут на таг се свързват със съответния HTML елемент, въз основа на определен атрибут или стойност на атрибут, които той притежава.
- ☞ Селекторите по атрибут не се използват самостоятелно, а се комбинират с други стилове.
- ☞ При синтаксисът на **селектор по атрибут на таг** между името на селектора и името на атрибута няма интервал, а името на атрибута се огражда в квадратни скоби.

Синтаксис

```
<селектор>[<име-на-атрибут-на-таг>]  
{  
    свойство1: стойност;  
    свойство2: стойност;  
    свойство3: стойност;  
    ...  
}
```

Пример

```
img[border]  
{  
    border: 0 none;  
}
```

6. Цветовете в CSS и HTML

- ✎ Цветовете в CSS се дефинират чрез шестнайсетична (hex или още hexadecimal) нотация, която представлява комбинация от цветовете **червено** (Red), **зелено** (Green) и **синьо** (Blue). Ето защо това представяне се нарича **RGB формат**.
- ✎ **Най-малката стойност**, която може да бъде зададена е **0** (съответно **00** в шестнайсетично представяне), а **най-голямата стойност** може да бъде е **255** (съответно **FF** в шестнайсетично представяне).
- ✎ RGB формата се състои от три двойки цифри, като всяка двойка е шестнайсетичното представяне съответно на цветовете червено – зелено – синьо, и започва със знака #.
- ✎ Комбинацията от тези цветове със стойности от 0 до 255 прави 16 милиона различни възможности ($255 * 255 * 255$).

6. Цветовете в CSS и HTML

☞ Преди години бяха определени **216 цвята**, които са съвместими и еднакво разпознаваеми от всички браузъри.

☞ Те са наречени **безопасни цветове**.

☞ Към настоящия момент съществуват константни наименования на някой от най-често използваните цветове.

☞ Примери за това са blue, green, yellow, white и т.н.

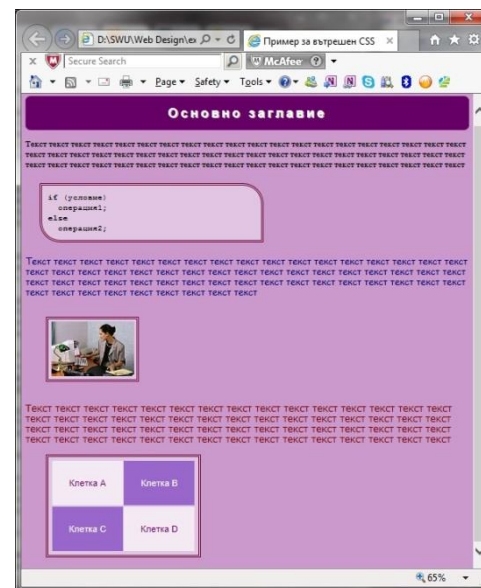
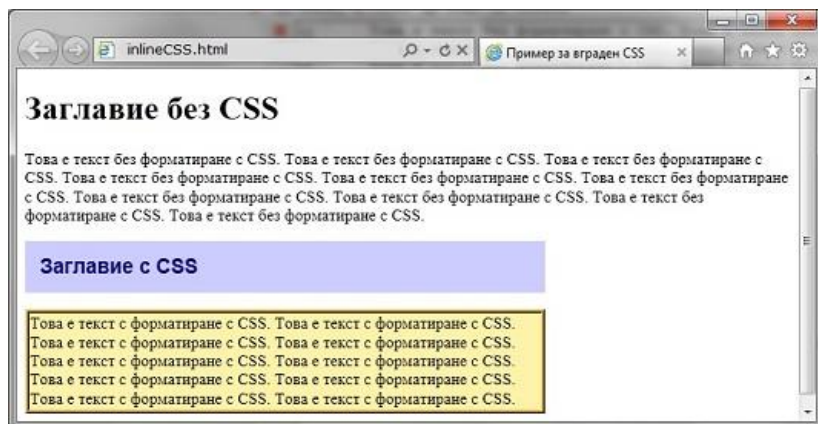
22.11.2020 г.

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

Задача

Задача:

- Изпробвайте трите различни начина на свързване на набори от стилове с HTML документ.
- Подробно описание на задачата и нейното решени, може да се види на уеб сайта на курса, в упражнение 1.



Литература

1. Jennifer Niederst Robbins and Aaron Gustafson, **Learning Web Design: A Beginner's Guide to (X)HTML, StyleSheets, and Web Graphics**, Paperback, 2007
2. Patrick McNeil, **The Web Designer's Idea Book: The Ultimate Guide To Themes, Trends & Styles In Website Design**, Paperback, 2008
3. Ethan Watrall and Jeff Siarto, **Head First Web Design**, Paperback, 2008
4. Денис Колисниченко, **HTML 5 & CSS 3 практическо програмиране за начинаещи**, изд. Асеновци, 2012
5. Жюстин Томас, **Програмиране на WEB дизайн**, изд. Нови знания, 2013
6. Алдениз Рашидов, **HTML, XHTML & CSS**, изд. Асеновци, 2012
7. Сергей Соколов, **CSS3 в примери**, изд. Асеновци, 2012

Уеб ресурси

1. W3C: HTML 5.1 –
<http://www.w3.org/html/wg/drafts/html/master/>
2. W3Schools.com - HTML 5 –
http://www.w3schools.com/html/html5_intro.asp
3. W3Schools.com - CSS 3 -
<http://www.w3schools.com/css/default.asp>
4. Сайт с учебни материали по дисциплината „Практикум по уеб дизайн“-
<http://timetable.swu.bg/lecture/rkraleva/LetenSem/WebDesign/yprWebDesign.html>
5. **Matthew J.Taylor (2009)** Four methods of adding CSS to HTML -
<http://matthewjamestaylor.com/blog/adding-css-to-html-with-link-embed-inline-and-import>

БЛАГОДАРЯ ЗА
ВНИМАНИЕТО!

