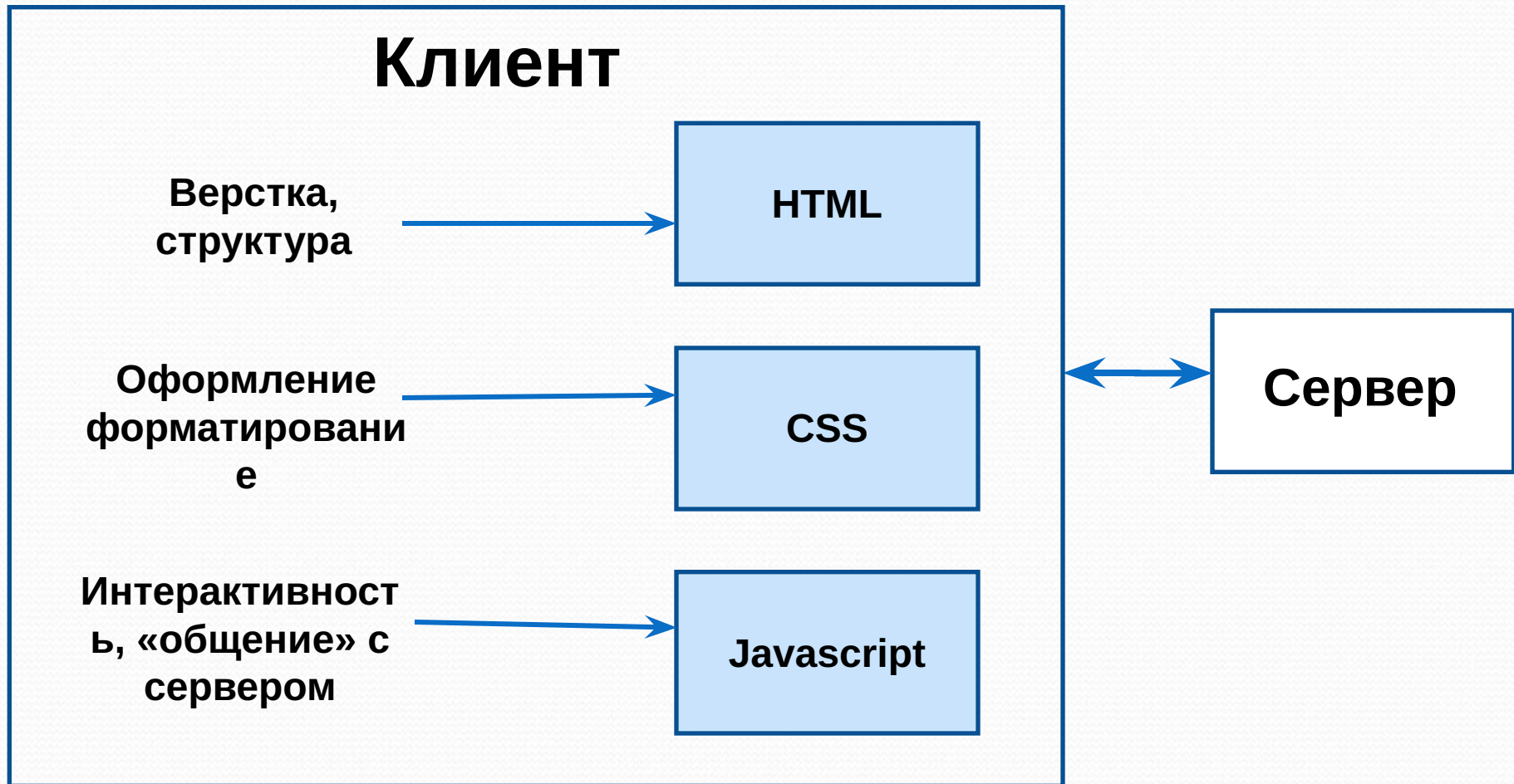


## Тема 2. Каскадные таблицы стилей CSS

---

Федорищев Леонид Александрович

# Клиентская часть



# CSS



```
body {  
  margin: 0;  
  padding: 0;  
  /*resetting default size to 10px or 62.5% of the 16px default*/  
  font-size: 62.5%;  
  font-family: Arial, Helvetica, sans-serif;  
  color: #666;  
}  
P {  
  font-size: 1.6em; /*equivalent to 16px*/  
  color: #333;  
  margin: 0 0 10px;  
  padding: 0;  
}
```

# CSS. Пример

```
<html>
<head>
<style type='text/css'>
#css
{
border-collapse:collapse;
font-family:'arial';
}
#css th
{
background-color:#DC63C2;
padding:7px;
font-size:1.1em;
border:1px black solid;
color:white;
}
#css td
{
padding:7px;
border:1px solid;
}
</style>
</head>
<body>
```

Пример оформленной с помощью CSS таблицы.

Страна	Население (млн.)
Россия	143
США	313
Китай	1347
Франция	65

Эта же таблица без оформления.

Страна	Население (млн.)
Россия	143
США	313
Китай	1347
Франция	65

# Синтаксис



# Синтаксис

```
Селектор — [ h1 {  
Свойство — [ color: red; ] — Значение  
}
```

# Пример.Шрифт

h1

{

font-family: Georgia, serif;

}

# Пример. Размер

```
h1  
{  
    font-size: 60px;  
}
```



# Селекторы

- название тега
- класс тега – class (.)
- идентификатор тега – id (#)
- атрибут тега – []
- \* все элементы страницы

# Селекторы тегов

```
p  
{  
  color:green;  
}  
h2  
{  
  color:red;  
}
```

# Селекторы тегов

```
1 <html>
2 <head>
3
4 <style type='text/css'>
5 p
6 {
7 color:green;
8 }
9 h2
10 {
11 color:red;
12 }
13 </style>
14
15
16 </head>
17 <body>
18 <p>Я первый абзац</p>
19 <h1>Это заголовок h1</h1>
20 <p>Я второй абзац</p>
21 <h2>Это заголовок h2</h2>
22 </body>
23 </html>
24
```

Я первый абзац

# Это заголовок h1

Я второй абзац

## Это заголовок h2

# Селекторы идентификаторов

```
/* Оформим элемент с id="test1" */  
#test1  
{  
  color:green;  
  font-family:verdana;  
  font-size:1.2em;  
}
```

# Селекторы идентификаторов

```
1 <html>
2 <head>
3
4 <style type='text/css'>
5 /* Оформим элемент с id='test1' */
6 #test1
7 {
8     color:green;
9     font-family:verdana;
10    font-size:1.2em;
11 }
12 </style>
13
14 </head>
15 <body>
16
17 <p id='test1'>Этот абзац будет оформлен
18
19 <p>А этот абзац изменения не затронут. <
20
21 </body>
22 </html>
23
```

Этот абзац будет оформлен с помощью CSS.

А этот абзац изменения не затронут.

# Селекторы классов

```
/* Свойства будут применены ко всем элементам с class="test1" */  
.test1  
{  
color:green;  
font-family:verdana;  
font-size:1.2em;  
}
```

# Селекторы классов

```
1 <html>
2 <head>
3 <style type='text/css'>
4 /* Свойства будут применены ко всем элементам с
5 .test1 {
6   color:green;
7   font-family:verdana;
8   font-size:1.2em;
9 }
10 </style>
11 </head>
12 <body>
13
14 <p class='test1'>Этот абзац будет оформлен с по
15
16 <p>А этот абзац изменения не затронут. </p>
17
18 <p class='test1'>И этот абзац тоже будет оформ
19
20 </body>
21 </html>
22
```

Этот абзац будет оформлен с помощью CSS.

А этот абзац изменения не затронут.

И этот абзац тоже будет оформлен с помощью CSS.

# Селекторы атрибутов

```
/* Оформит все элементы имеющие атрибут src */  
[src]  
{  
border:solid green 3px;  
}
```



# Селекторы атрибутов

```
1 <html>
2 <head>
3 <style type='text/css'>
4 /* Оформит все элементы, у которых задан атри
5 [src]
6 {
7 border-color:green;
8 border-style:solid;
9 border-width:3px;}
10 </style>
11
12 </head>
13 <body>
14
15 <img src='mountimg.jpg' alt='Ергаки 2010' />
16
17 <br /><br />
18
19 <img src='mountimg1.jpg' alt='Кузнецкий алата
20
21 <p>У данного элемента не будет зеленой рамки
22 </html>
23
```



У данного элемента не будет зеленой рамки потому, что у него нет атрибута src.

# Комбинирование селекторов

```
/* Свойства будут применены только к тем элементам с class="test1", котор  
h2.test1  
{  
color:green;  
font-family:verdana;  
font-size:1.2em;  
}
```

```
/* Свойства будут применены только к тем элементам p, которые находятся в  
div p  
{  
color:green;  
font-family:verdana;  
font-size:1.2em;  
}
```

```
/* Свойства будут применены только к тем элементам p, которые идут сразу  
div+p  
{  
color:green;  
font-family:verdana;  
font-size:1.2em;  
}
```

# Комбинирование селекторов

```
1 <html>
2 <head>
3 <style type='text/css'>
4 /* Свойства оформления будут применены только
5 h2.test1
6 {
7   color:green;
8   font-family:verdana;
9 }
10 </style>
11
12 </head>
13 <body>
14
15 <h2 class='test1'>Данный заголовок был оформл
16
17 <p class='test1'>Данный абзац не был оформлен
18
19 <h2>Данный заголовок так же не был оформлен</
20
21 </body>
22 </html>
23
```

**Данный заголовок был оформлен с помощью CSS**

Данный абзац не был оформлен

**Данный заголовок так же не был оформлен**

# Группировка селекторов

```
h1
{
font-family:verdana;
color:green;
}
h2
{
font-family:verdana;
color:green
}
p
{
font-family:verdana;
color:green;
}
```



```
h1,h2,p
{
font-family:verdana;
color:green;
}
```

# Конфликт селекторов?

## HTML:

```
<div class="hello_class" id="hello_id">Привет, мир</div>
```

## CSS:

```
div {  
    font-size: 11pt;  
    font-weight: bold;  
}  
#hello_id {  
    font-size: 27pt;  
    text-decoration: underline;  
}  
.hello_class {  
    font-size: 14pt;  
    font-style: italic;  
}
```

# Вес селекторов

- название тега имеет вес 1 балл;
- присвоенный тегу класс — 10 баллов;
- идентификатор тега — 100 баллов.

**!important** - способ явно повысить вес определенного CSS-

свойства

```
div {  
    font-size: 11pt !important;  
    font-weight: bold;  
}
```

# Каскадирование стилей.

## Вес стилей

- Стиль браузера.
- Стиль пользователя.
- Стиль автора.
- Стиль автора с добавлением !important.
- Стиль пользователя с добавлением !important.

# Псевдоклассы и псевдоэлементы

**селектор :**

**псевдокласс**

- :active
- :link
- :focus
- :hover
- :visited
- :first-child
- :lang

**селектор :**

**псевдоэлемент**

- :after
- :before
- :first-letter
- :first-line



# Текст. Цвет

## **color** **r**

- `p {color:green;}`
- `p {color:rgb(0,255,0);}`
- `p {color:#00ff00;}`

# Текст. Выравнивание

## **text-align**

### **n**

- p.ta1 {text-align:center;} (По центру)
- p.ta2 {text-align:left;} (Слева)
- p.ta3 {text-align:right;} (Справа)
- p.ta4 {text-align:justify;} (По ширине)

# Текст. Выравнивание

```
1 <html>
2 <head>
3 <style type='text/css'>
4 p.ta1
5 {text-align:center;}
6 p.ta2
7 {text-align:left;}
8 p.ta3
9 {text-align:right;}
10 p.ta4
11 {text-align:justify;}
12 </style>
13 </head>
14 <p class='ta1'>Текст данного элемента
15 <p class='ta2'>Текст данного элемента
16 <p class='ta3'>Текст данного элемента
17 <div style='width:48%;float:left;'>
18 <p class='ta4'>Текст данного элемента
19 </div>
20 <div style='width:48%;float:right;'>
21 <p class='ta4'>Текст данного элемента
22 </div>
23 </html>
24
```

Текст данного элемента выравнен по центру.

Текст данного элемента выравнен по левому краю.

Текст данного элемента выравнен по правому краю.

Текст данного элемента выравнен с помощью значения justified. Как Вы можете видеть, браузер автоматически растягивает строчки до одинаковой длины путем изменения величины отступов между словами. Попробуйте убрать выравнивание или измените ширину окна браузера, чтобы лучше понять принцип действия.

Текст данного элемента выравнен с помощью значения justified. Как Вы можете видеть, браузер автоматически растягивает строчки до одинаковой длины путем изменения величины отступов между словами. Попробуйте убрать выравнивание или измените ширину окна браузера, чтобы лучше понять принцип действия.

# Текст. Оформление

```
1 <html>
2 <head>
3 |
4 <style type='text/css'>
5 p.td1 {text-decoration:underline;}
6 p.td2 {text-decoration:line-through;}
7 p.td3 {text-decoration:overline;}
8 </style>
9
10 </head>
11 <p class='td1'>Текст данного элемента
12 <p class='td2'>Текст данного элемента
13 <p class='td3'>Текст данного элемента
14 </html>
15
```

Текст данного элемента подчеркнут.

~~Текст данного элемента перечеркнут.~~

Текст данного элемента находится под  
линией.

# Текст. Отступ между словами и буквами в тексте

С помощью CSS свойства **letter-spacing** Вы можете увеличивать или уменьшать отступ между буквами в тексте HTML элементов.

```
p.ls1 {letter-spacing:10px;}
```

С помощью свойства **word-spacing** Вы можете увеличивать или уменьшать отступ между словами в тексте HTML элементов.

```
p.ws1 {word-spacing:15px;}
```

# Текст. Отступ между словами и буквами в тексте

```
1 <html>
2 <head>
3
4 <style type='text/css'>
5 p.ls1
6 {
7     letter-spacing:10px;
8 }
9 </style>
10
11 </head>
12
13 <p class='ls1'> Величина отступа
14
15 </html>
16
```

В е л и ч и н а  
о т с т у п а м е ж д у  
б у к в а м и в  
д а н н о м э л е м е н т е  
р а в н а 1 0 п и к с .

# Текст. Другие свойства

Свойство	Описание	Значения
<code>direction</code>	Устанавливает направление текста.	<code>ltr</code> <code>rtl</code>
<code>line-height</code>	Устанавливает высоту строки.	<code>normal</code> пиксели %
<code>text-indent</code>	Устанавливает величину отступа первого символа текста.	пиксели %
<code>text-transform</code>	Устанавливает регистр букв текста элемента.	<code>none</code> <code>capitalize</code> <code>uppercase</code> <code>lowercase</code>
<code>vertical-align</code>	Устанавливает вертикальное выравнивание элемента.	<code>sub</code> <code>super</code> <code>top</code> <code>middle</code> <code>bottom</code> пиксели %
<code>white-space</code>	Устанавливает как должны обрабатываться пробелы внутри элемента.	<code>normal</code> <code>pre</code> <code>nowrap</code>

# Текст. Шрифт

**Шрифт:**

**font-family**  
`p {font-family:Arial;}`

**Размер шрифта:**

**font-size**  
`p {font-size:20px;}`



# Текст. Безопасные шрифты

Список безопасных шрифтов:

- Arial
- Arial Black
- Courier New
- Comic Sans MS
- Georgia
- Impact
- Times New Roman
- Trebuchet MS
- Verdana

# Шрифт. Размер

```
1 <html>
2 <head>
3
4 <style type='text/css'>
5   p.fz1 {font-size:20px;}
6   p.fz2 {font-size:30px;}
7   p.fz3 {font-size:13px;}
8 </style>
9 </head>
10
11 <p class='fz1'>Величина размера шрифта д
12 <p class='fz2'>Величина размера шрифта д
13 <p class='fz3'>Величина размера шрифта д
14
15 <br />
16 <b>Величина стандартного размера шрифта
17 </html>
18
```

Величина размера шрифта данного абзаца 20 пикселей.

Величина размера шрифта данного абзаца 30 пикселей.

Величина размера шрифта данного абзаца 13 пикселей.

**Величина стандартного размера шрифта 16 пикселей.**

# Шрифт. Стилль

```
1 <html>
2 <head>
3 <style type='text/css'>
4   p.italic {font-style:italic;}
5   p.bold {font-weight:bold;}
6 </style>
7 </head>
8
9 <p class='italic'>Данный абзац нап
10 <p class='bold'>Данный абзац напис
11
12 </html>
13
```

*Данный абзац написан курсивным шрифтом*

**Данный абзац написан жирным шрифтом**

# Оформление фона

## **background**

**d:**

background-attachment

background-color

background-image

background-position

background-repeat

```
body  
{ background-color:green; }
```

```
body  
{  
background-image:url('http:  
//www.wisdomweb.ru/editor  
/spider2.gif'); }
```

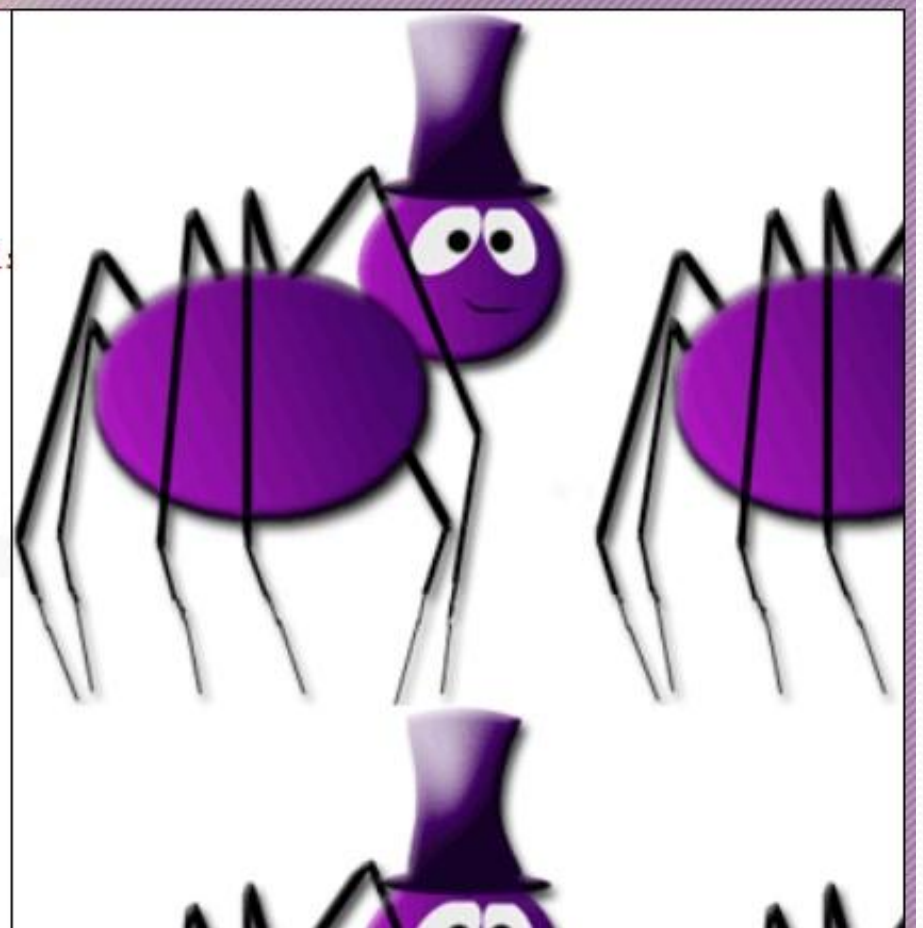
# Оформление фона: цвет

```
1 <html>
2 <head>
3 <style type='text/css'>
4 body
5 {
6   background-color:green;
7 }
8 </style>
9 </head>
10
11 <body>
12 <h2 style='color:white;'>С помощью
13 </body>
14
15 </html>
16
```

**С помощью CSS мы изменили цвет фона данной страницы на зеленый.**

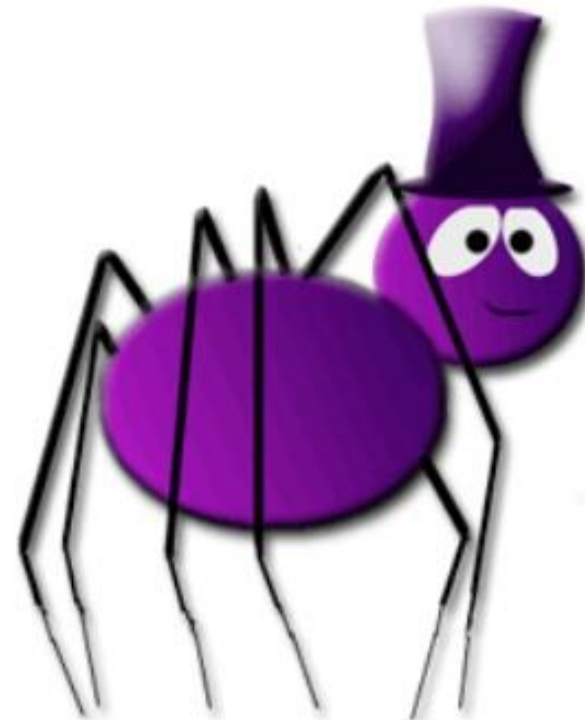
# Оформление фона: картинка

```
1 <html>
2 <head>
3 <style type='text/css'>
4 body
5 {
6   background-image:url('http://www.wi
7 }
8 </style>
9 </head>
10
11 <body>
12 </body>
13 |
14 </html>
15
```



# Оформление фона: картинка

```
1 <html>
2 <head>
3 <style type='text/css'>
4 body
5 {
6 background-image:url('http://www.
7 background-repeat:no-repeat;
8 background-position:40px 60px;
9 }
10 </style>
11 </head>
12 <body>
13 </body>
14 </html>
15
```



Свойство	Описание	Значение
<b>background</b>	Устанавливает все возможные свойства фона за одно определение	<b>background-color</b> <b>background-image</b> <b>background-repeat</b> <b>background-attachment</b> <b>background-position</b>
<b>background-attachment</b>	Указывает будет ли фоновая картинка привязана к одному месту, или будет прокручиваться вместе с текстом.	<b>scroll</b> <b>fixed</b>
<b>background-color</b>	Устанавливает фоновый цвет элемента.	<b>rgb(x,x,x)</b> <b>#xxx</b> <b>название_цвета</b>
<b>background-image</b>	Устанавливает фоновую картинку для элемента	<b>url(URL)</b> <b>none</b>
<b>background-repeat</b>	Указывает как фоновая картинка будет повторяться	<b>repeat</b> <b>repeat-x</b> <b>repeat-y</b> <b>no-repeat</b>
<b>background-position</b>	Указывает координаты расположения фоновой картинки.	<b>left top</b> <b>left center</b> <b>left bottom</b> <b>center top</b> <b>center center</b> <b>center bottom</b> <b>right top</b> <b>right center</b> <b>right bottom</b>  <b>x% y%</b> <b>xpos ypos</b>



# Оформление ссылок

**a:link** - определяет оформление обычной не посещенной ссылки.

**a:visited** - определяет оформление посещенной пользователем ссылки.

**a:hover** - определяет оформление ссылки, на которую наведен курсор мыши.

**a:active** - определяет оформление ссылки, на которую щелкнули мышкой.

# Оформление ссылок

```
1 <html>
2 <head>
3 <style type='text/css'>
4 a:link
5 {
6 text-decoration:none;
7 color:green;
8 }
9 a:visited
10 {
11 text-decoration:none;
12 color:green;
13 }
14 a:hover
15 {
16 text-decoration:underline;
17 color:red;
18 font-size:1.1em;
19 }
20 a:active
21 {
22 text-decoration:none;
23 color:red;
24 font-size:1.1em;
25 }
```

[wisdomweb.ru](http://wisdomweb.ru)

Попробуйте навести курсор мыши на ссылку выше.

# Оформление списков

```
1 <html>
2 <head>
3 <style type='text/css'>
4 ul.lis1
5 {
6 list-style-type:square;
7 }
8 ol.lis2
9 {
10 list-style-type:upper-roman;
11 }
12 </style>
13 </head>
14 <body>
15 <p> Крупнейшие города России: </
16 <ul class='lis1'>
17 <li>Москва</li>
18 <li>Санкт-Петербург</li>
19 <li>Новосибирск </li>
20 </ul>
21 <i> Маркер списка имеет вид квад
22 <p> Города России отсортированнь
23 <ol class='lis2'>
24 <li>Москва</li>
25 <li>Санкт-Петербург</li>
```

Крупнейшие города России:

- Москва
- Санкт-Петербург
- Новосибирск

*Маркер списка имеет вид квадрата (по умолчанию имеет вид круга).*

Города России отсортированные по численности населения:

- I. Москва
- II. Санкт-Петербург
- III. Новосибирск

*Элементы списка нумеруются римскими цифрами (по умолчанию арабскими).*

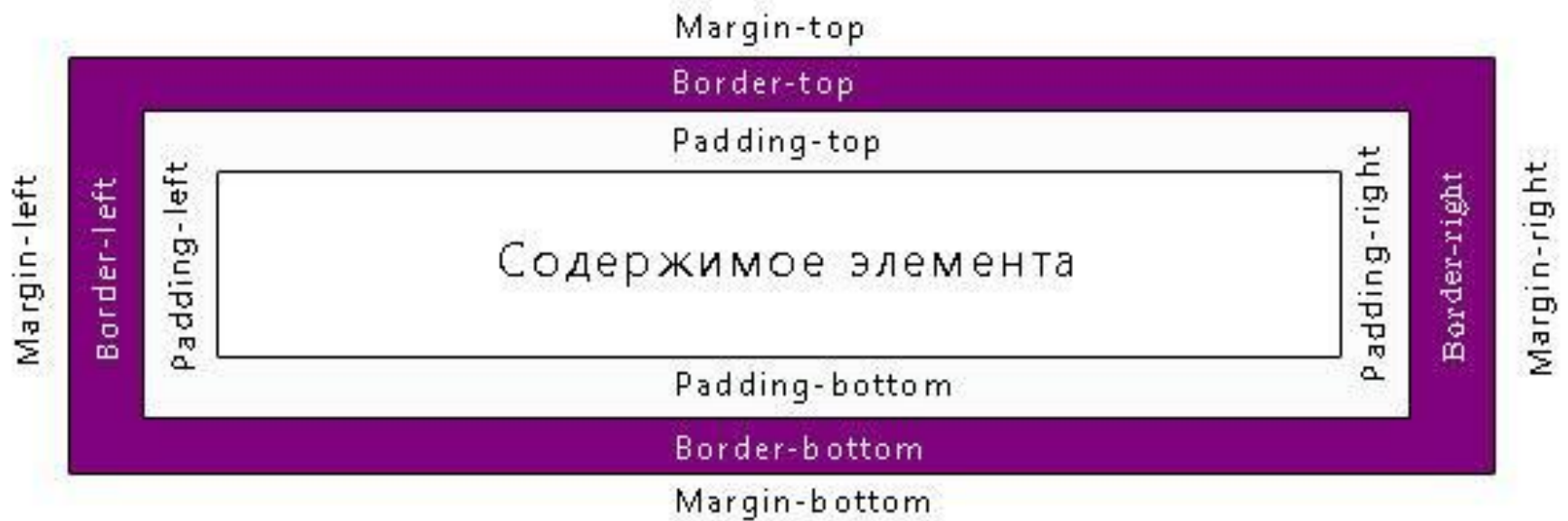
# Оформление таблиц

```
1 <html>
2 <head>
3 <style type='text/css'>
4 table, th, td
5 {
6 border-style:solid;
7 border-width:1px;
8 border-collapse:collapse;
9 padding:2px;
10 }
11 th
12 {
13 height:28px;
14 background-color:#f892dc;
15 color:white;
16 border-color:black;
17 }
18 .ts1 {background-color:#ffeffb;}
19 </style>
20 </head>
21 <body>
22 <h4> Самые высокие небоскребы по контин
23 <table>
24 <tr>
25 <th> Континент </th>
```

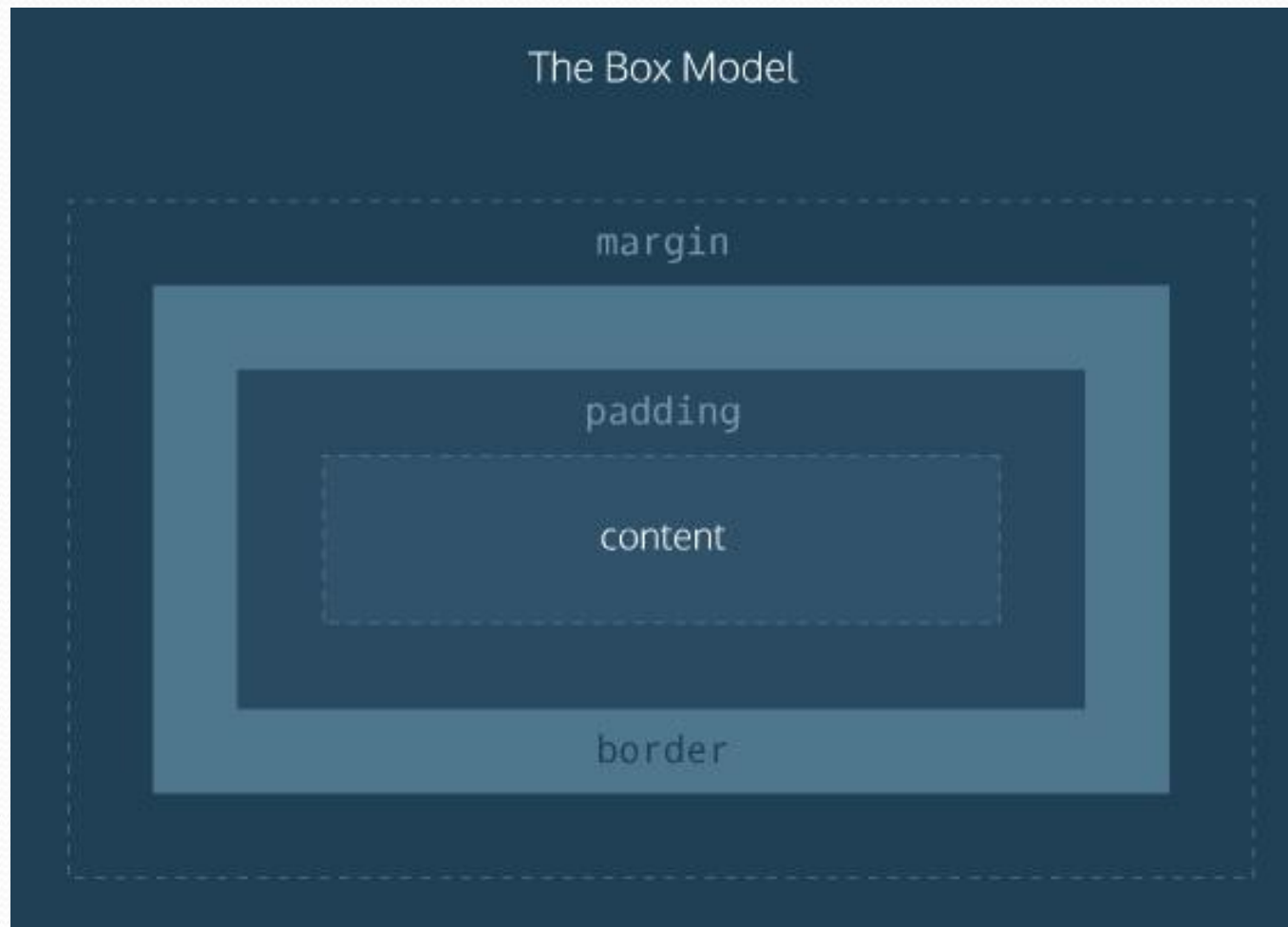
## Самые высокие небоскребы по континентам:

Континент	Название	Город	Высота, м
Австралия и Океания	Q1	Голд-Кост	323
Азия	Дубайская башня	Дубай	828
Африка	Карлтон Сентр Офис Тауэр	Йоханнесбург	223
Европа	Башня 'Москва'	Москва	302
Северная Америка	Сирс Тауэр	Чикаго	443
Южная Америка	Парке Централь	Каракас	225

# Блочная модель

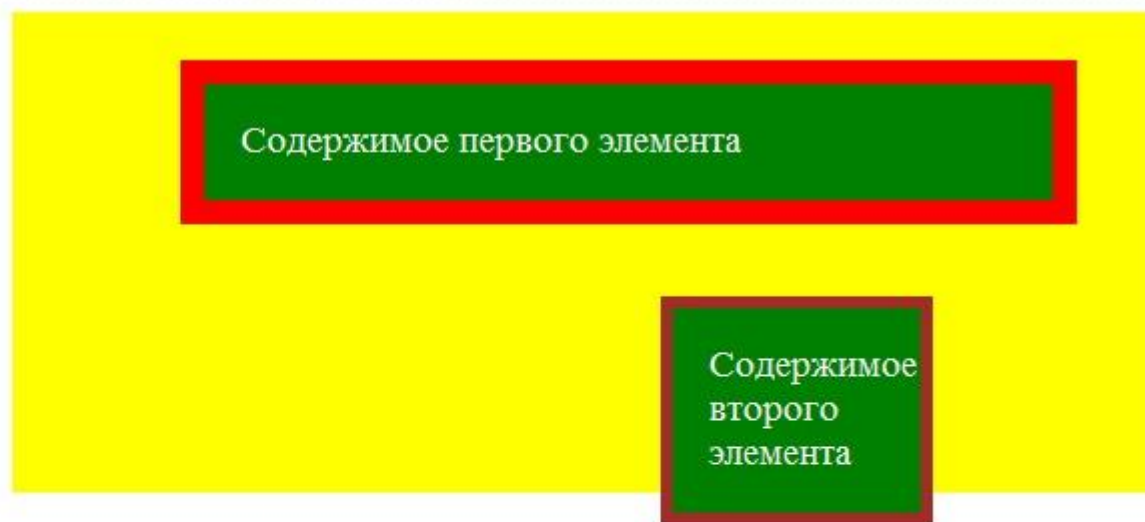


# Блочная модель



# Блоковая модель

```
1 <html>
2 <head>
3 <style type='text/css'>
4 #wrap {
5 margin:0px;
6 padding:20px;
7 height:160px;
8 background-color:yellow;
9 }
10 .ex1 {
11 border:10px red solid;
12 margin-left:50px;
13 margin-right:10px;
14 padding:15px;
15 background-color:green;
16 color:white;
17 }
18 .ex2 {
19 border:5px brown solid;
20 margin-top:30px;
21 margin-left:250px;
22 margin-right:70px;
23 padding:15px;
24 background-color:green;
25 color:white;
26 }
```



# Стиль границ

```
1 <html>
2 <head>
3 <style type='text/css'>
4 .bor1
5 {
6   border-style:solid;
7   border-width:4px;
8   border-style:dashed;
9 }
10 .bor2
11 {
12   border-style:solid;
13   border-width:2px;
14   border-color: green;
15 }
16 .bor3
17 {
18   border-style:solid;
19   border-width:thin;
20 }
21 </style>
22 </head>
23 <body>
24 <p class='bor1'>Данный элемент имеет г
25 <p class='bor2'>Данный элемент имеет г
```

Данный элемент имеет границу толщиной 4 пикселя.

Данный элемент имеет границу толщиной 2 пикселя.

Толщина границы установлена с помощью предопределенного значения thin.



# Стиль границы в одной строке

```
1 <html>
2 <head>
3 <style type='text/css'>
4 /* Вокруг элемента с классом bor1 будет
5 толщиной 2 пикселя */
6 .bor1
7 {
8     border:2px solid green;
9 }
10 </style>
11 </head>
12 <body>
13 <p class='bor1'>Данный элемент был оформ
14 </body>
15 </html>
16
```

Данный элемент был оформлен с помощью CSS.

# Padding: Внутренний отступ

```
1 <html>
2 <head>
3 <style type='text/css'>
4 .pad1
5 {
6   border-style:solid;
7   padding:20px;
8 }
9 .pad2
10 {
11   border-style:solid;
12   padding:10px;
13 }
14 .pad3
15 {
16   border-style:solid;
17   padding:5px;
18 }
19 </style>
20 </head>
21 <body>
22 <p class='pad1'>Элемент с внутренним о
23 <p class='pad2'>Элемент с внутренним о
24 <p class='pad3'>Элемент с внутренним о
25 </body>
```

Элемент с внутренним отступом равным 20 пикселям.

Элемент с внутренним отступом равным 10 пикселям.

Элемент с внутренним отступом равным 5 пикселям.

# Padding: Внутренний отступ

```
1 <html>
2 <head>
3 <style type='text/css'>
4 /* Отступ от содержимого элемента до его
5 .pad1
6 {
7   border-style:solid;
8   padding-top:30px;
9   padding-left:20px;
10  padding-bottom:10px;
11  padding-right:40px;
12 }
13 </style>
14 </head>
15 <body>
16 <p class='pad1'>Элемент с заданным внутр
17 </body>
18 </html>
19
```

Элемент с заданным внутренним отступом.

# Padding: краткая форма записи

```
1 <html>
2 <head>
3 <style type='text/css'>
4 /* Отступ от содержимого до границы эле
5 .pad1
6 {
7 border:2px solid;
8 padding:60px 20px 40px 50px;
9 }
10 /* Отступ от содержимого до границы эле
11 .pad2
12 {
13 border:2px solid;
14 padding:40px 30px 10px;
15 }
16 /* Отступ от содержимого до границы эле
17 .pad3
18 {
19 border:2px solid;
20 padding:40px 30px;
21 }
22 </style>
23 </head>
24 <body>
25 <p class='pad1'>Внутренний отступ свер:
```

Внутренний отступ сверху равен 60, справа 20, снизу 40, а слева 50 пикс.

Внутренний отступ сверху равен 40, слева и справа 30, а снизу 10 пикс.

Внутренний отступ сверху и снизу равен 40, а слева и справа 30 пикс.

# Margin: ВНЕШНИЙ ОТСТУП

```
1 <html>
2 <head>
3 <style type='text/css'>
4 .dvl
5 {
6   border-style:solid;
7   border-width:1px;
8   background-color:red;
9 }
10 .mar1, .mar2, .mar3
11 {
12   border-width:1px;
13   border-style:solid;
14   background-color:white;
15 }
16 .mar1
17 {
18   margin:25px;
19 }
20 .mar2
21 {
22   margin:10px;
23 }
24 .mar3
25 {
```

Элемент с внешним отступом 25 пикселей.

Элемент с внешним отступом 10 пикселей.

Элемент с внешним отступом 5 пикселей.

**Обратите внимание:** внешний отступ в данном примере для наглядности выделен красным цветом.

# Скрытие элементов

В CSS скрыть элементы можно двумя способами:

1. С помощью свойства **visibility:hidden**;
2. С помощью **display:none**.

Элемент скрытый первым способом будет невидим, но будет по прежнему занимать место на странице.

## Пример

```
/* Абзац стал невидим, но все еще занимает место на странице */  
p.dis1  
{  
  visibility:hidden;  
}
```

Второй способ позволяет полностью скрыть элемент, чтобы он больше не занимал места на странице.

## Пример

```
/* Абзац стал невидим и не занимает места */  
p.dis1  
{  
  display:none;  
}
```

# Скрытие элементов

```
1 <html>
2 <head>
3 <style type='text/css'>
4 #dis1
5 {
6   visibility:hidden;
7 }
8 </style>
9 </head>
10 <body>
11 <p id='dis1'>Данный абзац невидим, но за
12 <p>Это второй абзац.</p>
13 <b>Первый абзац не видим, но занимает ме
14 </body>
15 </html>
16
```

Это второй абзац.

**Первый абзац не видим, но занимает место.**

# Блочные и строковые элементы

В CSS встречаются два типа элементов по способу отображения:

- **Блочные элементы** полностью занимают всю ширину их родительского элемента. Примеры блочных элементов: p, h1-h6, div.
- **Строковые элементы** занимают только необходимую им ширину. Примеры строковых элементов: a, span.

С помощью CSS Вы можете изменять способ отображения элементов. В следующем примере элемент p отображен как строковый:

## Пример

```
/* Превратим блочный элемент p в строковый */  
#dis1  
{  
  display:inline;  
}
```

Также Вы можете отображать строковые элементы как блочные:

## Пример

```
/* Превратим строковый элемент a в блочный */  
a.dis1  
{  
  display:block;  
}
```



# Блочные и строковые элементы

## Block vs Inline

`<h1>`

`<p>`

`<ul>`

`display: block;`

`<img>`

`<img>`

`<img>`

`<a>`

`display: inline;`

# Размещение элементов

```
#pos1  
{  
    position:absolute;  
    top:10px; left:200px;  
}
```

# Размещение элементов

```
4 #pos1 {
5   position:absolute;
6   top:10px;
7   left:200px;
8   background-color:green;
9   padding:2px;
10  width:300px;
11  border:1px solid;
12 }
13 #pos2 {
14   position:absolute;
15   top:1px;
16   left:0px;
17   background-color:pink;
18   padding:2px;
19   width:400px;
20   border:1px solid;
21 }
22 #pos3 {
23   position:absolute;
24   top:100px;
25   right:70px;
26   background-color:red;
27   padding:2px;
28   height:90px;
```

Еще один абзац размещенный абсолютно.

Еще один абзац  
размещенный  
абсолютно.

# Относительное размещение

```
1 <html>
2 <head>
3 <style type='text/css'>
4 #pos1
5 {
6   position:relative;
7   top:0px;
8   left:300px;
9   border-style:solid;
10  width:300px;
11 }
12 </style>
13 </head>
14 <body>
15 <p id='pos1'>Данный абзац размеще
16 <p id='pos2'>Это обычный абзац.</
17 <br />
18 <p><b>Обратите внимание:</b> абза
19 Это значит, что относительно разм
20 </body>
21 </html>
22
```

Данный абзац размещен

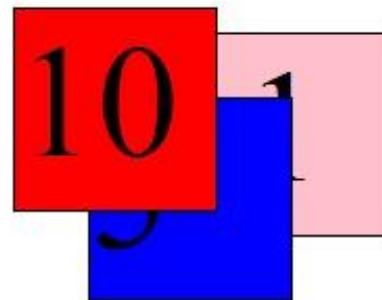
Это обычный абзац.

**Обратите внимание:** абзац pos2 отображен в том же месте, где он был бы отображен, если бы элемент pos1 был размещен статично. Это значит, что относительно размещенные элементы (в отличие от абсолютно и фиксировано размещенных) занимают место, на котором они изначально были определены.

# Наложение элементов

```
3 <style type='text/css'>
4 #pos1
5 {
6   position:absolute;
7   top:0px;
8   left:100px;
9   border:1px solid;
10  width:80px;
11  height:80px;
12  background-color:red;
13  font-size:4em;
14  z-index:10;
15 }
16 #pos2
17 {
18   position:absolute;
19   top:36px;
20   left:130px;
21   border:1px solid;
22   width:80px;
23   height:80px;
24   background-color:blue;
25   font-size:4em;
26   z-index:5;
27 }
```

Элемент с *z-index:10* - красный, с *z-index:5* - синий, а с *z-index:-1* - розовый.



# Выравнивание с margin

```
1 <html>
2 <head>
3 <style type='text/css'>
4 .ali1
5 {
6   margin-left:auto;
7   margin-right:auto;
8   width:50%;
9 }
10 </style>
11 </head>
12 <body>
13 <p class='ali1'>Данный абзац выра
14 </body>
15 </html>
16
```

Данный абзац  
выравнен по центру.

# Выравнивание с float

Элементы в CSS также могут быть выравнены с помощью свойства **float**.

Элемент выравненный с помощью float будет прижат к левой или правой границе родительского элемента (в зависимости от заданного значения) и заставит следующие за ним элементы "обтекать" его с противоположной стороны.

Свойство float часто используют с картинками (как в примере ниже), но оно также бывает полезно при обычном выравнивании.

## Пример

```
.f11
{
float:right;
}
```

# Float: обтекание элемента

```
1 <html>
2 <head>
3 <style type='text/css'>
4 .fl1
5 {
6   float:right;
7   margin:0px;
8   margin-left:5px;
9   width:340px;
10  height:260px;
11 }
12 </style>
13 </head>
14 <body>
15 <img src='mountimg3.jpg' class='fl1'
16 <p><b>Ергаки (Пальцы)</b> — горный уз
17 <p>Ергаки находится на территории Кра
18 <p>Ергаки — название природного парка
19 </body>
20 </html>
21
```

**Ергаки (Пальцы)** — горный узел, хребет в Западном Саяне. Расположен в истоках рек Большой Кебеж, Большой Ключ, Тайгиш, Верхняя Буйба, Средняя Буйба и Нижняя Буйба.



Ергаки находится на территории Красноярского края, в пределах Ермаковского и Каратузского районов. Через Ергаки (в восточной части) проходит федеральная трасса М-54. Ближайшие города (по трассе) — Искитим (200 км на север), Абакан (200 км на юг).



# Очищение от float

HTML элементы следующие за элементом с заданным float будут его "обтекать", чтобы избежать этого используйте свойство **clear**.

Значения переданные данному свойству (left или right) указывают с какой стороны находится элемент с float. Если Вы не уверены или элементы с float находятся с двух сторон воспользуйтесь свойством both.

## Пример

```
.f12  
{  
clear:both;  
}
```

# Способы подключения к HTML

- Внутренние стили (внутри тегов, атрибут style)
- Глобальные стили (указанные внутри тега style)
- Связанные стили (отдельный файл CSS, подключение в html через link)
- Импорт стилей (еще один вариант подключения внешних CSS-файлов)

# Строковое подключение СТИЛЯ

```
<p style="font-size:1.3em">
```

Абзац оформленный с помощью CSS.

```
</p>
```

# Внутренние стили

```
<head>  
  <style type='text/css'>  
    h1 {color:red;}  
    p {margin-right:38px;}  
    div {float:left;}  
  </style>  
</head>
```

# Подключение внешнего файла

```
<head>
```

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

```
  адрес_внешнего_файла_стилей" />
```

```
</head>
```