



# HTML и CSS. Вёрстка сайтов

Урок 3. Основы CSS



- Что такое CSS
- Синтаксис CSS
- Способы объявления CSS
- Селекторы
  - id
  - class
  - element
  - attribute
- Основные свойства стилей
- Вложенность, наследование и группировка свойств
- Псевдоклассы и псевдоэлементы
- Приоритеты применения стилей



# Что такое CSS

---

HTML и CSS. Вёрстка сайтов  
Урок 3. Основы CSS



CSS — каскадные таблицы стилей

Вспомните, как стили работают в MS Word.

- Какую информацию включает в себя описание стиля?
- Один стиль может быть основан на другом стиле. Что это значит?



HTML – структура документа

– Элементы

CSS – оформление документа

– Как выглядят отдельные элементы  
и группы элементов

- Конечно, сам HTML тоже форматирует контент, но эта его функция от версии к версии всё более передаётся CSS

```
<body bgcolor="green">  
  <h1 align="center">  
    <font color="red">Заголовок</font>  
  </h1>  
</body>
```

- Как будет выглядеть этот документ?
- Почему не следует так форматировать документ?



# Синтаксис CSS

---

HTML и CSS. Вёрстка сайтов  
Урок 3. Основы CSS

Стиль – это набор параметров с их значениями, например:

- название шрифта: Arial  
размер шрифта: 14 пикселей  
рамка снизу: сплошная синяя толщиной 2 пикселя  
...

Это и есть описание стиля. Его только требуется формализовать – придумать стандартные названия для свойств и правила записи значений, например:

- `font-family: Arial;`  
`font-size: 14px;`  
`border-bottom: 2px solid blue;`  
...

Справочник: <http://www.w3.org/TR/CSS/> (§4.1)





```
Селектор {  
    свойство1: значение1;  
    свойство2: значение2;  
}
```

- **Селектор** выбирает элемент или группу элементов, например селектор «h2» – это все заголовки 2-го уровня
- в фигурных скобках описывается стиль, применяемый к ним, как было показано ранее

# Различные способы оформления CSS



```
Селектор { свойство1: значение1; свойство2: значение2; }
```

```
Селектор  
{  
    свойство1: значение1;  
    свойство2: значение2;  
}
```

```
Селектор {  
    свойство1: значение1;  
    свойство2: значение2;  
}
```

Селектор {

...

/\* здесь пишем комментарий \*/

...

// можно и так

...

/\* и здесь можно тоже  
его указать

\*/

...

}

- Зачем нужны комментарии?



# Способы объявления CSS

---

HTML и CSS. Вёрстка сайтов  
Урок 3. Основы CSS

## 1. Inline-стили

- описание стиля записывается в атрибут **style** конкретного элемента

## 2. стили документа

- селекторы с описаниями стилей записываются в теле элемента `html/head/style`

## 3. внешние (подключаемые) стили

- селекторы с описаниями располагаются в отдельном файле, а в `html/head` добавляется ссылка

- Например:

```
<body style="background: #0f0;" >  
    <h1 style="text-align: center; color: #f00;">  
        Заголовок  
    </h1>  
    ...  
</body>
```



```
<head>
  <style type="text/css">
    body{ background: #0f0; }
    h1{   text-align: center;
          color: #f00;}
  </style>
</head>
<body>
  <h1>Заголовок</h1>
  ...
</body>
```



style.css

```
body{ background: #0f0; }  
h1{   text-align: center;  
      color: #f00;}
```

```
<head>  
    <link rel="stylesheet"  
          href="style.css"  
          type="text/css"/>  
</head>  
<body>  
    <h1>Заголовок</h1>  
    ...  
</body>
```



# Выбор способа подключения стилей



	«+»	«-»
Inline-стили	<ul style="list-style-type: none"><li>Быстро прикрепляется к элементу без селекторов</li></ul>	<ul style="list-style-type: none"><li>Повторы, если есть одинаковые элементы</li><li>Сложно редактировать</li></ul>
Стили документа	<ul style="list-style-type: none"><li>Один стиль — несколько элементов</li></ul>	<ul style="list-style-type: none"><li>Повторы, если сайт состоит из большого количества похожих страниц</li></ul>
Внешние стили	<ul style="list-style-type: none"><li>Один стиль — весь сайт</li><li>CSS-файл кэшируется</li><li>Можно переключать стили</li></ul>	<ul style="list-style-type: none"><li>Увеличивается количество файлов (?)</li></ul>



# Селекторы (id, class, element)

---

HTML и CSS. Вёрстка сайтов  
Урок 3. Основы CSS



```
#first {  
    color: red;  
}
```

```
<p id="first">  
    цвет этого абзаца будет красным  
</p>
```

***id*** — это общий атрибут и может использоваться почти с любым элементом. ID обязан быть уникальным.

- В чём преимущество такой записи перед inline-стилем?



```
.hasborder {  
    border: 1px solid #000;  
}  
.hasbg{  
    background-image: url("")  
}
```

...

<h2 class="hasborder">Элемент h2 будет в рамке </h2>

<p class="hasborder hasbg">И абзац текста тоже в рамке, но  
еще и с фоновой картинкой</p>

...



```
.hasborder {  
    border: 1px solid #000;  
}  
p {  
    color: red;  
}
```

...

`<h2 class="hasborder">Элемент h2 будет в рамке </h2>`

`<p class="hasborder">Красный в рамке абзац</p>`

`<p>Просто красный абзац</p>`

`<p>Просто красный абзац</p>`

...



```
img[title]{           // все рисунки с title
    width: 200px;
}
```

```
input[type="text"]{   // все поля ввода
    font-size: 18px;
}
```

```

<input type="text">
```



# Основные свойства стилей

---

HTML и CSS. Вёрстка сайтов  
Урок 3. Основы CSS

## Относительные (относительно родительского элемента)

- px — пиксел
- % — процент
- em — высота шрифта
- ex — высота буквы «x»

## Абсолютные

- cm — сантиметр
- mm — миллиметр
- in — дюйм
- pt — пункт (= 1/72 in) — типографская единица (кегель шрифта измеряется в pt)
- pc — пика (= 12 pt)



## Именованные

- red, green, blue, black, white, yellow...

В числовой форме — 3 компонента (R, G и B) значениями 00÷FF или 0÷F

- #FA96CF
- #FFAA00 = #FA0

## RGB

- rgb(255, 130, 0)
- rgb(100%, 70%, 0%)

# Свойства background\* (фон)



background-color: цвет

background-image: url(*"URL рисунка"*)

background-repeat: repeat | repeat-x | repeat-y | no-repeat

background-attachment: scroll | fixed

background-position: X Y

- X — значение | left | center | right
- Y — значение | top | center | bottom

background: color image repeat attachment position

# Свойства border\* (границы)



border-width: толщина

border-style: none | hidden | dotted | dashed | solid |  
double | groove | ridge | inset | outset

border-color: цвет

border: толщина style цвет

# Свойства border\* (границы)



Аналогично определяются

border-top: толщина style цвет

border-right: толщина style цвет

border-bottom: толщина style цвет

border-left: толщина style цвет

А также

border-top-weight

border-right-style

border-bottom-color

и др. варианты

# Свойство color (цвет содержимого)

## Свойства font\* (шрифт)



color: цвет

font-style: normal | italic | oblique

font-variant: normal | small-caps

font-weight: normal | bold | bolder | lighter

font-size: число | xx-small | x-small | small |  
medium | large | x-large | xx-large

line-height: normal

font-family: {serif | sans-serif | cursive | fantasy |  
monospace | название\_шрифта}, ...

font: style variant weight size [/высота\_строки] family

# Свойства list-style\* (формат списков)



list-style-type: disc | circle | square | decimal |  
decimal-leading-zero | lower-roman | upper-roman |  
lower-greek | lower-latin | upper-latin | armenian |  
georgian | lower-alpha | upper-alpha | none

list-style-position: outside | inside

list-style-image: url("*URL рисунка*")

list-style: type position image

# Свойства text\* (формат текста)



text-align: left | right | center | justify

text-indent: размер\_отступа

text-decoration: none | underline | overline | line-through | blink

text-transform: capitalize | uppercase | lowercase | none



# Вложенность, наследование и группировка свойств

---

HTML и CSS. Вёрстка сайтов  
Урок 3. Основы CSS



# Контекстные селекторы



```
p.main strong a {  
    font-size: 28px;  
    color: red;  
}
```

`<p class="main">`В этом параграфе  
    `<strong><a href="#">`эта ссылка`</a></strong>` будет  
    размером 28 px и красного цвета,  
    `<a href="#">`а эта будет обычной`</a>`.

`</p>`

`<p>`А в этом параграфе  
    `<strong><a href="#">`ссылка`</a></strong>`  
    тоже будет обычной.

`</p>`

# Контекстные селекторы



```
p.main a {  
    font-size: 28px;  
    color: red;  
}
```

`<p class="main">`В этом параграфе  
    `<strong><a href="#">`эта ссылка`</a></strong>` будет  
    размером 28 px и красного цвета,  
    `<a href="#">`и эта тоже`</a>`.

`</p>`

`<p>`А в этом параграфе  
    `<strong><a href="#">`ссылка`</a></strong>`  
    будет обычной.

`</p>`

# Селекторы дочерних элементов



```
p.main > a {  
    font-size: 20px;  
    color: green;  
    background: yellow;  
}
```

```
<p class="main">  
    <a href="#">Ссылка 1</a>  
</p>  
<p class="main">  
    <em><a href="#">Ссылка 2</a></em>  
</p>
```

# Селекторы соседних элементов



```
p.main em + a {  
    font-size: 20px;  
    color: green;  
    background: yellow;  
}
```

```
<p class="main">  
    <a href="#">Ссылка 1</a>
```

```
</p>
```

```
<p class="main">  
    <em>Соседним элементом будет</em>&nbsp;<br>  
    <a href="#">Ссылка 2</a>  
    <a href="#">но не Ссылка 3</a>
```

```
</p>
```

```
p {  
    font-size: 18px;  
    color: red;  
}
```

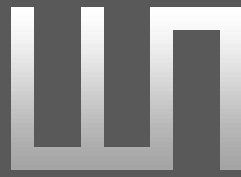
```
<p>В этом случае весь текст в этом параграфе  
    <b>будет</b> красного цвета и шрифтом  
    <b><i>размера 18 px</i><b>  
</p>
```



```
p {  
    font-size: 18px;  
    color: red;  
}
```

<p>В этом случае весь текст в этом параграфе  
 <b>будет</b> красного цвета и шрифтом  
 <b><i>размера 18 px</i><b>, только  
 <a href="#">эта ссылка</a> не будет  
</p>

# Наследование свойств стиля



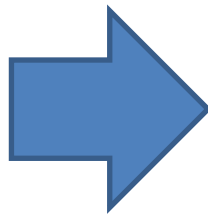
```
p {  
    font-size: 18px;  
    color: red;  
}  
p a {  
    color: inherit;  
}
```

`<p>`В этом случае весь текст в этом параграфе  
    `<b>будет</b>` красного цвета и шрифтом  
    `<b><i>размера 18 px</i><b>`, включая  
    `<a href="#">`эту ссылку`</a>`  
`</p>`

# Группировка свойств



```
h1 {  
    color: blue;  
    text-align: center;  
    font-family: Verdana;  
}  
h3 {  
    color: blue;  
    text-align: center;  
    font-family: Arial;  
}  
p {  
    color: blue;  
    text-align: center;  
    font-size: 12px;  
}
```



```
h1, h3, p {  
    color: blue;  
    text-align: center;  
}  
h1 {  
    font-family: Verdana;  
}  
h3 {  
    font-family: Arial;  
}  
p {  
    font-size: 12px;  
}
```





# Псевдоклассы и псевдоэлементы

---

HTML и CSS. Вёрстка сайтов  
Урок 3. Основы CSS

```
Селектор:псевдокласс {  
    свойство:значение;  
}
```

```
a:hover{  
    color: #ccc;  
}
```

# Псевдокласс, определяющий состояние



- :hover – курсор мыши в пределах элемента
- :active – при активации элемента
- :focus – при получении фокуса элемента
- :link – непосещенные ссылки
- :visited – посещенные ссылки

## Примеры

```
• a:link {  
    color: #0D0;  
    font-family: monospace;  
}  
a:visited {  
    color: #F00;  
    font-family: Arial;  
}
```

```
a:hover {  
    color: #FFF;  
    font-family: Verdana;  
    background: #911;  
    font-weight: bold;}  
a:active {  
    color: #001  
    font-family: "Times New Roman";  
}
```

# Псевдокласс, определяющий язык целевого элемента



```
strong:lang(en){  
    color: #f00;  
}
```

```
<p>  
    <strong>Меня зовут</strong> -  
    <strong lang="en">My name is</strong>  
</p>
```

# Псевдокласс, определяющий псевдоэлемент



Селектор элемента, являющегося  
первым ребёнком у своего родителя

```
li:first-child{  
    font-size: 24px;  
    color: #F23401;  
}
```

```
<ul>  
    <li>Первый элемент</li>  
    <li>Второй элемент</li>  
    <li>Третий элемент</li>  
</ul>
```

## Другие псевдоэлементы

- :first-letter {}
  - :first-line {}
  - :after {}
  - :before {}
- к псевдоэлементам :before и :after применимо свойство  
content: "текст" | url() | counter() | attr() | open-quote | close-quote  
(см. <http://www.w3.org/TR/CSS21/generate.html#propdef-content>)



# Приоритеты применения стилей

---

HTML и CSS. Вёрстка сайтов  
Урок 3. Основы CSS



```
body { color: blue; }  
p { color: red; }  
p { color: green; }
```

...

```
<p style="color:black;">Какого цвета?</p>
```





1. Стили автора документа
2. Стилъ, заданный пользователем  
в настройках браузера
3. Стилъ самого браузера

# Приоритеты авторских стилей. Директива !important



## 1. Директива !important

```
p {  
    color: red !important;  
}  
  
p {  
    color: green;  
}
```

- Эта директива может быть применена в пользовательском стиле

### 2. Атрибут style

```
p {  
    color: green;  
}
```

```
<h1 style="color: red;">Цвет будет красным</h1>
```

## 3. Стили документа

```
body {  
    color: blue;  
}  
h1 {  
    color: green;  
}
```

```
<body>  
    <h1>Цвет будет зелёным, а не  
    унаследованным от body</h1>  
</body>
```

## 3.1. Чем точнее селектор — тем выше приоритет

- *Селекторы элементов и псевдоэлементы* — по 1 баллу (0, 0, 0, 1)
- *Селекторы атрибутов, классы и псевдоклассы* — по 10 баллов (0, 0, 1, 0)
- *Идентификаторы* — по 100 баллов (0, 1, 0, 0)
- Атрибут `style` — 1000 баллов (1, 0, 0, 0)

# Приоритеты авторских стилей.

## Приоритеты селекторов



- `p { }` –  
1 балл (селектор элемента)
- `p::first-letter { }` –  
2 балла (1 - селектор элемента и 1 - псевдоэлемент)
- `input[type="submit"] { }` –  
11 баллов (по 1 селектору элементов и атрибутов)
- `div .head .new { }` –  
21 балл (2 класса и 1 селектор элемент)
- `#header a:hover { }` –  
111 баллов (идентификатор, элемент и псевдокласс)



- 4. Внешние стили
- 5. Наследуемые стили