
Уеб технологии

4 к. и 5 к., спец. БИС,
3 к., спец. Информатика,
2016-2017 уч. година



Каскадни стилови набори (Cascading Style Sheets - CSS) - използване

Cascading Style Sheets - CSS

Технология, която е предназначена за описание на оформлението на страниците

```
h1 { color: white;
      background: orange;
      border: 1px solid bl
      padding: 0 0 0 0;
      font-weight: bold;
}
/* begin: seaside-theme */

body {
  background-color:white;
  color:black;
  font-family:Arial,sans-serif,
  margin: 0 4px 0 0;
  border: 12px solid;
}
```

CSS

CSS - стилев набор

- ❖ стиливият набор съдържа съвкупност от стилове, с които се описва оформлението на страницата и отделните ѝ елементи
- ❖ всеки стил се отнася до съответен елемент от веб страницата
- ❖ връзката между елемента и стила се задава явно (указва се към кой фрагмент от страницата се прилага стилът) или неявно (стилът автоматично се прилага към всички елементи на веб страницата)

Версии

❖ CSS1 1995 г.

❖ ...

❖ текущая версия CSS3

CSS - съвкупност от правила

- ❖ правила, селектори, декларации, свойства, стойности



Формат за описание на стил

- ❖ описанието на стила се състои от селектор и списък от свойства със зададени стойности

<селектор> {

<свойство 1>: <стойност 1>;

<свойство 2>: <стойност 2>;

...

<свойство n>: <стойност n>;

}

Формат за описание на стил

Селектор - свързва стила към съответния елемент от веб страницата

↓
.blok1 {
 width: 300px;
 background: #dcc
}

СВОЙСТВО →

Задаване име на селектор

- ❖ при именуването на селектора – първият символ трябва да бъде буква (A / a - Z / z), а останалите символи могат да са цифри, тирета и букви
- ❖ без интервал в името на селектора
- ❖ може с главни / малки букви - CSS не са case-sensitive
- ❖ коментарите – в CSS те започват с /* и завършват с */

Синтактични правила

- ❖ след селектора списъкът от свойства се поставя във фигурни скоби
- ❖ вътре в списъка двойките свойство-стойност се разделят с точка и запетая
- ❖ свойството се отделя от стойността с точка и запетая
- ❖ след последната двойка свойство стойност не се поставя точка и запетая, защото някои браузъри може неправилно да интерпретират описанието на стила

CSS селектори

- ❖ Определят за кой елемент да се отнася правилото
 - ❖ към таг елемент
 - ❖ към id, class
 - ❖ по структура, напр. 2-рата картинка в 1-вия абзац
 - ❖ и др.

Видове селектори

❖ по таг

```
h1 {font-family:Arial, sans-serif;}
```

❖ по id за стилизиране на конкретна част от страница

```
#element1 {color:FF0000;}
```

❖ по клас за многократно приложение на свойства

```
.myClass {color:FF0000;}
```

Пример (по таг)

`p { color: #FF0000 }`

- ❖ неявно свързване, стилът ще се приложи към всички абзаци от страницата, защото името му съвпада с html тага `<p>`

Пример (по id)

- ❖ свързване с атрибута id; **пред името на стила се поставя # вместо .** Стойността на id трябва да е уникална в рамките на уеб страницата, защото този атрибут се използва за задаване на име на елемента на страницата и може да се използва в JavaScript

```
#bigtext  
{  
  font-size: large;  
}
```

```
<p id="bigtext">Внимание! </p>
```

Пример (по клас)

- ❖ чрез class – за стил, дефиниран от потребителя, който може да бъде прилаган многократно към всеки избран таг

.myClass

```
{  
font-size: 16px;  
font-style: italic;  
color: #FF0000;  
}
```

.warning

```
{  
color: #FF0000 ;  
font-weght:bold  
}
```

пример:

```
<p class="warning">Внимание! </p>
```

Пример

- ❖ прилагане на няколко стила към даден елемент от страницата

```
.warning { color: #FF0000 ;  
           font-weight:bold  
           }
```

```
.big { font-size: large  
       }
```

```
<p class="warning big">Внимание! </p>
```

Пример

- ❖ когато не е указана дефиниция за определен таг, но той е зададен в рамките на друг, то той наследява свойства, например тага `<i>` е в рамките на `<h1>` и наследява правилото за `h1` и се показва в червен цвят

```
h1 { color: red }
```

```
<h1>Текст 1<i>текст 2 </i>текст 3</h1>
```

Вложени селектори

- ❖ за селектиране на елемент, който се намира в друг елемент

`p.item a {font-weght: bold } -`

Parent Child

в абзаците със стил `item`

да се стилизират всички хиперлинкове

- ❖ * универсален селектор

да внимава при използването му

`p * {color: red }`

`* { background: #000000}`

Вложени селектори

- ❖ + за селектиране на елемент, който е непосредствено след друг елемент и двата елемента трябва да имат един и същ родителски елемент

<елемент 1> + <елемент2> {стил}

Родител

```
h2 + p {  
  background-color: yellow;  
}
```

наследник1

наследник2

...

<h2>Заглавие</h2>

<p>абзац1</p>

<p>абзац2</p>

Текстът на първия абзац се намира непосредствено след тага h2, неговият фон ще е жълт, а текстът на втория абзац с фона по подразбиране

Вложени селектори

- ❖ > за елемент, който е вложен в друг елемент

<елемент 1> > <елемент2> {стил}

```
blockquote > p {  
  font-style: italic  
}
```

```
<blockquote>
```

```
<p>Наклонен текст.</p>
```

```
<div>
```

```
<p>Tekst</p>
```

```
</div>
```

```
</blockquote>
```

Стилът ще се приложи само ако тагът <p> е вложен в <blockquote>, за тага абзаца в тага <div> няма да се приложи

Вложени селектори

- ❖ ~ за селектиране на елемент, който се намира зад друг елемент, но може да е отделен от първия след друг елемент и двата елемента трябва да имат един и същ родителски елемент

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

```
<body>
```

```
<div>
```

```
<p>Paragraph 1 in the div.</p>
```

```
<p>Paragraph 2 in the div.</p>
```

```
</div>
```

```
<p>Paragraph 3. Not in a div.</p>
```

```
<p>Paragraph 4. Not in a div.</p>
```

```
</body>
```

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. Not in a div.

Paragraph 4. Not in a div.

- ❖ class1.class2 - прилагат се стиловете и от двата класа
p.post-text.special {font-weight: italic }

Селектори по атрибут на таг

- ❖ [] - за избор на елементи, базирани на атрибути
- ❖ не се използват самостоятелно, а заедно със стиловите за оформяне на тага или с други стилове
- ❖ стилът може да се приложи към елементите, таговете, които имат атрибут с указаното име

<селектор>[име на атрибут на тага] {стил}

a[target] {

background-color: yellow;

}

Стилизира линковете с атрибут target

Групиране на селектори

❖ със запетая

```
html, body, div, h1, ul, a, img {  
margin: 0;  
padding: 0  
}
```

Вместо:

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

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

...

Псевдоелементи

- ❖ разновидност на специалните селектори, свързват се към определен фрагмент от уеб страницата, напр. първи символ в реда, първи ред, използват се заедно със стил
<селектор><псевдоелемент> {стил}

::first-letter - стилът се прилага върху първата буква от абзаца

::first-line - стилът се прилага върху първия ред от абзаца

Примери:

p::first-letter {font-size: larger}

P::first-line {font-size larger}

Псевдокласове

- ❖ стилът се прилага към елемента въз основа на неговото състояние, напр. връзката е посетена. Използват се заедно с основен селектор за да стане ясно върху кой обект ще се приложи стила.

`<селектор><псевдоклас> {стил}`

Примери:

`a:link {text-decoration: underline}`

`a:active:hover {
text-decoration: underline;
color: magenta
}`

комбиниране на псевдокласовете

Псевдокласове за връзки

Псевдоклас	Описание
:link	Обикновена (непосетена връзка)
:visited	Посетена връзка
:active	Активна връзка, която в момента е натисната с мишката
:focus	Връзка, имаща фокус
:hover	Връзка, над която се намира показалецът на мишката

Структурни псевдокласове

- ❖ например за прилагане на стил към елемент въз основа на местоположението му спрямо родителския му елемент

Примери:

```
li:first-child {color: red}
```

```
li:last-child {color: blue}
```

Структурни псевдокласове

- ❖ ТИЛ КЪМ ЕЛЕМЕНТ ВЪЗ ОСНОВА НА МЕСТОПОЛОЖЕНИЕТО МУ СПРЯМО РОДИТЕЛСКИЯ МУ ЕЛЕМЕНТ

Примери:

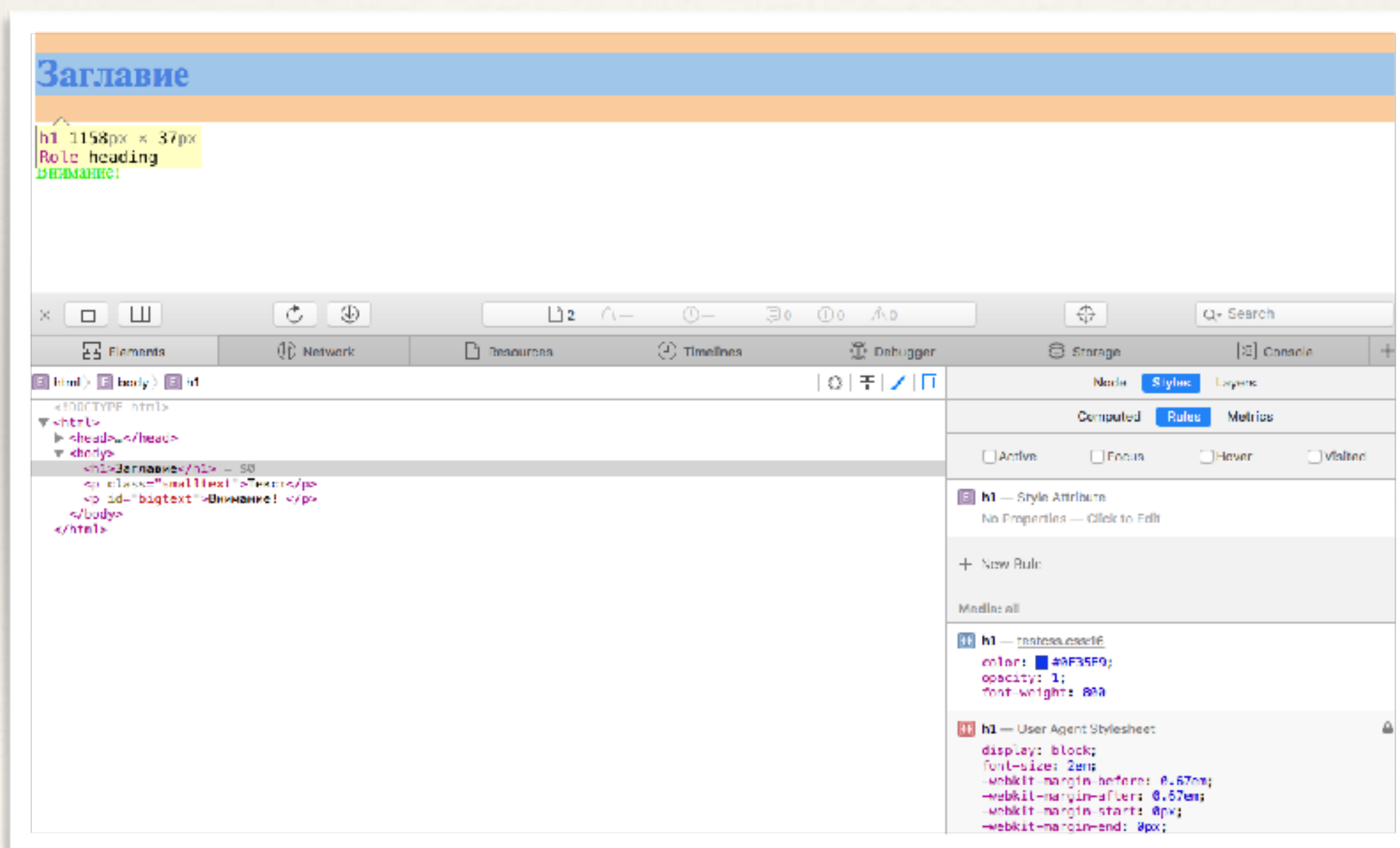
li:first-child {color: red}

li:last-child {color: blue}

- ❖ :root
- ❖ E:nth-child(n) - n-тия child на елемент
- ❖ E:nth-last-child(n) -n-тия child от зад напред
- ❖ E:nth-of-type(n) напр. n-тия елемент от тип параграф
- ❖ ...

Защо се нарича каскаден модел?

❖ правилата се наследяват



Наследяване на стилове

- ❖ наследяването на стилове се отнася само за някои правила, основно текст и лист ориентираните - color, font-size, font-family, line-height, text-align, list-style и др.
- ❖ правилата за позициониране не се наследяват - width, border, height, margin, padding, position, float и др.

Стилове за различни медии

- ❖ media screen
- ❖ media print

```
@media only screen and (max-width: 500px) {  
    .gridmenu {  
        width:100%;  
    }  
  
    .gridmain {  
        width:100%;  
    }  
  
    .gridright {  
        width:100%;  
    }  
}
```

Съхраняване на стилите набори

- ❖ инлайн стилове - в кода на html страницата
- ❖ вътрешни (вградени) стилове - в хедъра на страницата
- ❖ външни стилове - в отделен .css файл

Свързване на вътрешни стилове

```
<head>
```

```
<style type="text/css">
```

```
#content, section
```

```
{
```

```
display:block;
```

```
width:85%;
```

```
}
```

```
#right_column
```

```
{
```

```
width: 140px;
```

```
}
```

```
</style>
```

```
</head>
```

Свързване на стилове от външен файл

```
<html>
```

```
<head>
```

```
...
```

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

```
...
```

```
</head>
```

```
...
```

```
</html>
```

Свързване с няколко стилкови набора

<html>

<head>

...

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

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

...

</head>

...

</html>

Комбиниран вариант на използване на стилове

```
<head>  
<link rel="stylesheet" href="styles.css" type="text/css">  
<style type="text/css">  
    #right_column  
    {  
    width: 140px;  
    }  
</style>  
</head>
```

Приоритет на стиловете

- ❖ най-висок приоритет имат инлайн стиловете, след това са вътрешните и накрая външните
- ❖ ако един и същ стил е описан във вътрешния и във външния стилев набор, ще се насложат свойствата и при задаване на едно и също свойство ще се използва стойността на вътрешния стил или инлайн стила ако има такъв

<head>

...

<link rel="stylesheet" href="style1.css" type="text/css">

<link rel="stylesheet" href="style2.css" type="text/css">

<style>

...

</style>

</head>

Пример - приоритет на стилове

```
.redtext
```

```
{
```

```
  color: #FF0000;
```

```
  font-weight: normal
```

```
}
```

```
strong
```

```
{
```

```
  color: #00FF00;
```

```
  font-weight: bold
```

```
}
```

```
<p><strong class="redtext">текст</strong></p>
```

Текстът ще се покаже в червен цвят и с нормално изчертаване

Важно свойство - !important

❖ добавя се **!important**

```
.redtext
```

```
{  
    color: #FF0000;  
    font-weight: normal  
}
```

```
strong
```

```
{  
    color: #00FF00;  
    font-weight: bold !important  
}
```

```
<p><strong class="redtext">текст</strong></p>
```

Текстът ще се покаже в червен цвят и с удебелено изчертаване

Приоритети

- ❖ с най-нисък приоритет са браузър стиловете
- ❖ след това са стиловете зададени от автора - инлайн, в заглавната част и във външен файл
- ❖ след това са стиловете, за които авторът е задал с !important

Важно свойство - !important

❖ добавя се **!important**

```
.redtext
```

```
{
```

```
  color: #FF0000;
```

```
  font-weight: normal
```

```
}
```

```
strong
```

```
{
```

```
  color: #00FF00;
```

```
  font-weight: bold !important
```

```
}
```

```
<p><strong class="redtext">текст</strong></p>
```

Текстът ще се покаже в червен цвят и с удебелено изчертаване

Импортиране на CSS правила

- ❖ В началото на CSS файла се добавя:

```
@import url("main.css");
```

- ❖

```
<style type="text/css">  
@import url("main.css");  
</style>
```

CSS стойности

- ❖ стойностите в CSS са стрингове
- ❖ цветовете се задават в RGB формат - #<интензивност на червения цвят><интензивност на зеления цвят><интензивност на синия цвят>
- ❖ размери - pixels, points, centimeters, inches, percentage

Стойности за цветове

- ❖ чрез дума - red, blue, orange ...
- ❖ чрез шестнадесетичен код: #FFFFFF – бяло, #000000 – черно, #0099CC
- ❖ чрез кратко шестнадесетично записване - #09C
- ❖ чрез RGB формат: rgb (255, 255, 255), rgb (0, 0, 0), rgb(0, 120, 205)
- ❖ чрез Rgb като процент: rgb(100%, 100%, 100%), rgb(0%, 0%, 0%), rgb(0%, 50%, 70%)

Стойности по подразбиране на стиловете в различните браузъри

- ❖ когато няма CSS информация се използва стила по подразбиране на съответния браузър
- ❖ Внимание!!! Стиловете по подразбиране за всички браузъри са различни. Най-често това са margins, paddings и fonts. Добре е да се нулират.

Например:

```
body, h1, p, ul, ol, li  
{  
padding:0;  
margin: 0;  
}
```