



Универзитет „Св. Кирил и Методиј“ - Скопје
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**

Аудиториски вежби 7

Вовед во CSS

Веб дизајн



6 май 2013 г.

- Подобра контрола на изгледот на буквите и распоредот на контролите низ страната
- Помалку работа
- Сите слични елементи заеднички се форматираат и може со еден стил да се дизајнира цел сајт.
- Помали документи
- Нема font тагови и вгнездени табели
- Подостапни документи
- Нема неприродна поделба на податоците заради презентација.
- Презентацискиот HTML излегува од игра
- Сите прегледувачи го поддржуваат

Основна синтакса

- Нуди единствениот начин за презентирање на изгледот на страницата
- Каскаден стил се состои од едно или повеќе правила кои опишуваат како треба да изгледа определен елемент од страницата
- Синтакса:

```
selektor { karakteristika:vrednost }
```

- Можат да се дефинираат и надвор од страницата, во рамките на .css датотека
- Верзиите на CSS се поддржани различно на старите пребарувачи
- Моментално поддржана верзија е 2.1
- Во тек е стандардизација и имплементација на CSS 3 стандардот

Видови на CSS стилови

Според местото каде се дефинирани CSS стиловите:

- Локални (Inline)
- Вгнездени (Embedded)
- Надворешни (External)

Локални CSS стилови:

```
<p style="font-size:11pt; font-face:Verdana, sans-serif; color: #ffffff;">
```

- Локалните стилови се употребуваат само во ретки специфични случаи, кога е потребно само еден елемент на само една страница треба да има различен изглед.
- Најчесто се користат за дефинирање на позиција на слој (layer).

Локални CSS стилови:

```
<h1 style="color: red">Crven naslov</h1>
<p style="font-size: 12px; font-family: 'Trebuchet MS', sans-serif">
  Ovde se primenuva prethodno definiraniot lokalен стил.
</p>

<span style="display: none;"> Ova ne se prikazuva na ekran </span>
```

- Потсеќа на употребата на презентациски тагови во HTML (пр. bgcolor, font)
- Треба да се избегнува ваквиот начин на додавање на стил, бидејќи го нарушува издвојувањето на изгледот од презентацијата

Вгнездени CSS стилови

- Вгнездените стилови се дефинирани во заглавието на HTML страницата, односно во head тагот.
- Овие каскадни стилови се претставени со <style> тагот и се достапни за користење на целата страница.
- Кога каскадните стилови важат само за една страница и се малку - може да се дефинираат како вгнездени
- Но, сепак подобро е да имаат воспоставена структура и да се чуваат оддвоено од HTML-от во посебни датотеки (./css/prva.css, ./css/vtora.css, ./css/master.css итн.)

```
<style type="text/css">
  body {background-color: blue}
  h2 {text-align: right; color: red}
</style>
```

Вгнездени CSS стилови

Пример

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <style type="text/css">
      h1 {color: #666;}
      p {font-size: 90%; font-family: Verdana, sans-serif; }
    </style>
    <title>Style Sheets</title>
  </head> ... </html>
```

Надворешни CSS стилови

- Надворешните стилови дозволуваат искористување на сите предности што ги нудат каскадните стилови.
- Ова е **препорачаниот** начин за дефинирање на CSS
- Се дефинираат во .css датотека која постои независно од страницата.
- Може да има повеќе линкувани стилови во еден документ
- Се постигнува целосна независност на изгледот на страницата од нејзината структура.
- Само со промена на правилата во .css датотеката може да се промени изгледот и чувството на целиот сајт.

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


Надворешни CSS стилови

Втор начин

```
<style type="text/css">  
    @import url(http://pathname/stylesheet.css);  
    p {font-face: Verdana;}  
</style>
```

Ова се правила со кои се одбира кои презентациски стилови на кои елементи ќе се применат. Има повеќе видови од кои поважни се:

- Селектор на тип (Type Selector)
- Класен селектор (class selector)
- Селектор според ID (ID selector)
- Универзален селектор
- Атрибутен селектор (Attribute selector)
- Псевдо-класни селектори (Pseudo-class selector)
- Комбинатори

Селектор на тип (Type Selector)

- Се применува на одреден тип на елементи од HTML документот.
- Со type selector се овозможува да се додаваат стилови на секој елемент од одреден тип.

Употреба:

```
imeNaTip { imeNaStil: vrednost; }
```

Пример:

```
span{ background-color: DodgerBlue;}  
<span>Here's a span with some text.</span>  
<p>Here's a p with some text.</p>
```

Here's a span with some text.

Here's a p with some text.

Класен селектор (class selector)

- Класниот селектор се применува на сите HTML елементи кои го имаат зададениот класен атрибут.
- Во рамките на класниот атрибут може да се задаваат повеќе класи (одвоени со празно место)

Синтакса – мора да има точка пред името на класата:

```
.imeNaKlasa { imeNaStil: vrednost; }
```

Пример:

```
span{ background-color: DodgerBlue;}  
.class1 { background-color: DodgerBlue; }  
.class2 {color: Red;}  
<span class="class1">Here is one span tag</span>  
<span class="class2">Here's another.</span>  
<span class="class2 class1">Here's third.</span>
```

Here is one span tag Here's another. Here's third.

Селектор според ID (ID selector)

- Се извршува на елементите кои го имаат зададеното ID. Се применува ако треба да се зададе стил на специфичен елемент.
- Овој тип на стилови се дефинира со додавање на # пред името на соодветното ID

Синтакса:

```
#idNaElement { imeNaStil: vrednost; }
```

Пример:

```
#identified {background-color: DodgerBlue;}  
.class2 {color: Red;}  
<span id="identified" class="class2">Some text.</span> <span>Here's another.</span>
```

Some text. Here's another.

Универзален селектор

- Претставува селектор за секој елемент во рамките на HTML документот
- Универзалниот селектор е корисен за да се постават предефинирани стилови за сите елементи во рамките на HTML документот
- Корисен за поставување на т.н. reset стил за да се израмнат разликите при различните пребарувачи

Синтакса:

```
* {style properties}
```

Пример:

```
* {  
margin: 0;  
padding: 0; }
```

Атрибутен селектор (Attribute selector)

Се користи за да се додели стил доколку е присутен атрибут или негова вредност во рамките на HTML. Употреби:

`[attr]` Репрезентира елемент со соодветен атрибут

`[attr=value]` Репрезентира елемент кај кој соодветниот атрибут ја има вредноста “value”.

`[attr ~ =value]` Репрезентира елемент кај кој соодветниот атрибут може да има повеќе вредности (разделени со празно место), и кај кој барем една од вредностите е “value”.

`[attr|=value]` Репрезентира елемент кај кој соодветниот атрибут може да има повеќе зборови (разделени со “-”), и кај кои барем еден збор е “value”.

Атрибутен селектор (Attribute selector)

Се користи за да се додели стил доколку е присутен атрибут или негова вредност во рамките на HTML. Употреби:

`[attr^=value]` Репрезентира елемент со соодветен атрибут каде што вредноста започнува со “value”.

`[attr $=value]` Репрезентира елемент со соодветен атрибут каде што вредноста завршува со “value”.

`[attr *=value]` Репрезентира елемент со соодветен атрибут каде што вредноста го содржи делот “value” како подстринг

Атрибутен селектор

Примери

```
<style type="text/css">
  /* All spans with a "lang" attribute are bold */
  span[lang] {font-weight:bold;}
  /* All spans in Portuguese are green */
  span[lang="pt"] {color:green;}
  /* All spans in US English are blue */
  span[lang~="en-us"] {color: blue;}
  /* Any span in Macedonian is red, matches simplified (mk-MK) or traditional
     (mk-MK) */
  span[lang|="mk"] {color: red;}
  /* All internal links have a gold background */
  a[href^="#"] {background-color:gold}
  /* All links to urls ending in ".mk" are red */
  a[href$=".mk"] {color: red;}
  /* All links to with "example" in the url have a grey background */
  a[href*="example"] {background-color: #CCCCCC;}
</style>
```

Атрибутен селектор

Примери

```
<div class="hello-example">
  <a href="http://example.com">English:</a>
  <span lang="en-us en-gb en-au en-nz">Hello World!</span>
</div>
<div class="hello-example">
  <a href="#portuguese">Portuguese:</a>
  <span lang="pt">Ola Mundo!</span>
</div>
<div class="hello-example">
  <a href="http://example.mk">Macedonian:</a>
  <span lang="mk-MK">Tekst na makedonski</span>
</div>
<div class="hello-example">
  <a href="http://example.cn">Macedonian:</a>
  <span lang="mk-MK">Tekst na makedonski</span>
</div>
```

English: Hello World!
Portuguese: Olá Mundo!
Macedonian: Македонски Текст
Macedonian: Македонски Текст

Псевдо-класни селектори (Pseudo-class selector)

За anchor (a) елементите постојат и следниве селектори:

- a:link** Се применува стил на некликнат (непосетен) линк
- a:visited** Се применува стил на непосетен линк
- a:hover** Се применува стил во моментот кога глумчето се наоѓа над линк
- a:active** Се применува стил во моментот кога копчето на глумчето се наоѓа над линк и е кликнато

Дополнително постојат и:

- :focus** Се применува кога некој елемент е во фокус (пр. форма која прима вредности)
- :first-child** Се применува стил на првото дете на некој елемент (пр. `li:firstchild {font-weight: bold;}`)
- :lang()** Се однесува за елементи за кои е поставен јазик (пр. `p:lang(en) {color:green;}`)

Комбинатори

Потомок селектор A B

Комбинатори претставуваат метода каде што се користат повеќе селектори за да се даде поголема специфичност.

Потомок селектор A B (Descendant Selector): Примени го стилот само ако B е потомок на A (не мора B да биде директно дете на A).

Синтакса:

```
span { background-color: white; }  
div span { background-color: DodgerBlue;}
```

Пример:

```
<div>  
<p>  
  <span>Span 1.</span><span>Span 2.</span>  
</p>  
</div>  
<span>Span 3.</span>
```

Span 1 Span 2
Span 3.

Комбинатори

Дете селектор A > B

Дете селектор A > B (Child Selector): Примени го стилот само ако B е директно дете на A.

Синтакса:

```
span { background-color: white; }  
div > span { background-color: DodgerBlue;}
```

Пример:

```
<div>  
  <span>Span 1. In the div. <span>Span 2. In the span that's in the div.</span>  
  </span>  
</div>  
<span>Span 3. Not in a div at all</span>
```

Span 1. In the div Span 2. In the span that's in the div.
Span 3. Not in a div at all.

Комбинатори

Соседно дете селектор A + B

Соседно дете селектор A + B (Child Selector): Примени го стилот само ако B е следно соседно дете на A.

Синтакса:

```
li + li {color: red;}
```

Пример:

```
<ul>  
  <li>One</li>  
  <li>Two</li>  
  <li>Three</li>  
</ul>
```

- One
- Two
- Three

Стилови кои се користат

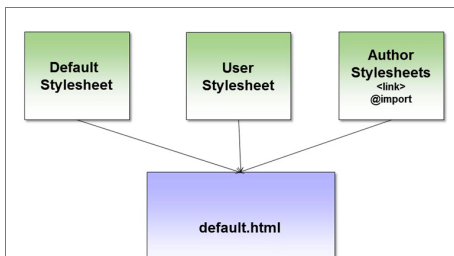
Стилови на авторот - ова се стиловите кои се креирани од дизајнерот на веб страницата

Стилови на корисникот - стилови кои се специфични за корисникот, најчесто во фајлот user.css

Стилови на корисничкиот агент (default) - ова се стиловите кои се креирани од дизајнерот на веб-страницата се предефинирани стилови од прегледувачот (наречен уште и агент) страницата

Важност на правилата:

- 1 Авторски
- 2 Кориснички
- 3 Предефинирани



- Каскадните стилови имаат својство на наследување
- Кога на два вгнездени тагови им се додава различен каскаден стил (пр. ќелија, `<td>` и параграф `<p>`)
 - Пасусот ќе ги наследи сите карактеристики дефинирани за ќелијата, а на нив ќе ги додаде сите карактеристики дефинирани за самиот пасус.
 - Ако во двата елемента постојат исти карактеристики, последниот елемент во хиерархијата, ќе го има последниот збор.
 - Се наследуваат само својствата кои се применливи (пр. својства поврзани со текст и фонтови `img` елементите не може да ги наследуваат.

Наследување

Според место на дефинирање

- Локалните стилови се најприоритетни
- Вгнездените стилови доаѓаат по нив
- Надворешните се најмалку приоритетни

Идејата е дека стиловите што поблиску се до елементите толку поголем приоритет имаат. Ова се применува за стиловите што се конфликтни. За тие својства за кои нема конфликти се прави унија (се применуваат сите).

Истото важи и за селекторите според нивиот тип. Колку типот на селекторот е поспецифичен толку е поприоритетен (пр. ID селектор е поприоритетен од Class селектор, а тој е поприоритетен од Type селектор...)

Колку приоритетни се комбинаторите?

Задавање на важност

Ако сакаме некое правило да не се надвлее од следно правило, се вклучува индикаторот `!important`:

```
p {color: blue !important;}
```

```
<p style="color: red">
```

Сепак параграфот ќе е син.

Единствен начин да се надвлее е со друг `!important` параметар зададен на поприоритетен начин (објаснето на преходниот слајд).

- Специфичност претставува начинот на кој стил ќе му го зададе на соодветниот елемент.
- Се пресметува на следниот начин:
 - Доколку елементот има style таг цифрата а ќе ја поставиме на 1
 - Бројот на ID селектори ќе го ставиме во b
 - Бројот на останати атрибути и псевдо-класи во секторот ги поставуваме во c
 - Бројот на имиња на елементи и псевдо-елементи во селекторот се поставува во d
- Специфичноста се добива со „залепување“ на цифрите во соодветниот редослед

a-b-c-d

Специфичност

Примери

```
* {} /* a=0 b=0 c=0 d=0 -> specificity = 0,0,0,0 */
li {} * a=0 b=0 c=0 d=1 -> specificity = 0,0,0,1 */
li:first-line {} /* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul li {} /* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul ol+li {} /* a=0 b=0 c=0 d=3 -> specificity = 0,0,0,3 */
h1 + *[rel=up]{} /* a=0 b=0 c=1 d=1 -> specificity = 0,0,1,1 */
ul ol li.red {} /* a=0 b=0 c=1 d=3 -> specificity = 0,0,1,3 */
li.red.level {} /* a=0 b=0 c=2 d=1 -> specificity = 0,0,2,1 */
#x34y {} /* a=0 b=1 c=0 d=0 -> specificity = 0,1,0,0 */
style="" /* a=1 b=0 c=0 d=0 -> specificity = 1,0,0,0 */
```



Прашања ?