

CSS

Язык описания представлений

По материалам курса University of
Washington

<http://www.cs.washington.edu/education/courses/cse190m/07sp/index.shtml>

Что такое CSS?

- это язык описания внешнего представления для содержимого, описанного в HTML-страницах;
- определяет внешний вид текста – шрифты, размеры, цвет;
- определяет расположение элементов друг относительно друга;
- описание внешнего представления может быть физически отделено от описания содержания.

В прежнем стандарте HTML допускалось использование описания внешнего представления с помощью атрибутов и отдельных элементов.

```
<p><font face="Arial">Добро пожаловать в университет ИТМО. Вы  
получите <b>самое полное, <i>лучшее, <u>ЛУЧШЕЕ</u></i></b>  
образование в России с <font size="+1" color="red">МИНИМАЛЬНЫМ  
</font>начальным багажом знаний!</font></p>
```

Добро пожаловать в университет ИТМО. Вы получите **самое полное, лучшее, ЛУЧШЕЕ** образование в России с **МИНИМАЛЬНЫМ** начальным багажом знаний!

Однако, подобный способ описания представления
НЕ ПОДДЕРЖИВАЕТСЯ в «строгом» XHTML!

Базовая структура элементов стиля

Элементы стиля описываются в виде

атрибут: значение;

и погружаются либо в атрибут `style` непосредственно в элементе HTML, либо собираются в заголовке документа (или на отдельной CSS-странице).

```
<h2 style="color: blue; text-align: center;">Курс лекций.</h2>
```

```
<head>
  <style type="text/css">
    h2 { color: blue; text-align: center; }
  </style>
  <h2 class="red-class" { color: red; text-align: center; }
</head>
```

```
<body>
  <h2>Курс лекций.</h2>
  <h2 class="red-class">Практические занятия.</h2>
</body>
```

Пример:

В ИТМО вы получите **самое полное, лучшее, качественное** образование в России с **минимальным** начальным багажом знаний!

```
<p>
  <font face="Arial">В ИТМО вы получите
    <b>самое полное,
      <i>лучшее,
        <u>качественное</u>
      </i>
    </b>образование в России с
    <font size="+1"
color="red">минимальным</font>
      начальным багажом знаний!
    </font>
  </p>
<p style="font-family: Arial;">В ИТМО вы получите
  <span style="font-weight: bold;">самое полное,
    <span style="font-style: italic;">лучшее,
      <span style="text-decoration:
underline;">качественное</span>
    </span>
  </span>образование в России с
  <span style="font-size: larger; color: red;">минимальным</span>
    начальным багажом знаний!
  </p>
```

Привязывание страницы стилей к документу

```
p { color: white; background-color: black; }  
h1 { font-size: large; font-weight: bold; }  
h2 { font-weight: 500; color: blue; }
```

mystyles.css

```
<html>  
  <head>  
    <link rel="stylesheet" type="text/css" href="mystyles.css"/>  
  </head>  
  <body>  
    содержание страницы HTML-документа  
  </body>  
</html>
```

example.html

Ссылка может быть как на «локальную» страницу стилей, созданную специально для этого документа, так и на «глобальную», хранящуюся в сети Интернет.

```
<head>  
  <link rel="stylesheet" type="text/css"  
    href="http://www.google.com/uds/css/gsearch.css" />  
</head>
```

example1.html

Некоторые атрибуты и варианты значений

Атрибуты шрифта (font) и текста (text).

```
font-family: "lucida console", "courier new", sans-serif;  
font-size: small;  
font-size: larger;  
font-size: 10px;  
font-size: 80%;  
font-weight: bold;  
font-weight: 400;  
font-style: italic;
```

```
font: sans-serif bold x-large;
```

```
text-align: center;  
text-align: right;  
text-transform: uppercase;  
text-indent: 2cm;  
text-decoration: underline;  
text-decoration: blink;
```

Атрибуты цвета

```
color: red;  
color: rgb(25, 30, 120);  
color: #c0c0c0;
```

```
background-color: yellow;
```

Допустимые словесные названия цветов:

aqua – голубой,
black – черный,
blue – синий,
fuchsia – розовый,
gray – серый,
green – зеленый,
lime – светло-зеленый,
maroon – коричневый,

navy – темно-синий,
olive – оливковый (желто-зеленый),
purple – фиолетовый,
red – красный,
silver – серебряный (светло-серый),
teal – «морской волны»,
white – белый,
yellow – желтый.

Вот как выглядят эти цвета: [colors.html](https://www.w3schools.com/colors/colors_names.asp)

Глобальное определение стиля в документе

<html>

<head>

<style type="text/css">

body { color: blue; font-family: sans-serif;
width: 400px; text-align: justify; }

strong { color: red; font-weight: 600; }

em { background-color: silver; }

</style>

</head>

<body>

<p>Италия издавна была одной из самых романтичных и посещаемых стран мира. С годами она ничуть не утратила своей притягательной силы, так как в этой солнечной стране царит атмосфера, располагающая к веселью, общению и отдыху на любой вкус. Не только красивые и разнообразные ландшафты, семейные пляжи, но и многочисленные памятники истории и культуры очаровывают путешественников, прибывающих в Италию со всех концов земли.

</p>

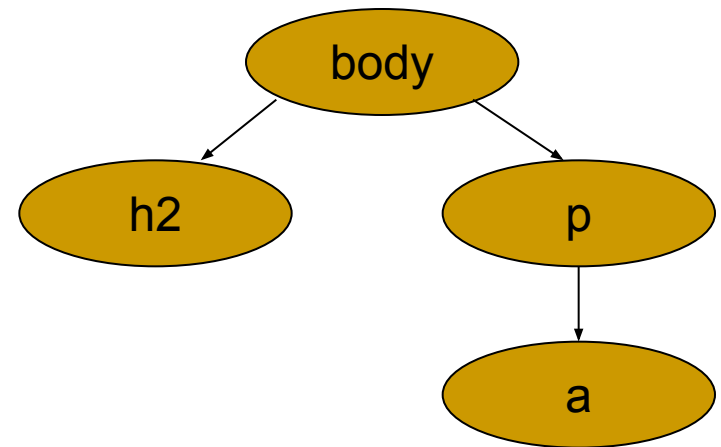
</body>

</html>

Взаимодействие стилей

```
body { font-family: sans-serif; background-color: yellow; }  
p { color: red; background-color: aqua; }  
a { text-decoration: overline underline; }  
h2 { font-weight: bold; text-align: center; }
```

```
<body>  
  <h2>Это заголовок</h2>  
  <p>А это абзац со  
    <a href="myref.html">ссылкой</a>  
    внутри  
  </p>  
</body>
```



Это заголовок _____

А это абзац со ссылкой внутри

Использование классов

```
p { font-family: sans-serif; }  
p.special { background-color: yellow; font-weight: bold; }  
.standout { color: red; font-family: cursive; }
```

```
<body>  
  <h2 class="standout">Это заголовок класса standout</h2>  
  <p>Это просто абзац</p>  
  <p class="special">Это абзац класса special</p>  
  <p class="standout special">Это абзац классов special  
    и standout</p>  
</body>
```

[classes.html](#)

Это заголовок класса *standout*

Это просто абзац

Это абзац класса *special*

Это абзац классов *special* и *standout*

Псевдо-классы

```
:link      { color: #FF0000 }      /* Еще не посещенная гиперссылка */
:visited   { color: #00FF00 }      /* Уже посещенная гиперссылка */
:hover     { color: #FF00FF }      /* Курсор мыши над ссылкой */
:active    { color: #0000FF }      /* Выделенная гиперссылка */
```

```
<body>
  <h2>Проверяем, как работают разные псевдо-классы</h2>
  <p>Ссылка на страницу
    <a href="classes.html" title="Страница с предыдущего слайда">
      classes.html</a>.</p>
</body>
```

<pseudo-classes.html>

Использование тегов div и span

```
div.style1 { font-family: sans-serif; }
div.style2 { font-family: Times; color: blue; }
.bold { font-weight: bold; }

<body>
  <div class="style1">
    <h2>Это заголовок класса style1</h2>
    <p>Это абзац класса style1</p>
  </div>
  <div class="style2">
    <h2>Это заголовок класса style2</h2>
    <p>Это абзац <span class="bold">класса</span> style2</p>
  </div>
</body>
```

Это заголовок класса style1

Это абзац класса style1

Это заголовок класса style2

Это абзац класса style2

[div.html](#)

Приписывание стилей конкретному элементу

```
#myElement { font-family: sans-serif; color: blue; }
```

```
<body>  
  <h2 id="myElement">Это мой заголовок</h2>  
</body>
```

Это мой заголовок

Каскадирование стилей

Чем определяется стиль конкретного элемента?

1. Стилем, определенным браузером «по умолчанию»
2. Стилем, указанным в отдельной CSS-странице, привязанной к HTML-документу элементом `<link>` в заголовке:
`<link rel="stylesheet" type="text/css" href="styles.css"/>`
3. Стилем, указанным в заголовке HTML-документа с помощью элемента `<style>`:
`<style> body { background-color: yellow; } </style>`
4. Стилем, указанным в самом элементе с помощью атрибута `style`:
`<p style="margin: 0.5in;">Отступ в полдюйма от края</p>`

Чем «ближе» определение стиля к элементу, тем приоритетнее он будет в случае конфликта параметров стиля.

Дополнительные возможности

```
selector1 selector2 { styles }
```

Стиль применяется к элементам, определенным селектором `selector2`, только если этот элемент находится внутри элемента, определенного селектором `selector1`.

```
selector1 > selector2 { styles }
```

Стиль применяется к элементам, определенным селектором `selector2`, только если этот элемент находится *непосредственно* внутри элемента, определенного селектором `selector1`.

Дополнительные возможности - пример

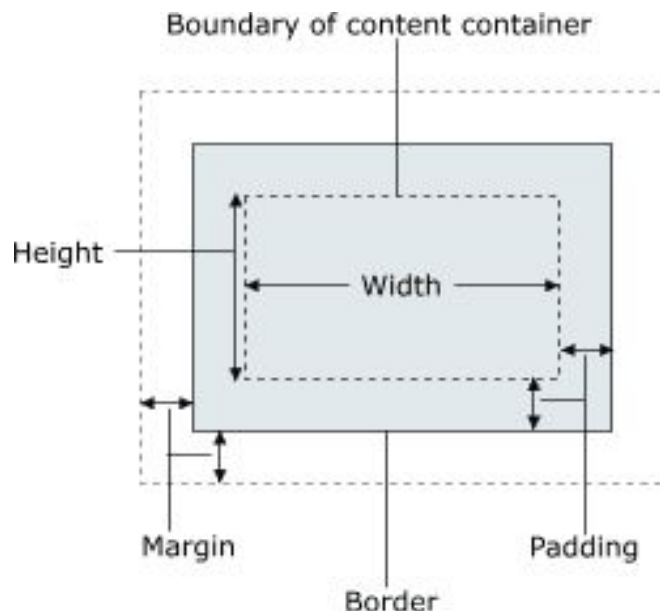
```
li strong { text-decoration: underline; }
```

```
<p>Почему люди пользуются <strong>поиском Google?</strong>  
</p><ul><li>Он <strong>очень</strong> быстрый</li>  
<li>Он дает нужные результаты</li>
```

Почему люди пользуются **поиском Google?**

- Он **очень** быстрый
- Он дает нужные результаты

Размещение фрагментов с помощью CSS



Блочные элементы имеют:

- внутреннее содержание заданной ширины и высоты (width, height);
- прослойку (padding);
- границу (border);
- поля (margin).

По умолчанию блоки располагаются вертикально, при этом поля соседних блоков перекрываются (общее поле двух соседних блоков равно по высоте максимальному из двух полей элементов).

B1

B2



Пример размещения блоков

```
<style type="text/css">
  p { font-family: sans-serif;
      font-size: 16pt;
      border: 2px solid red; }
  p.class1 { width: 400px;
             background-color: yellow;
             padding: 0.5cm;
             margin: 0.5cm; }
  p.class2 { width: 500px;
             background-color:
             green; padding: 0.3cm;
             margin: 1cm; }
</style>

<body>
  <p class="class1">Первый параграф</p>
  <p class="class1">Второй параграф</p>
  <p class="class2">Третий параграф</p>
  <p class="class2">Четвертый параграф</p>
  <p class="class1">Пятый параграф</p>
</body>
```

[blocks.html](#)

Характеристики границы (border)

```
{ border: width style color; }
```

width – ширина – 10px; 25%; 0.5in; thin; thick; medium;

style – стиль – none; solid; dotted; dashed; double;

color – цвет – red; rgb(240,240,240); #C02510;

Можно задавать отдельные характеристики границы

```
{ border-width: ...;  
  border-style: ...;  
  border-color: ...; }
```

Можно задавать характеристики границы по сторонам блока

```
{ border-top: ...;  
  border-bottom: ...;  
  border-left: ...;  
  border-right: ...; }
```

или и то и другое вместе

```
{ border-bottom-width: ...;  
  border-left-style: ...;  
  border-top-color: ...; }
```

Поля и заполнение

```
{ margin: width; padding: width; }
```

width – ширина – 10px; 5%; 0.5in;

Можно задавать ширину полей и заполнения отдельно по сторонам

```
{ margin-left: ...;  
  padding-bottom: ...;  
  padding-top: ...; }
```

Пример:

```
<h1 style="margin-top: 0.2cm; padding: 0.1cm;">Заголовок</h1>
```

Размещение блоков на странице

```
{ width: 70%; margin-left: auto; margin-right: auto; }
```

Заданный таким образом, как показано выше, стиль позволяет разместить блок по центру страницы (соответственно слева или справа, если задано только одно поле).

Размещение текста (или других строчных элементов) *внутри* блока задается иначе:

```
{ text-align: left; }    (или center или right)
```

```
<h1 style="width: 50%;  
        margin-left: auto;  
        background-color: yellow;">Заголовок</h1>
```

```
<h2 style="width: 70%;  
        margin-left: auto; margin-right: auto;  
        background-color: yellow;  
        text-align: right;">Еще один заголовок</h2>
```

Замечание: IE6 может не распознавать правильно указания `margin: auto;`

Дополнительно о размерах блоков

Вместо указания точной ширины/высоты блока можно задавать *максимальные* или *минимальные* размеры:

```
{ min-width: 100px; max-height: 5cm; }
```

В случае, если содержимое блока превышает минимальный размер – он автоматически увеличивается;

В случае, если содержимое блока превышает максимальный размер – содержимое автоматически обрезается;

Позиционирование блоков

Блоки можно позиционировать, убирая их из «автоматического» размещения по вертикали и указывая свой вариант размещения. Это можно сделать двумя способами, один из которых – указание блока как «плавающего».



Содержимое блоков «обтекает» плавающий блок таким образом, что не происходит наложения содержимого. Чтобы дополнительно визуально отделить плавающий блок, можно использовать задание в этом блоке полей (margin).

Позиционирование блоков (продолжение)

Задание «плавающего» блока:

```
img.to-right { float: right; width: 200px; }  
img.to-left  { float: left;  width: 200px; }
```

selenicereus.html

```
<body>  
    
  <p>На фотографии справа цветок совсем не ослепительно белый,  
    а кремовый. Но это не естественный его цвет, просто освещение  
    ночью искусственное, вот и дает оно желтый оттенок  
    на фотографии.</p>  
    
  <p>Фото слева: Вид на этот замечательный цветок сбоку, размер -  
    28 см (29 июня 2003 года, 2 часа ночи). Его аромат, как я  
    обнаружил, исходит из желтой зоны между коричневыми  
    чашелистиками и белыми лепестками.</p>  
</body>
```

Чтобы указать, что следующий блок не должен содержать «обтекающий» текст, можно явно указать это с помощью атрибута `clear` с возможными значениями `right`, `left`, `both`, например:

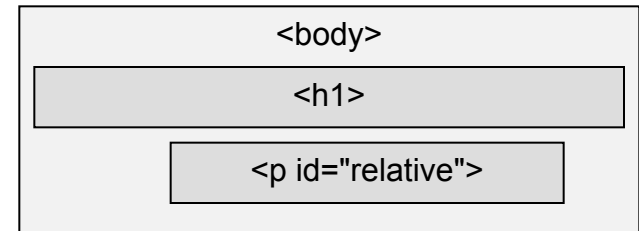
```
p.wide-paragraph { clear: both; }
```


Позиционирование блоков (продолжение)

Второй способ выведения блока из общего алгоритма размещения блоков – явное позиционирование на странице относительно начала страницы или других блоков.

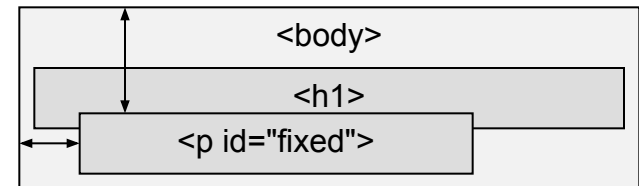
```
#relative { position: relative; right: 10%; width: 200px; }
```

Задание позиции `relative` означает, что блок смещается относительно своего «естественного» положения.



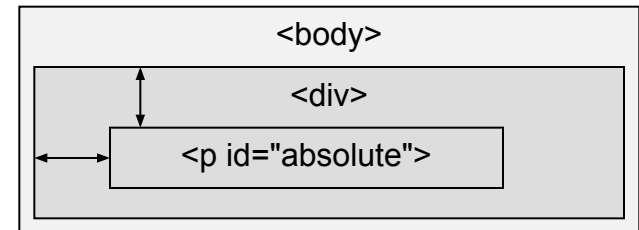
```
#fixed { position: fixed; top: 300px; left: 5em; width: 15cm; }
```

Задание позиции `fixed` означает, что блок располагается на фиксированном месте страницы.



```
#absolute { position: absolute; top: 200px; width: 200px; }
```

Задание позиции `absolute` означает, что положение блока указано относительно охватывающего его блока.



Позиционирование блоков (продолжение)

Пример.

```
<div>
  <p>Это самый обычный параграф. Таких параграфов ...</p>
</div>
<div>
  <p style="position: relative; left: 10%; width: 80%">
    Это тоже самый обычный параграф, но ...</p>
</div>
<div style="position: fixed; left: 100px; width: 80%;">
  <p style="position: absolute; left: 5em; top: 200px; width: 50%;">
    В этом параграфе указана абсолютная позиция ...</p>
</div>
<div style="position: fixed; left: 100px; width: 80%;">
  <p style="position: fixed; left: 100px; top: 300px; width: 50%;">
    Наконец, для этого параграфа указано его фиксированное ...</p>
</div>
```

[positioning.html](#)

Порядок видимости блоков

Порядок расположения блоков в случае, когда они перекрывают друг друга, определяется атрибутом стиля z-index. Чем больше значение этого атрибута, тем выше находится соответствующий объект в «стопке» перекрывающих друг друга объектов.

```
#advertising {  
    width: 350px; top: 100px; left: 50px; padding: 20px;  
    position: fixed; z-index: 100;  
    background-color: red; opacity: 0.5;  
    font-family: arial; font-size: 24pt; }
```

```
<body>  
    <h1>Это совершенно нормальный заголовок</h1>  
    <p>The CSS positioning properties...</p>  
    <p>FirePublish is the first multi-platform...</p>  
    <p>Anchor Pseudo-classes: A link...</p>  
    <div id="advertising">  
        <p style="color: white;">Эту песню не задушишь, не убьешь!</p>  
    </div>  
</body>
```

[advertising.html](#)