Современные веб-технологии

CSS3

Основы CSS3

Определение стиля состоит из двух частей:

селектор, который указывает на элемент,

блок объявления стиля - набор команд, которые устанавливают правила форматирования.

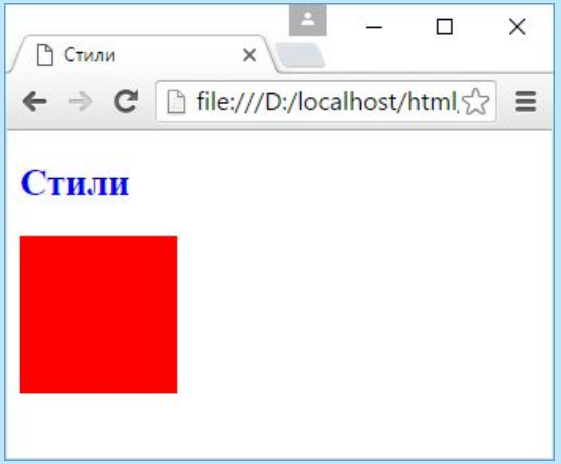
```
div{
    background-color:red;
    width: 100px;
    height: 60px;
}
```

Основы CSS3

Существуют различные способы определения стилей.

Атрибут **style**

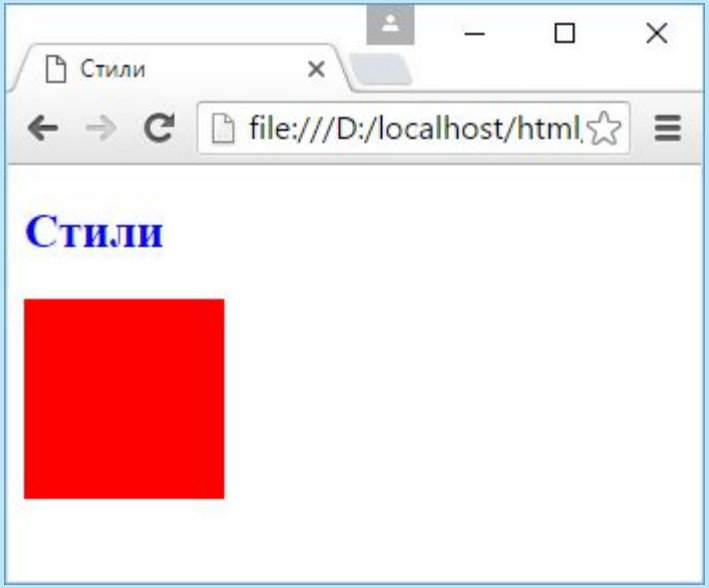
Первый способ заключается во встраивании стилей непосредственно в элемент с помощью атрибута style:




```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Стили</title>
    </head>
    <body>
        <h2 style="color:blue;">Стили</h2>
        <div style="width: 100px; height: 100px; background-color: red;"></div>
    </body>
</html>
```

Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Стили</title>
      <style>
      h2{
         color:blue;
      }
      div{
         width: 100px;
         height: 100px;
         background-color: red;
      }
      </style>
   </head>
   <body>
      <h2>Стили</h2>
      <div></div>
   </body>
</html>
```



Основы CSS3

файл styles.css

```
h2{
    color:blue;
}
div{
    width: 100px;
    height: 100px;
    background-color: red;
}
```

Файл HTML

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Стили</title>
        <link rel="stylesheet" type="text/css" href="styles.css"/>
    </head>
    <body>
        <h2>Стили</h2>
        <div></div>
    </body>
</html>
```

Основы CSS3

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" type="text/css" href="styles.css"/>
        <style>
            div{
                width:200px;
            }
        </style>
    </head>
    <body>
        <div style="width:120px;"></div>
    </body>
</html>
```

А в файле style.css определен следующий стиль:

```
div{
    width:50px;
    height:50px;
    background-color:red;
}
```

Система приоритетов

Если у элемента определены встроенные стили (inline-стили), то они имеют высший приоритет, то есть в примере выше итоговой шириной будет 120 пикселей.

Далее в порядке приоритета идут стили, которые определены в элементе style.

Наименее приоритетными стилями являются те, которые определены во внешнем файле.

Основы CSS3

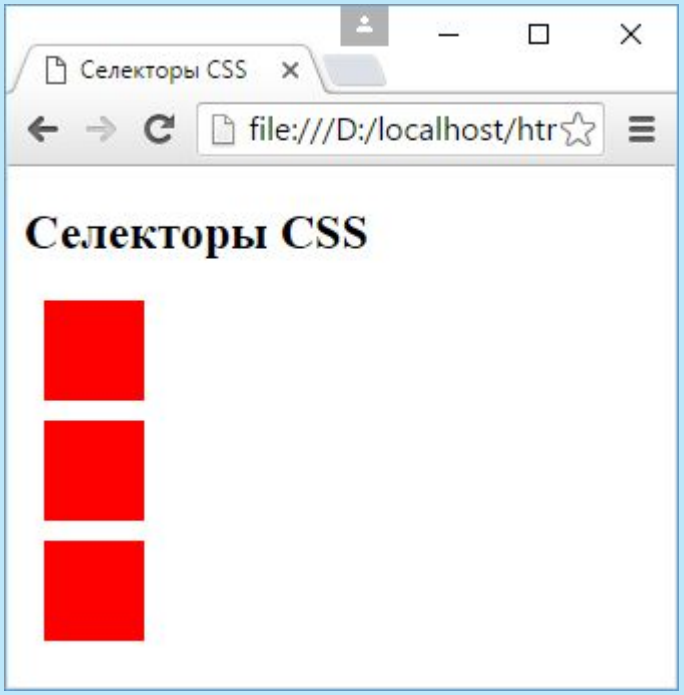

Семантическая разметк
The W3C CSS Validation
Not secure | jigsaw.w3.org/css-validator/
Deutsch English Español Français 한국어 Italiano Nederlands 日本語 Polski Português Русский فارسی Svenska Български Українська Čeština Romanian Magyar Ελληνικά हिन्दी 简体中文
W3C
CSS Validation Service
Check Cascading Style Sheets (CSS) and (X)HTML documents with style sheets
By URI
By file upload
By direct input
Validate by URI
Enter the URI of a document (HTML with CSS or CSS only) you would like validated:
Address:
More Options
Check
mozilla
The W3C validators are developed with assistance from the Mozilla Foundation, and supported by community donations.
Flattr us!
Donate and help us build better tools for a better web.
our CSS style sheet embedded in an (X)HTML document, you
Connecting...

Основы CSS3

Определение стиля начинается с селектора.

```
div{
   width:50px; /* ширина */
   height:50px; /* высота */
   background-color:red; /* цвет фона */
   margin: 10px; /* отступ от других элементов */
}
```

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы CSS</title>
      <style>
         div{
            width:50px;
            height:50px;
            background-color:red;
            margin: 10px;
         }
      </style>
   </head>
   <body>
      <h2>Селекторы CSS</h2>
      <div></div>
      <div></div>
      <div></div>
   </body>
</html>
```



Основы CSS3

Классы

Иногда для одних и тех же элементов требуется различная стилизация. И в этом случае можно использовать классы.

Для определения селектора класса в CSS перед названием класса ставится точка:

```
.redBlock{
    background-color:red;
}
```

Название класса может быть произвольным. Например, в данном случае название класса - "redBlock". Однако при этом в имени класса разрешается использовать буквы, числа, дефисы и знаки подчеркивания, причем начинать название класса должно обязательно с буквы.

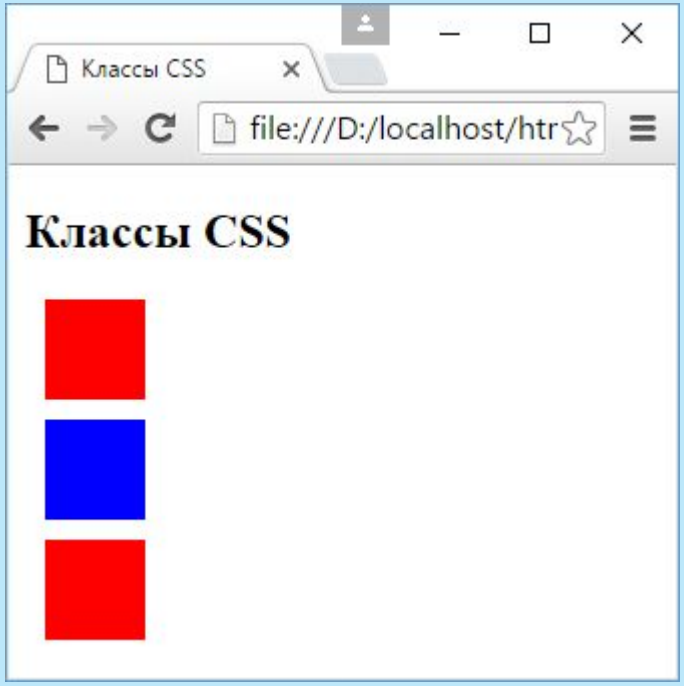
Также стоит учитывать регистр имен: названия "article" и "ARTICLE" будут представлять разные классы.

После определения класса мы можем его применить к элементу с помощью атрибута class. Например:

```
<div class="redBlock"></div>
```

Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Классы CSS</title>
      <style>
         div{
            width: 50px;
            height: 50px;
            margin: 10px;
         }
         .redBlock{
            background-color: red;
         }
         .blueBlock{
            background-color: blue;
         }
      </style>
   </head>
   <body>
      <h2>Классы CSS</h2>
      <div class="redBlock"></div>
      <div class="blueBlock"></div>
      <div class="redBlock"></div>
   </body>
</html>
```

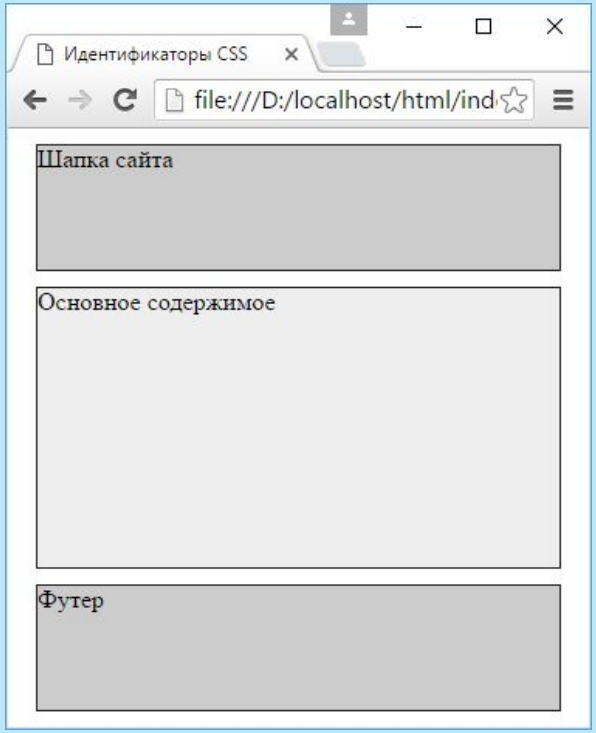


Основы CSS3

Идентификаторы

Для идентификации уникальных на веб-станице элементов используются идентификаторы, которые определяются с помощью атрибутов id. Определение стилей для идентификаторов аналогично определению классов, только вместо точки ставится символ решетки #:

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Идентификаторы CSS</title>
      <style>
         div{
            margin: 10px;
            border: 1px solid #222;
         }
         #header{
            height: 80px;
            background-color: #ccc;
         }
         #content{
            height: 180px;
            background-color: #eee;
         }
         #footer{
            height: 80px;
            background-color: #ccc;
         }
      </style>
   </head>
   <body>
      <div id="header">Шапка сайта</div>
      <div id="content">Основное содержимое</div>
      <div id="footer">Футер</div>
   </body>
</html>
```



Основы CSS3

Универсальный селектор

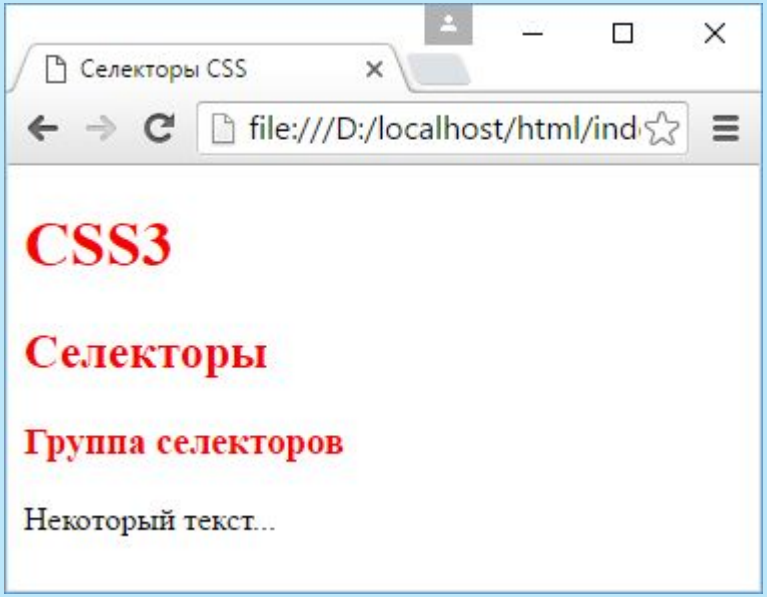
Кроме селекторов тегов, классов и идентификаторов в css также есть так называемый универсальный селектор, который представляет знак звездочки (*). Он применяет стили ко всем элементам на html-странице:

```
*{
    background-color: red;
}
```

Стилизация группы селекторов

Иногда определенные стили применяются к целому ряду селекторов. Например, мы хотим применить ко всем заголовкам подчеркивание. В этом случае мы можем перечислить селекторы всех элементов через запятую:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Селекторы CSS</title>
        <style>
            h1, h2, h3, h4{
                color: red;
            }
        </style>
    </head>
    <body>
        <h1>CSS3<h1>
        <h2>Селекторы</h2>
        <h3>Группа селекторов</h3>
        <p>Некоторый текст...</p>
    </body>
</html>
```



Основы CSS3

Веб-страница может иметь сложную организацию, одни элементы внутри себя могут определять другие элементы. Вложенные элементы иначе можно назвать потомками. А контейнер этих элементов - родителем.

Например, пусть элемент body на веб-странице имеет следующее содержимое:

```
<body>
    <h2>Заголовок</h2>
    <div>
        <p>Текст</p>
    </div>
</body>
```

Внутри элемента body определено три вложенных элемента: h2, div, p. Все эти элементы являются потомками элемента body.

А внутри элемента div определен только один вложенный элемент - p, поэтому элемент div имеет только одного потомка.

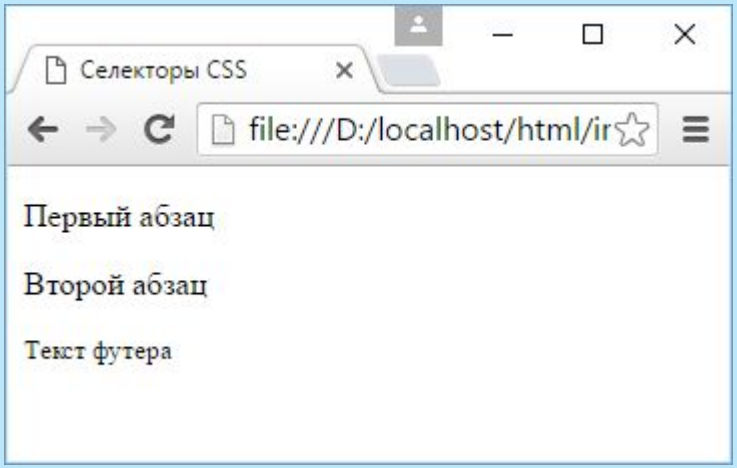
Для применения стиля к вложенному элементу селектор должен содержать вначале родительский элемент и затем вложенный:

```
#main p{
    font-size: 16px;
}
```

То есть данный стиль будет применяться только к тем элементам p, которые находятся внутри элемента с идентификатором main.

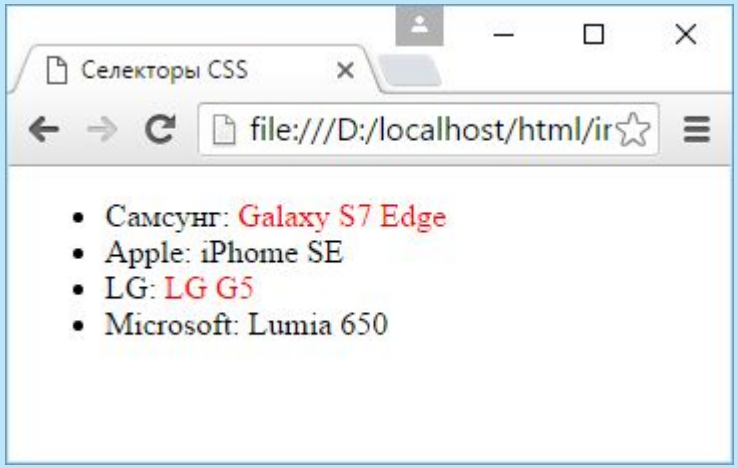
Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы CSS</title>
      <style>
         #main p{
            font-size: 16px;
         }
         #footer p{
            font-size: 13px;
         }
      </style>
   </head>
   <body>
      <div id="main">
         <p>Первый абзац</p>
         <p>Второй абзац</p>
      </div>
      <div id="footer">
         <p>Текст футера</p>
      </div>
   </body>
</html>
```



Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы CSS</title>
      <style>
         li .redLink{
            color: red;
         }
      </style>
   </head>
   <body>
      <ul>
         <li>Самсунг: <a class="redLink">Galaxy S7 Edge</a></li>
         <li>Apple: <a>iPhome SE</a></li>
         <li>LG: <a class="redLink">LG G5</a></li>
         <li>Microsoft: <a>Lumia 650</a></li>
      </ul>
   </body>
</html>
```



Основы CSS3

Селекторы дочерних элементов отличаются от селекторов потомков тем, что позволяют выбрать элементы только первого уровня вложенности. Например:

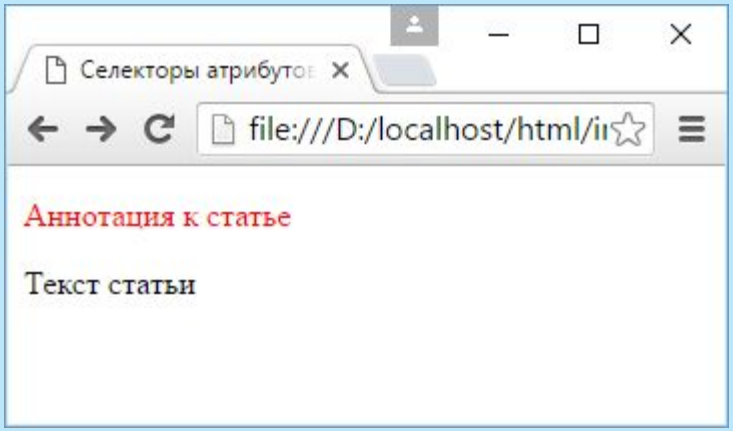
```
<body>
   <h2>Заголовок</h2>
   <div>
      <p>Текст</p>
   </div>
</body>
```

Хотя вложенными в элемент body элементами являются целых три - h2, div, p, но дочерними из них являются только два - div и h2, так как они находятся в первом уровне вложенности. А элемент p находится на втором уровне вложенности, так как вложен внутрь элемента div, а не просто элемента body.

Для обращения к дочерним элементам используется знак угловой скобки.

Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы атрибутов в CSS3</title>
      <style>
         .article > p{
           color: red;
         }
      </style>
   </head>
   <body>
      <div class="article">
         <p>Аннотация к статье</p>
         <div class="content">
            <p>Текст статьи</p>
         </div>
      </div>
   </body>
</html>
```



Основы CSS3

Селекторы элементов одного уровня или смежных элементов позволяют выбрать элементы, которые находятся на одном уровне вложенности. Иногда такие элементы еще называют сиблинги (siblings) или сестринскими элементами. Например:

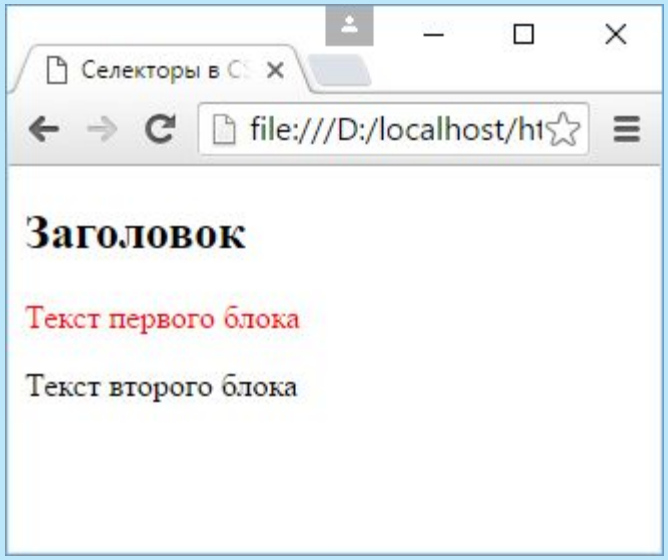
```
<body>
    <h2>Заголовок</h2>
    <div>
        <p>Текст первого блока</p>
    </div>
    <div>
        <p>Текст второго блока</p>
    </div>
</body>
```

Здесь элементы h2 и оба блока div являются смежными, так как находятся на одном уровне. А элементы параграфов и заголовок h2 не являются смежными, так как параграфы вложены в блоки div.

Чтобы стилизовать первый смежный элемент по отношению к определенному элементу, используется знак плюса +

Основы CSS3

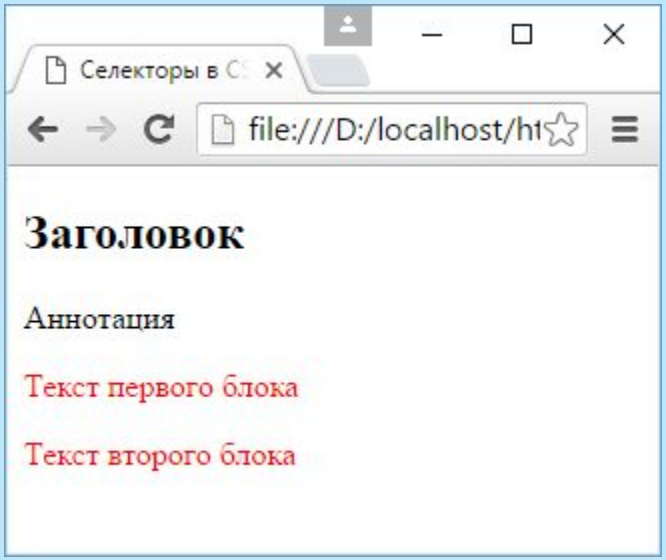
```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы в CSS3</title>
      <style>
         h2+div { color: red; }
      </style>
   </head>
   <body>
      <h2>Заголовок</h2>
      <div>
         <p>Текст первого блока</p>
      </div>
      <div>
         <p>Текст второго блока</p>
      </div>
   </body>
</html>
```



Основы CSS3

Если нам надо стилизовать вообще все смежные элементы одного уровня, неважно непосредственно идут они после определенного элемента или нет, то в этом случае вместо знака плюса необходимо использовать знак тильды "~":

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Селекторы в CSS3</title>
        <style>
            h2~div { color: red; }
        </style>
    </head>
    <body>
        <h2>Заголовок</h2>
        <p>Аннотация</p>
        <div>
            <p>Текст первого блока</p>
        </div>
        <div>
            <p>Текст второго блока</p>
        </div>
    </body>
</html>
```



Основы CSS3

В дополнение к селекторам тегов, классов и идентификаторов нам доступны селекторы псевдоклассов, которые несут дополнительные возможности по выбору нужных элементов.

Список доступных псевдоклассов:

:root: позволяет выбрать корневой элемент веб-страницы, наверное наименее полезный селектор, так как на правильной веб-странице корневым элементом практически всегда является элемент <html>

:link: применяется к ссылкам и представляет ссылку в обычном состоянии, по которой еще не совершен переход

:visited: применяется к ссылкам и представляет ссылку, по которой пользователь уже переходил

:active: применяется к ссылкам и представляет ссылку в тот момент, когда пользователь осуществляет по ней переход

:hover: представляет элемент, на который пользователь навел указатель мыши. Применяется преимущественно к ссылкам, однако может также применяться и к другим элементам, например, к параграфам

:focus: представляет элемент, который получает фокус, то есть когда пользователь нажимает клавишу табуляции или нажимает кнопкой мыши на поле ввода (например, текстовое поле)

:not: позволяет исключить элементы из списка элементов, к которым применяется стиль

:lang: стилизует элементы на основании значения атрибута lang

:empty: выбирает элементы, которые не имеют вложенных элементов, то есть являются пустыми

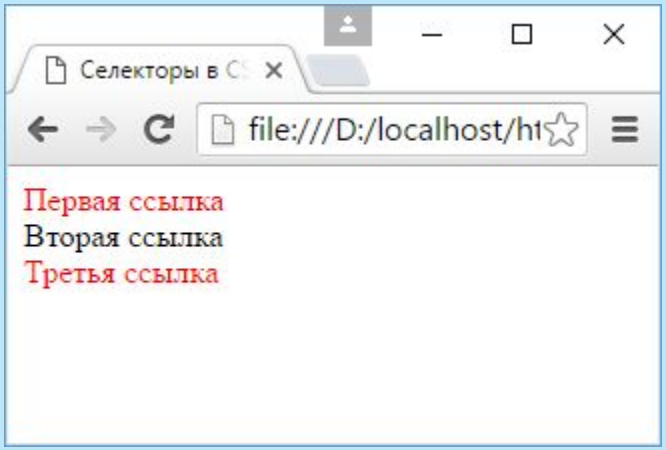
При применении псевдоклассов перед ними всегда ставится двоеточие.

Основы CSS3

Селектор :not

Селектор :not() позволяет выбрать все элементы кроме определенных, то есть исключить некоторые элементы из выбора.

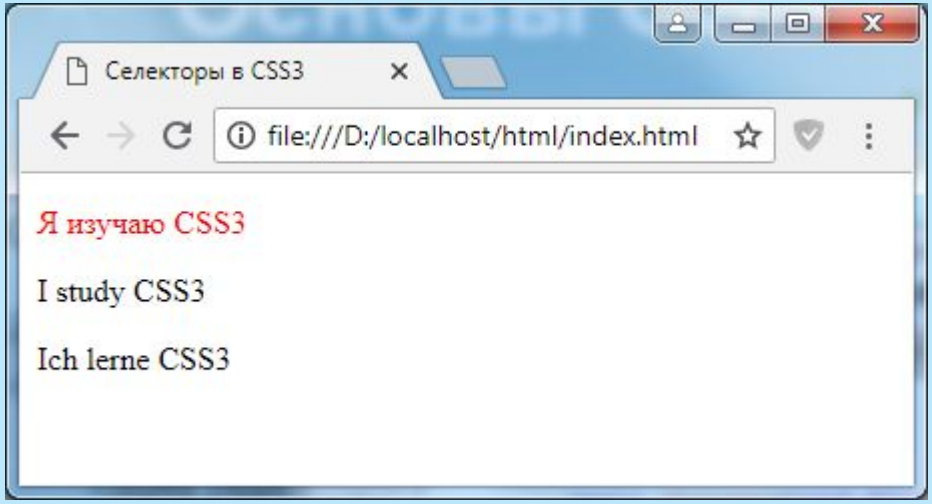
```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы в CSS3</title>
      <style>
         a:not(.blueLink) { color: red; }
      </style>
   </head>
   <body>
      <a>Первая ссылка</a><br/>
      <a class="blueLink">Вторая ссылка</a><br/>
      <a>Третья ссылка</a>
   </body>
</html>
```



Основы CSS3

Псевдокласс :lang
Селектор :lang выбирает элементы на основании атрибута lang:

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы в CSS3</title>
      <style>
         :lang(ru) {
            color: red;
         }
      </style>
   </head>
   <body>
      <form>
         <p lang="ru-RU">Я изучаю CSS3</p>
         <p lang="en-US">I study CSS3</p>
         <p lang="de-DE">Ich lerne CSS3</p>
      </form>
   </body>
</html>
```



Основы CSS3

Особую группу псевдоклассов образуют псевдоклассы, которые позволяют выбрать определенные дочерние элементы:

:first-child: представляет элемент, который является первым дочерним элементом

:last-child: представляет элемент, который является последним дочерним элементом

:only-child: представляет элемент, который является единственным дочерним элементом в каком-нибудь контейнере

:only-of-type: выбирает элемент, который является единственным элементом определенного типа (тега) в каком-нибудь контейнере

Основы CSS3

:nth-child(n): представляет дочерний элемент, который имеет определенный номер n, например, второй дочерний элемент

:nth-last-child(n): представляет дочерний элемент, который имеет определенный номер n, начиная с конца

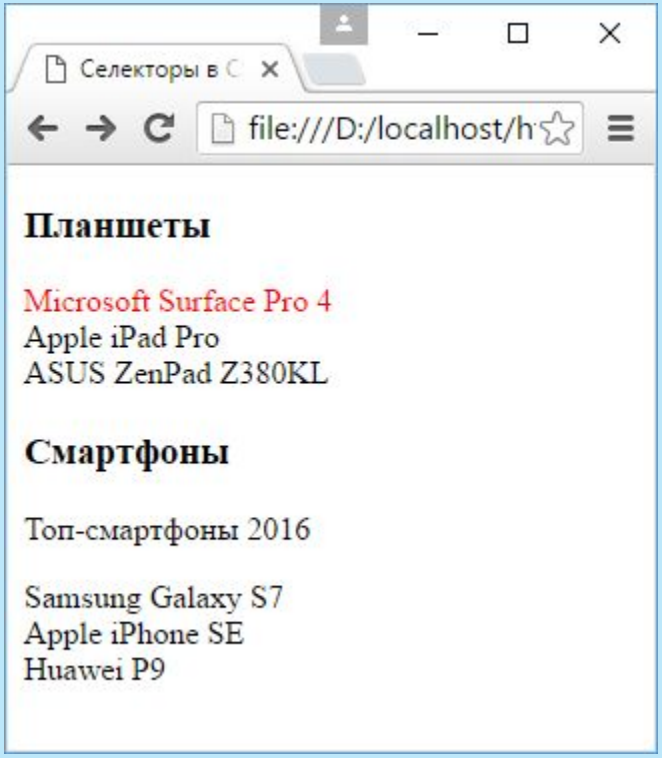
:nth-of-type(n): выбирает дочерний элемент определенного типа, который имеет определенный номер

:nth-last-of-type(n): выбирает дочерний элемент определенного типа, который имеет определенный номер, начиная с конца

Основы CSS3

Псевдокласс first-child. Используем псевдокласс first-child для выбора первых ссылок в блоках:

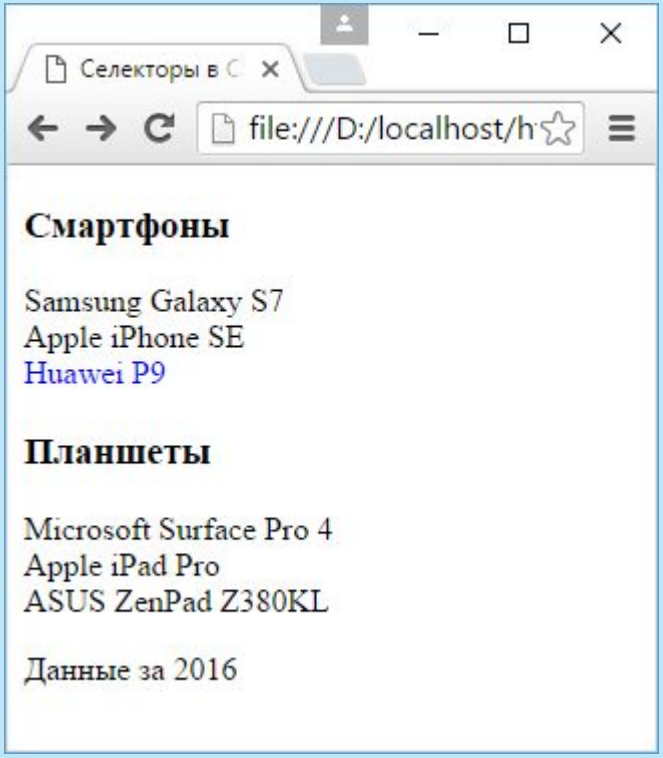
```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы в CSS3</title>
      <style>
         a:first-child{
            color: red;
         }
      </style>
   </head>
   <body>
      <h3>Планшеты</h3>
      <div>
         <a>Microsoft Surface Pro 4</a><br/>
         <a>Apple iPad Pro</a><br/>
         <a>ASUS ZenPad Z380KL</a>
      </div>
      <h3>Смартфоны</h3>
      <div>
         <p>Топ-смартфоны 2016</p>
         <a>Samsung Galaxy S7</a><br/>
         <a>Apple iPhone SE</a><br/>
         <a>Huawei P9</a>
      </div>
   </body>
</html>
```



Основы CSS3

Псевдокласс last-child. Используем псевдокласс last-child:

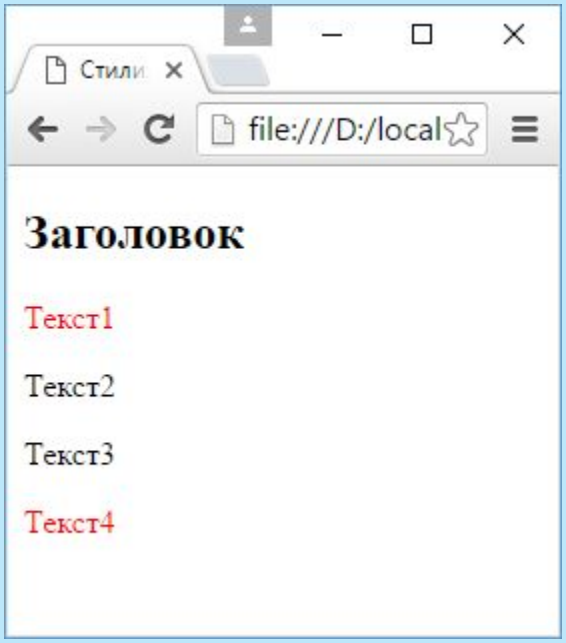
```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы в CSS3</title>
      <style>
         a:last-child{
           color: blue;
         }
      </style>
   </head>
   <body>
      <h3>Смартфоны</h3>
      <div>
         <a>Samsung Galaxy S7</a><br/>
         <a>Apple iPhone SE</a><br/>
         <a>Huawei P9</a>
      </div>
      <h3>Планшеты</h3>
      <div>
         <a>Microsoft Surface Pro 4</a><br/>
         <a>Apple iPad Pro</a><br/>
         <a>ASUS ZenPad Z380KL</a>
         <p>Данные за 2016</p>
      </div>
   </body>
</html>
```



Основы CSS3

Селектор only-child. Селектор :only-child выбирает элементы, которые являются единственными дочерними элементами в контейнерах:

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы в CSS3</title>
      <style>
      p:only-child{
         color:red;
      }
      </style>
   </head>
   <body>
      <h2>Заголовок</h2>
      <div>
         <p>Текст1</p>
      </div>
      <div>
         <p>Текст2</p>
         <p>Текст3</p>
      </div>
      <div>
         <p>Текст4</p>
      </div>
   </body>
</html>
```

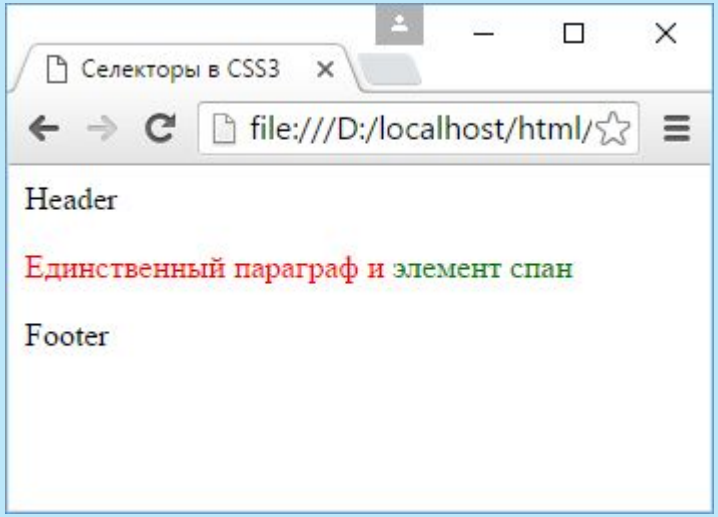


Параграфы с текстами "Текст1" и "Текст4" являются единственными дочерними элементами в своих внешних контейнерах, поэтому к ним применяется стиль - красный цвет шрифта.

Основы CSS3

Псевдокласс only-of-type. Псевдокласс only-of-type выбирает элемент, который является единственным элементом определенного типа в контейнере. Например, единственный элемент div, при этом элементов других типов в этом же контейнере может быть сколько угодно.

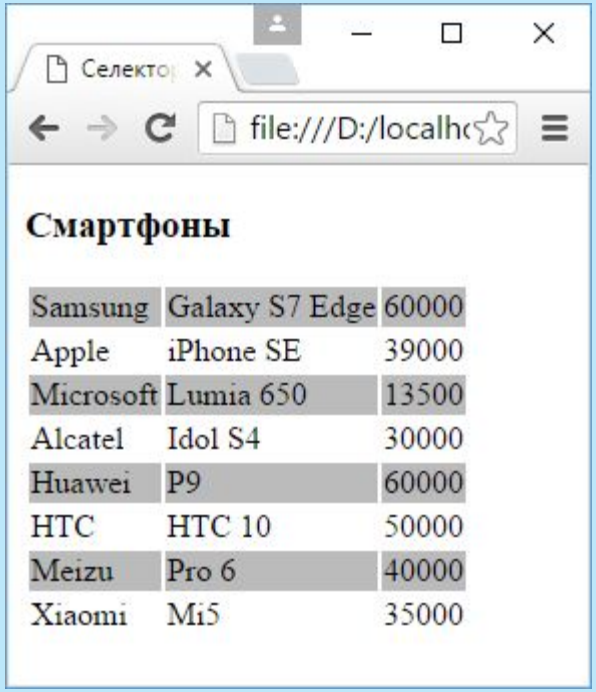
```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы в CSS3</title>
      <style>
      span:only-of-type{
         color: green;   /* зеленый цвет */
      }
      p:only-of-type{
         color: red; /* красный цвет */
      }
      div:only-of-type{
         color: blue;    /* синий цвет */
      }
      </style>
   </head>
   <body>
      <div>
         Header
      </div>
      <p>Единственный параграф и <span>элемент спан</span></p>
      <div>
         Footer
      </div>
   </body>
</html>
```



Основы CSS3

Псевдокласс nth-child. Псевдокласс nth-child позволяет стилизовать каждый второй, третий элемент, только четные или только нечетные элементы и т.д. Например, стилизуем четные и нечетные строки таблицы:

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Селекторы в CSS3</title>
      <style>
         tr:nth-child(odd) { background-color: #bbb; }
         tr:nth-child(even) { background-color: #fff; }
      </style>
   </head>
   <body>
      <h3>Смартфоны</h3>
      <table>
         <tr><td>Samsung</td><td>Galaxy S7 Edge</td><td>60000</td></tr>
         <tr><td>Apple</td><td>iPhone SE</td><td>39000</td></tr>
         <tr><td>Microsoft</td><td>Lumia 650</td><td>13500</td></tr>
         <tr><td>Alcatel</td><td>Idol S4</td><td>30000</td></tr>
         <tr><td>Huawei</td><td>P9</td><td>60000</td></tr>
         <tr><td>HTC</td><td>HTC 10</td><td>50000</td></tr>
         <tr><td>Meizu</td><td>Pro 6</td><td>40000</td></tr>
         <tr><td>Xiaomi</td><td>Mi5</td><td>35000</td></tr>
      </table>
   </body>
</html>
```



Основы CSS3

Чтобы определить стиль для нечетных элементов, в селектор передается значение "odd":

tr:nth-child(odd){}
Для стилизации четных элементов в селектор передается значение "even":

tr:nth-child(even){}
Также в этот селектор мы можем передать номер стилизуемого элемента:

tr:nth-child(3) { background-color: #bbb; }
В данном случае стилизуется третья строка.

Еще одну возможность представляет использование заменителя для номера, который выражается буквой n:

tr:nth-child(2n+1) { background-color: #bbb; }
Здесь стиль применяется к каждой второй нечетной строке.

Число перед n (в данном случае 2) представляет тот дочерний элемент, который будет выделен следующим. Число, которое идет после знака плюс, показывают, с какого элемента нужно начинать выделение, то есть, +1 означает, что нужно начинать с первого дочернего элемента.

Таким образом, в данном случае выделение начинается с 1-го элемента, а следующим выделяется 2 * 1 + 1 = 3-й элемент, далее 2 * 2 + 1 = 5-й элемент и так далее.

Основы CSS3

Псевдокласс :nth-of-type позволяет выбрать дочерний элемент определенного типа по определенному номеру:

```
tr:nth-of-type(2) {
    background-color: #bbb;
}
```

Аналогично работает псевдокласс nth-last-of-type, только теперь отсчет элементов идет с конца:

```
tr:nth-last-of-type(2n) {
    background-color: #bbb;
}
```

Основы CSS3

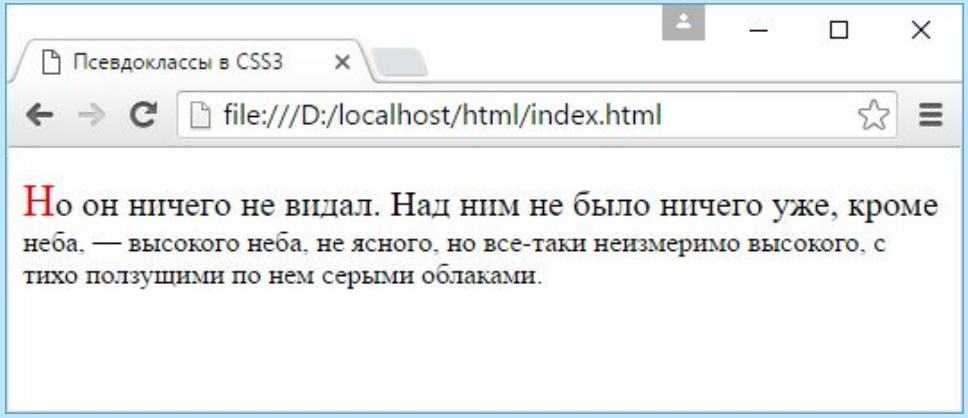
Псевдоэлементы обладают рядом дополнительных возможностей по выбору элементов веб-страницы и похожи на псевдокдассы.

Список доступных псевдоэлементов:

- **::first-letter**: позволяет выбрать первую букву из текста
- **::first-line**: стилизует первую строку текста
- **::before**: добавляет сообщение до определенного элемента
- **::after**: добавляет сообщение после определенного элемента
- **::selection**: выбирает выбранные пользователем элементы

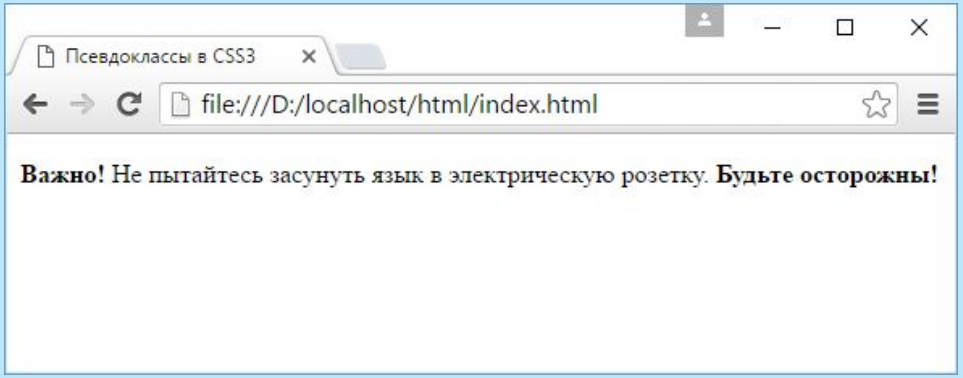
Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Псевдоклассы в CSS3</title>
      <style>
         ::first-letter { color:red; font-size: 25px; }
         ::first-line { font-size: 20px; }
      </style>
   </head>
   <body>
      <p>Но он ничего не видал. Над ним не было ничего уже, кроме неба, — высокого неба, не ясного, но все-таки неизмеримо высокого, с тихо ползущими по нем серыми облаками.</p>
   </body>
</html>
```



Основы CSS3

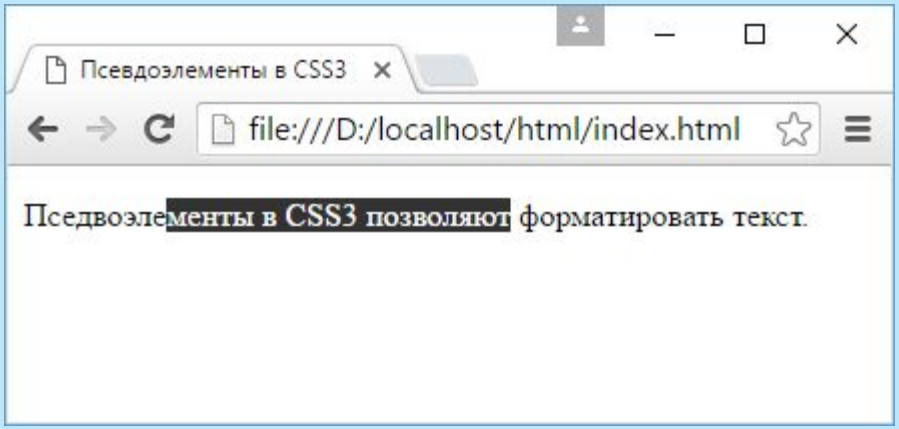
```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Псевдоклассы в CSS3</title>
      <style>
         .warning::before{ content: "Важно! "; font-weight: bold; }
         .warning::after { content: " Будьте осторожны!"; font-weight: bold;}
      </style>
   </head>
   <body>
      <p><span class="warning">Не пытайтесь засунуть язык в электрическую
розетку.</span></p>
   </body>
</html>
```



Основы CSS3

Используем псевдоэлемент selection для стилизации выбранных элементов:

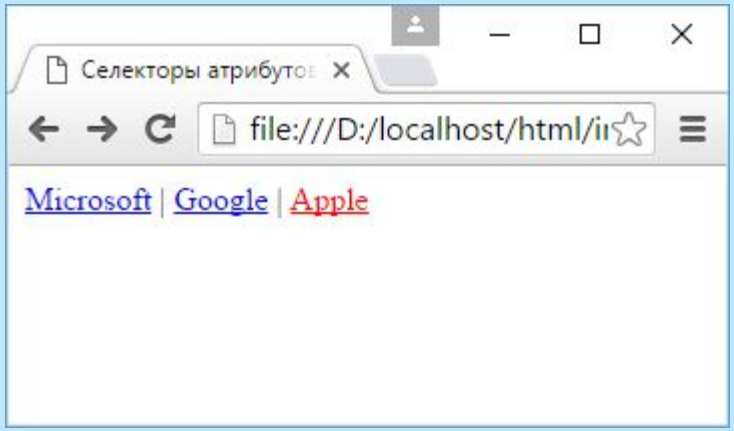
```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Псевдоэлементы в CSS3</title>
      <style>
         ::selection {
            color: white;
            background-color: black;
         }
      </style>
   </head>
   <body>
      <p>Пседвоэлементы в CSS3 позволяют форматировать текст.</p>
   </body>
</html>
```



Основы CSS3

Кроме селекторов элементов можно использовать селекторы их атрибутов.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Селекторы атрибутов в CSS3</title>
        <style>
            .link[href="http://apple.com"]{
                color: red;
            }
        </style>
    </head>
    <body>
        <a class="link" href="http://microsoft.com">Microsoft</a> |
        <a class="link" href="https://google.com">Google</a> |
        <a class="link" href="http://apple.com">Apple</a>
    </body>
</html>
```



Основы CSS3

Специальные символы позволяют конкретизировать значение атрибутов. Например символ ^ позволяет выбрать все атрибуты, которые начинаются на определенный текст.

Например, нам надо выбрать все ссылки, которые используют протокол https, то есть ссылка должна начинаться на "https://". В этом случае можно применить следующий селектор:

```
a[href^="https://"]{
    color: red;
}
```

Если значение атрибута должно иметь в конце определенный текст, то для проверки используется символ $. Например, нам надо выбрать все изображения в формате jpg. В этом случае мы можем проверить, оканчивается ли значение атрибута src на текст ".jpg":

```
img[src$=".jpg"]{
    width: 100px;
}
```

Основы CSS3

И еще один символ "*" (звездочка) позволяет выбрать все элементы с атрибутами, которые в своем значении имеют определенный текст (не важно где - в начале, середине или конце):

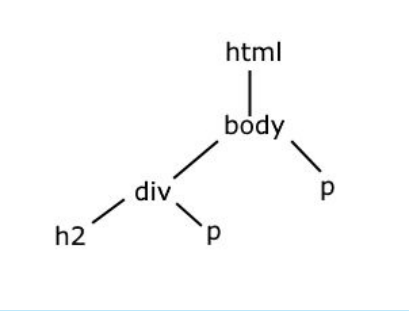
```
a[href*="microsoft"]{
    color: red;
}
```

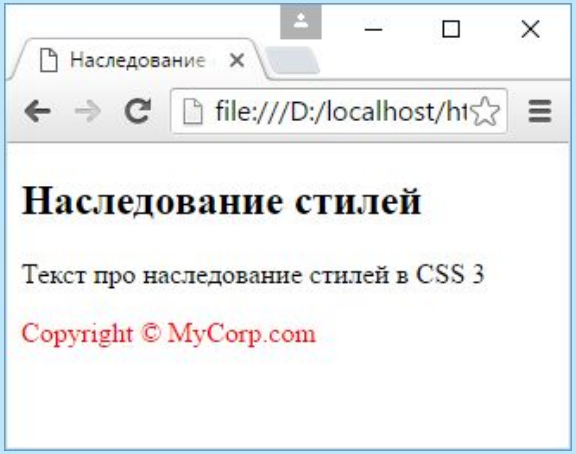
Данный атрибут выберет все ссылки, которые в своем адресе имеют текст "microsoft".

Основы CSS3

Для упрощения определения стилей в CSS применяется механизм наследования стилей. Этот механизм предполагает, что вложенные элементы могут наследовать стили своих элементов-контейнеров.

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Наследование стилей в CSS3</title>
      <style>
         body {color: red;}
         div {color:black;}
      </style>
   </head>
   <body>
      <div>
         <h2>Наследование стилей</h2>
         <p>Текст про наследование стилей в CSS 3</p>
      </div>
      <p>Copyright © MyCorp.com</p>
   </body>
</html>
```





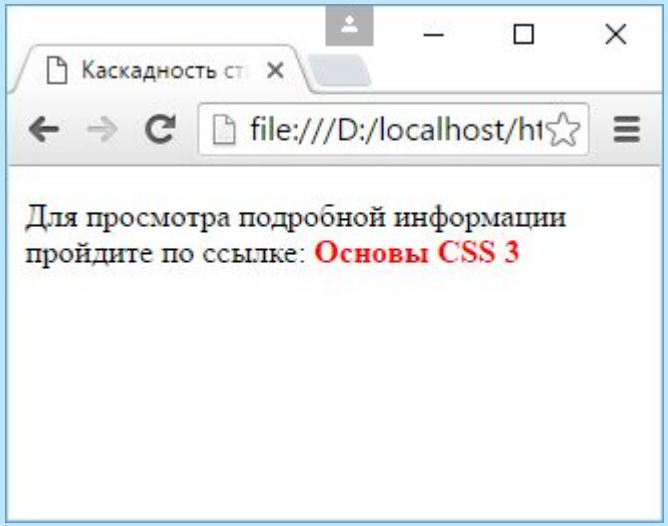
Основы CSS3

Когда к определенному элементу применяется один стиль, то все относительно просто. Однако если же к одному и тому же элементу применяется сразу несколько различных стилей, то возникает вопрос, какой же из этих стилей будет в реальности применяться?

В CSS действует механизм каскадности, которую можно определить как набор правил, определяющих последовательность применения множества стилей к одному и тому же элементу.

Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Каскадность стилей в CSS3</title>
      <style>
         .redLink {color: red;}  /* красный цвет текста */
         .footer a {font-weight: bold;}  /* выделение жирным */
         a {text-decoration: none;}  /* отмена подчеркивания ссылки */
      </style>
   </head>
   <body>
      <p class="footer">Для просмотра подробной информации пройдите по ссылке:
         <a class="redLink" href="index.php">Основы CSS 3</a></p>
   </body>
</html>
```



Основы CSS3

Если же стили конфликтуют между собой, например, определяют разный цвет текста, то в этом случае применяется сложная система правил для вычисления значимости каждого стиля. Все эти правила описаны в спецификации по CSS: Calculating a selectors specificity.

Для определения стиля к элементу могут применяться различные селекторы, и важность каждого селектора оценивается в баллах. Чем больше у селектора пунктов, тем он важнее, и тем больший приоритет его стили имеют над стилями других селекторов.

- Селекторы тегов имеют важность, оцениваемую в 1 балл.
- Селекторы классов, атрибутов и псевдоклассов оцениваются в 10 баллов.
- Селекторы идентификаторов оцениваются в 100 баллов.
- Встроенные inline-стили (задаваемые через атрибут style) оцениваются в 1000 баллов.

Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Каскадность стилей в CSS3</title>
      <style>
         #index {color: navy;}   /* темно-синий цет текста */
         .redLink {color: red; font-size: 20px;} /* красный цет текста и высота шрифта 20 пикселей */
         a {color: black; font-weight: bold;}    /* черный цет текста и выделение жирным */
      </style>
   </head>
   <body>
      <a id="index" class="redLink" href="index.php">Основы CSS 3</a>
   </body>
</html>
```

Основы CSS3

Здесь к ссылке применяется сразу три стиля. Эти стили содержат два не конфликтующих правила:

font-size: 20px;

font-weight: bold;

которые устанавливают высоту шрифта 20 пикселей и выделение ссылки жирным. Так как каждое из эти правил определено только в одном стиле, то в итоге они будут суммироваться и применяться к ссылке без проблем.

Кроме того, все три стиля содержат определение цвета текста, но каждый стиль определяет свой цвет текста. Так как селекторы идентификаторов имеют больший удельный вес, то в конечном счете будет применяться темно-синий цвет, задаваемый селектором:

#index {color: navy;}

Правило !important

CSS предоставляет возможность полностью отменить значимость стилей. Для этого в конце стиля указывается значение !important:

```
a {font-size: 18px; color: red !important;}
#menu a {color: navy;}
```

В этом случае вне зависимости от наличия других селекторов с большим количеством баллов к ссылкам будет применяться красный цвет, определяемый первым стилем.

Основы CSS3

Существует несколько различных способов определения цвета текста.

Шестнадцатеричного значение.

Оно состоит из отдельных частей, которые кодируют в шестнадцатеричной системе значения для красного, зеленого и синего цветов.

Например, #1C4463.

Здесь первые два символа 1C представляю значение красной компоненты цвета, далее 44 - значение зеленой компоненты цвета и 63 - значение уровня синего цвета. Финальный цвет, который мы видим на веб-странице, образуется с помощью смешивания этих значений.

Если каждое из трех двухзначных чисел содержит по два одинаковых символа, то их можно сократить до одного. Например, #5522AA можно сократить до #52A, или, к примеру, #eeeeee можно сократить до #eee. При этом не столь важно, в каком регистре будут символы.

Основы CSS3

Существует несколько различных способов определения цвета текста.

Значение RGB.
Значение RGB также представляет последовательно набор значений для красного, зеленого и синего цветов (Red — красный, Green — зеленый, Blue — синий). Значение каждого цвета кодируется тремя числами, которые могут представлять либо процентные соотношения (0–100%), либо число от 0 до 255.

background-color: rgb(100%,100%,100%);

Здесь каждый цвет имеет значение 100%. И в итоге при смешивании этих значений будет создаваться белый цвет. А при значениях в 0% будет генерироваться черный цвет:

background-color: rgb(0%, 0%, 0%);

Между 0 и 100% будут находиться все остальные оттенки.

Но, как правило, чаще применяются значения из диапазона от 0 до 255. Например,

background-color: rgb(28, 68, 99);

Основы CSS3

Существует несколько различных способов определения цвета текста.

Значение RGBA.
Это тоже самое значение RGB плюс компонент прозрачности (Alpha). Компонент прозрачности имеет значение от 0 (полностью прозрачный) до 1 (не прозрачный). Например:

background-color: rgba(28, 68, 99, .6);

Значение HSL.
HSL представляет аббривиатуру: Hue - тон, Saturation - насыщенность и Lightness - осветленность. HSL задает три значения.

Первое значение Hue угол в круге оттенков - значение в градусах от 0 до 360. Например, красный - 0 (или 360 при полном обороте круга). Каждый цвет занимает примерно 51°.

Второе значение - Saturation - представляет насыщенность, то указывает, насколько чистым является цвет. Насыщенность определяется в процентах от 0 (полное отсутствие насыщенности) до 100% (яркий, насыщенный цвет).

Третье значение - Lightness - определяет осветленность и указывается в процентах от 0 (полностью черный) до 100 (полностью белый). Для получения чистого цвет применяется значение 50 %.

background-color: hsl(206, 56%, 25%);
Данный цвет является эквивалентом значений #1C4463 и rgb(28, 68, 99)

Основы CSS3

Существует несколько различных способов определения цвета текста.

Значение HSLA. Аналогично RGBA здесь к HSL добавляется компонента прозрачности в виде значения от 0 (полностью прозрачный) до 1 (не прозрачный). Например:

background-color: hsl(206, 56%, 25%, .6);

Строковые значения. Существует ряд константных строковых значений, например, red (для красного цвета) или green (для зеленого цвета). К примеру,

color: red;

является эквивалентом

color: #ff0000;

Основы CSS3

Прозрачность

Ряд настроек цвета позволяют установить значение для альфа-компоненты, которая отвечает за прозрачность.

Но также в CSS есть специальное свойство, которое позволяет установить прозрачность элементов - свойство opacity.

В качестве значения оно принимает число от 0 (полностью прозрачный) до 1 (не прозрачный):

```
div{
    width: 100px;
    height: 100px;
    background-color: red;
    opacity: 0.4;
}
```

Основы CSS3

Семейство шрифтов

Свойство font-family устанавливает семейство шрифтов, которое будет использоваться. Например:

```
body{
    font-family: Arial;
}
```

В данном случае устанавливается шрифт Arial.

Шрифт свойства font-family будет работать, только если у пользователя на локальном компьютере имеется такой же шрифт. По этой причине нередко выбираются стандартные шрифты, которые широко распространены, как Arial, Verdana и т.д.

Также нередко применяется практика нескольких шрифтов:

```
body{
    font-family: Arial, Verdana, Helvetica;
}
```

В данном случае основным шрифтом является первый - Arial. Если он на компьютере пользователя не поддерживается, то выбирается второй и т.д.

Основы CSS3

Если название шрифта состоит из нескольких слов, например, Times New Roman, то все название заключается в кавычки:

```
body{
    font-family: "Times New Roman";
}
```

Кроме конкретных стилей также могут использоваться общие универсальные шрифты, задаваемые с помощью значений sans-serif и serif:

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

Так, если ни Arial, ни Verdana не поддерживаются на компьютере пользователя, то используется sans-serif - универсальный шрифт без засечек.

Основы CSS3

Типы шрифтов

Шрифты с засечками

- Шрифты с засечками названы так, потому что на на концах основных штрихов имеют небольшие засечки. Считается, что они подходят для больших кусков текста, так как визуально связывают одну букву с другой, делая текст более читабельным.
- Распространенные шрифты с засечками: Times, Times New Roman, Georgia, Garamond. Универсальный обобщенный шрифт с засечками представляет значение **serif**.

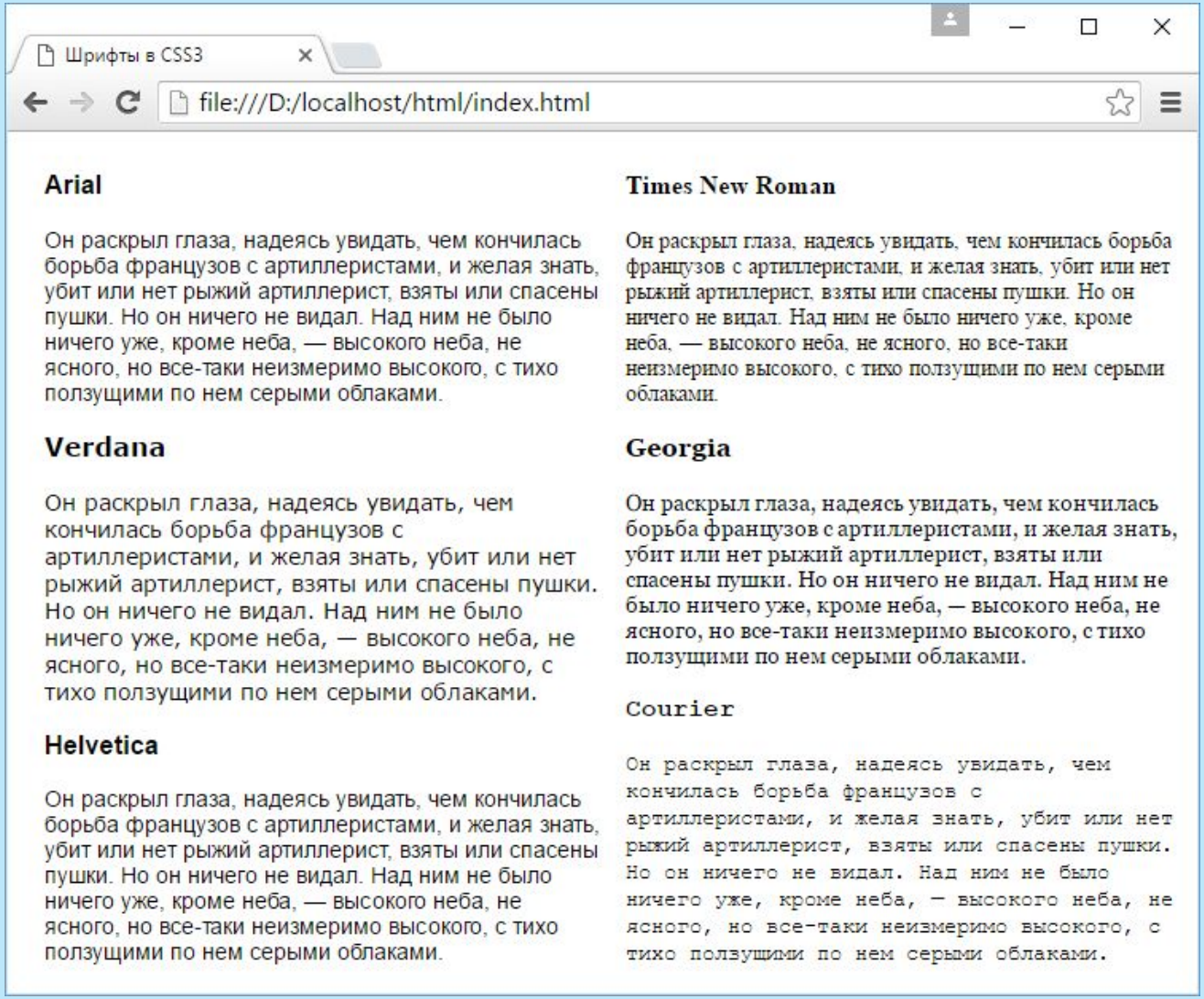
Шрифты без засечек

- В отличие от шрифтов с засечками шрифты из этой группы не имеют засечек. Наиболее распространенные шрифты этой группы: Arial, Helvetica, Verdana.

Моноширинные шрифты

- Моноширинный шрифт преимущественно применяется для отображения программного кода и не предназначен для вывода стандартного текста статей. Свое название эти шрифты получили от того, что каждая буква в таком шрифте имеет одинаковую ширину. Примеры подобных шрифтов: Courier, Courier New, Consolas, Lucida Console.

Основы CSS3


Шрифты в CSS3
file:///D:/localhost/html/index.html
Arial
Он раскрыл глаза, надеясь увидать, чем кончилась борьба французов с артиллеристами, и желая знать, убит или нет рыжий артиллерист, взяты или спасены пушки. Но он ничего не видал. Над ним не было ничего уже, кроме неба, — высокого неба, не ясного, но все-таки неизмеримо высокого, с тихо ползущими по нем серыми облаками.
Verdana
Он раскрыл глаза, надеясь увидать, чем кончилась борьба французов с артиллеристами, и желая знать, убит или нет рыжий артиллерист, взяты или спасены пушки. Но он ничего не видал. Над ним не было ничего уже, кроме неба, — высокого неба, не ясного, но все-таки неизмеримо высокого, с тихо ползущими по нем серыми облаками.
Helvetica
Он раскрыл глаза, надеясь увидать, чем кончилась борьба французов с артиллеристами, и желая знать, убит или нет рыжий артиллерист, взяты или спасены пушки. Но он ничего не видал. Над ним не было ничего уже, кроме неба, — высокого неба, не ясного, но все-таки неизмеримо высокого, с тихо ползущими по нем серыми облаками.
Times New Roman
Он раскрыл глаза, надеясь увидать, чем кончилась борьба французов с артиллеристами, и желая знать, убит или нет рыжий артиллерист, взяты или спасены пушки. Но он ничего не видал. Над ним не было ничего уже, кроме неба, — высокого неба, не ясного, но все-таки неизмеримо высокого, с тихо ползущими по нем серыми облаками.
Georgia
Он раскрыл глаза, надеясь увидать, чем кончилась борьба французов с артиллеристами, и желая знать, убит или нет рыжий артиллерист, взяты или спасены пушки. Но он ничего не видал. Над ним не было ничего уже, кроме неба, — высокого неба, не ясного, но все-таки неизмеримо высокого, с тихо ползущими по нем серыми облаками.
Courier
Он раскрыл глаза, надеясь увидать, чем кончилась борьба французов с артиллеристами, и желая знать, убит или нет рыжий артиллерист, взяты или спасены пушки. Но он ничего не видал. Над ним не было ничего уже, кроме неба, — высокого неба, не ясного, но все-таки неизмеримо высокого, с тихо ползущими по нем серыми облаками.

Основы CSS3

Толщина шрифта

Свойство font-weight задает толщину шрифта. Оно может принимать 9 числовых значений: 100, 200, 300, 400,...900. 100 - очень тонкий шрифт, 900 - очень плотный шрифт.

В реальности чаще для этого свойства используют два значения: normal (нежирный обычный текст) и bold (полужирный шрифт):

font-weight: normal;

font-weight: bold;

Курсив

Свойство font-style позволяет выделить текст курсивом. Для этого используется значение italic:

```
p {font-style: italic;}
```

Если надо отменить курсив, то применяется значение normal:

```
p {font-style: normal;}
```

Цвет шрифта

Свойство color устанавливает цвет шрифта:

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

Основы CSS3

Не всегда стандартные встроенные шрифты, как Arial или Verdana, могут быть удобны. Нередко встречается ситуация, когда веб-дизайнер хочет воспользоваться возможностями какого-то другого шрифта, которого нет среди встроенных, но который доступен из внешнего файла.

Такой шрифт можно подключить с помощью директивы font-face:

```
@font-face {
    font-family: 'Roboto';
   src:
url(http://fonts.gstatic.com/s/roboto/v15/mErvLBYg_cXG3rLvUsKT_fesZW2xOQ-xsNqO47m55DA.woff2);
}
```

Свойство font-family задает название шрифта, а свойство src - путь к шрифту.

Основы CSS3

В качестве альтернативы можно загрузить файл шрифта на локальный компьютер и уже оттуда подгружать его на веб-страницу. Как правило, для хранения своих шрифтов рядом с веб-страницей создается папка fonts:

```
@font-face{
    font-family: 'Roboto';
    src:url('fonts/roboto.ttf');
}
```

После подключения шрифта, его можно использовать в стилях:

```
p{
    font-family: Roboto;
}
```

Основы CSS3

Существует несколько различных форматов шрифтов:

- TrueType(расширение ttf),
- Open Type (расширение otf),
- Embedded Open Type (расширение eot),
- Web Open Font Format (woff/woff2),
- Scalable Vector Graphic (svg).

Основы CSS3

```
@font-face {
    font-family:'FontAwesome';
    src:
url('https://maxcdn.bootstrapcdn.com/font-awesome/4.6.1/fonts/fontawesome-webfont.
eot');
    src:
url('https://maxcdn.bootstrapcdn.com/font-awesome/4.6.1/fonts/fontawesome-webfont.
eot?#iefix') format('embedded-opentype'),

url('https://maxcdn.bootstrapcdn.com/font-awesome/4.6.1/fonts/fontawesome-webfont.
woff2') format('woff2'),

url('https://maxcdn.bootstrapcdn.com/font-awesome/4.6.1/fonts/fontawesome-webfont.
woff') format('woff'),

url('https://maxcdn.bootstrapcdn.com/font-awesome/4.6.1/fonts/fontawesome-webfont.t
tf') format('truetype'),

url('https://maxcdn.bootstrapcdn.com/font-awesome/4.6.1/fonts/fontawesome-webfont.
```

Основы CSS3

Для установки размера шрифта используется свойство font-size:

```
div{
    font-size: 18px;
}
```

В данном случае высота шрифта составит 18 пикселей. Пиксели представляют наиболее часто используемые единицы измерения. Чтобы задать значение в пикселях, после самого значения идет сокращение "px".

Если к тексту явным образом не применяется высота шрифта, то используются значения браузера по умолчанию. Например, для простого текста в параграфах это 16 пикселей. Это базовый стиль текста.

Базовый стиль для разных элементов текста отличается: если для параграфов это 16 пикселей, то для заголовков h1 это 32 пикселя, для заголовков h2 - 24 пикселя и т..д.

Основы CSS3

Для измерения шрифта также можно использовать самые разные единицы измерения.

Ключевые слова
В CSS имеется семь ключевых слов, которые позволяют назначить размер шрифта относительно базового:

- medium: базовый размер шрифта браузера (16 пикселей)
- small: 13 пикселей
- x-small: 10 пикселей
- xx-small: 9 пикселей
- large: 18 пикселей
- x-large: 24 пикселя
- xx-large: 32 пикселя

Основы CSS3

Проценты
Проценты позволяют задать значение относительно базового или унаследованного шрифта. Например:

font-size: 150%;

В данном случае высота шрифта будет составлять 150% от базового, то есть 16px * 1,5 = 24px

Наследование шрифта может изменить финальное значение. Например, следующую ситуацию:

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Шрифты в CSS3</title>
      <style>
         div {font-size: 10px;}
         p {font-size: 150%;}
      </style>
   </head>
   <body>
   <div>
      <p>Однажды в студеную зимнюю пору</p>
   </div>
   </body>
</html>
```

Единица em
Единица измерения em во многом эквивалентна процентам. Так, 1em равен 100%, .5em равно 50% и т.д.

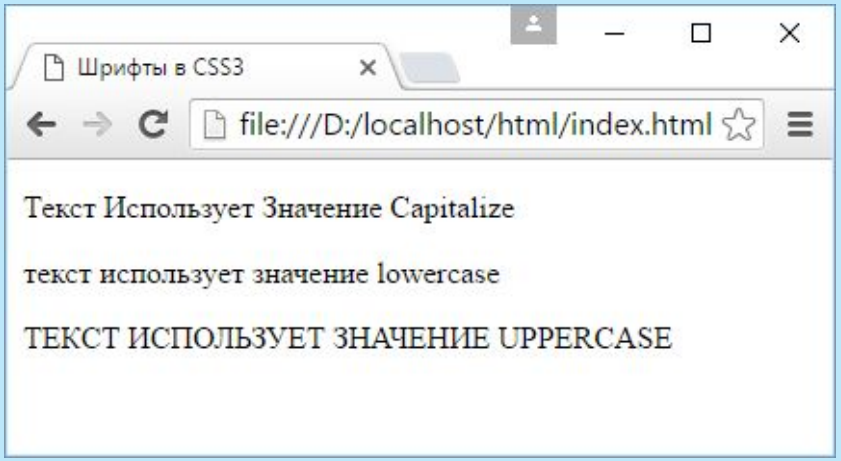
Основы CSS3

Свойство **text-transform** изменяет регистр текста.
Оно может принимать следующие значения:

- capitalize: делает первую букву слова заглавной
- uppercase: все слово переводится в верхний регистр
- lowercase: все слово переводится в нижний регистр
- none: регистр символов слова никак не изменяется

Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Шрифты в CSS3</title>
      <style>
         p.lowercase {text-transform: lowercase;}
         p.uppercase {text-transform: uppercase;}
         p.capitalize { text-transform: capitalize;}
      </style>
   </head>
   <body>
   <div>
      <p class="capitalize">Текст использует значение capitalize</p>
      <p class="lowercase">Текст использует значение lowercase</p>
      <p class="uppercase">Текст использует значение uppercase</p>
   </div>
   </body>
</html>
```



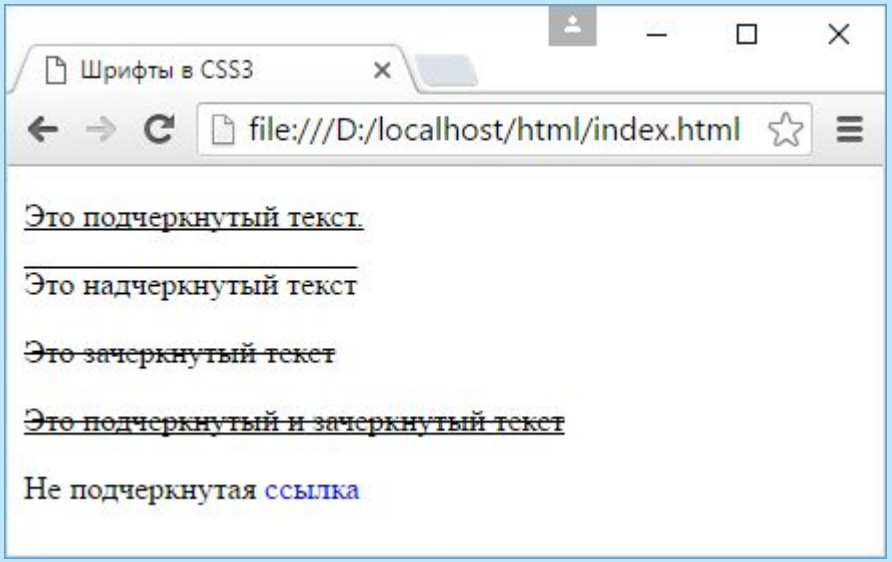
Основы CSS3

Свойство **text-decoration** позволяет добавить к тексту некоторые дополнительные эффекты. Это свойство может принимать следующие значения:

- underline: подчеркивает текст
- overline: надчеркивает текст, проводит верхнюю линию
- line-through: зачеркивает текст
- none: к тексту не применяется декорирование

Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Шрифты в CSS3</title>
      <style>
         p.under { text-decoration: underline;}
         p.over {text-decoration: overline;}
         p.line {text-decoration: line-through;}
         p.mixed {text-decoration: underline line-through;}
         a.none {text-decoration: none;}
      </style>
   </head>
   <body>
   <div>
      <p class="under">Это подчеркнутый текст.</p>
      <p class="over">Это надчеркнутый текст</p>
      <p class="line">Это зачеркнутый текст</p>
      <p class="mixed">Это подчеркнутый и зачеркнутый текст</p>
      <p>Не подчеркнутая <a href="index.php" class="none">ссылка<a></p>
   </div>
   </body>
</html>
```



Основы CSS3

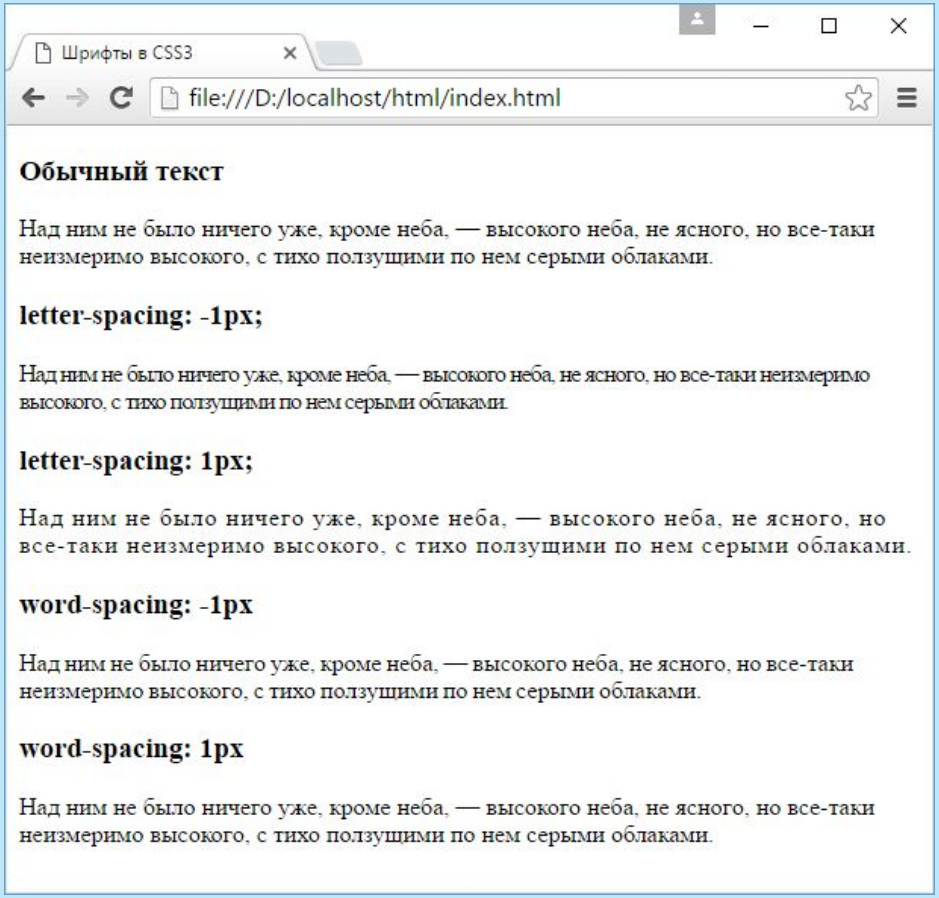
Два свойства CSS позволяют управлять интервалом между символами и словами текста.

Для межсимвольного интервала применяется атрибут **letter-spacing**,

а для интервала между словами **word-spacing**.

Основы CSS3

```
<style>
  p.smallLetterSpace {
  letter-spacing: -1px;
}
  p.bigLetterSpace {
  letter-spacing: 1px;
}
  p.smallWordSpace{
  word-spacing: -1px;
}
  p.bidWordSpace{
word-spacing: 1px;
}
</style>
```



Основы CSS3

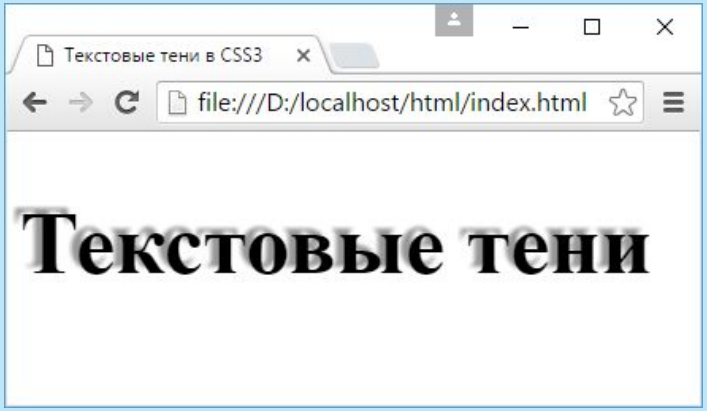
С помощью свойства **text-shadow** можно создать тени для текста.

Для этого свойства необходимо задать четыре значения:

- горизонтальное смещение тени относительно текста,
- вертикальное смещение тени относительно текста,
- степень размытости тени
- цвет отбрасываемой тени.

Например:

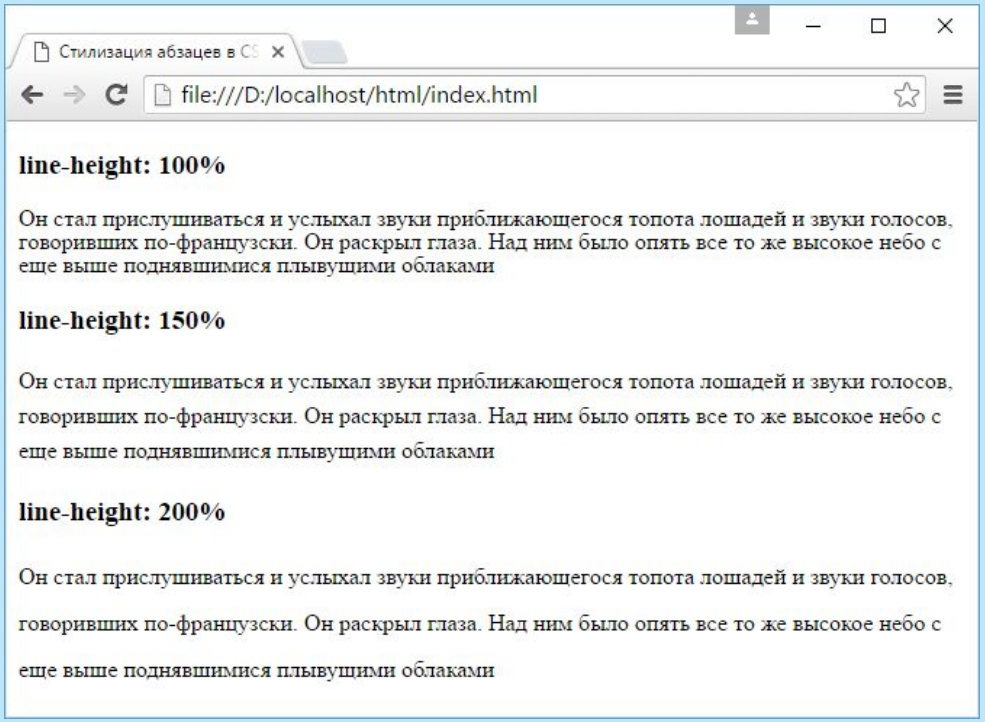
```
h1{
    text-shadow: 5px 4px 3px #999;
}
```



Основы CSS3

Свойство **line-height** определяет межстрочный интервал. Для его установки можно использовать пиксели, проценты или единицы em. Как правило, применяются либо проценты, либо em. Например:

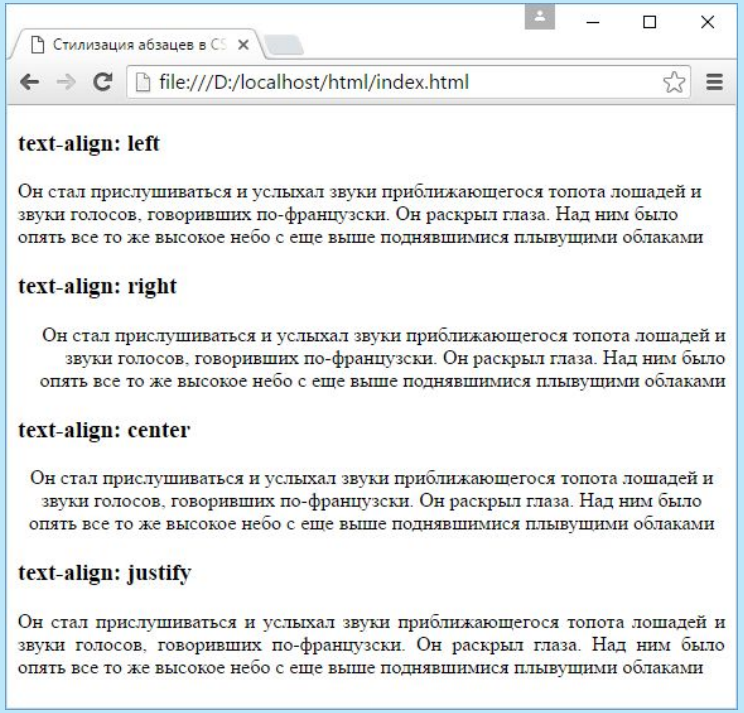
```
p{
    line-height: 150%;
}
```



Основы CSS3

Свойство **text-align** выравнивает текст относительно одной из сторон веб-страницы. Оно принимает следующие значения:

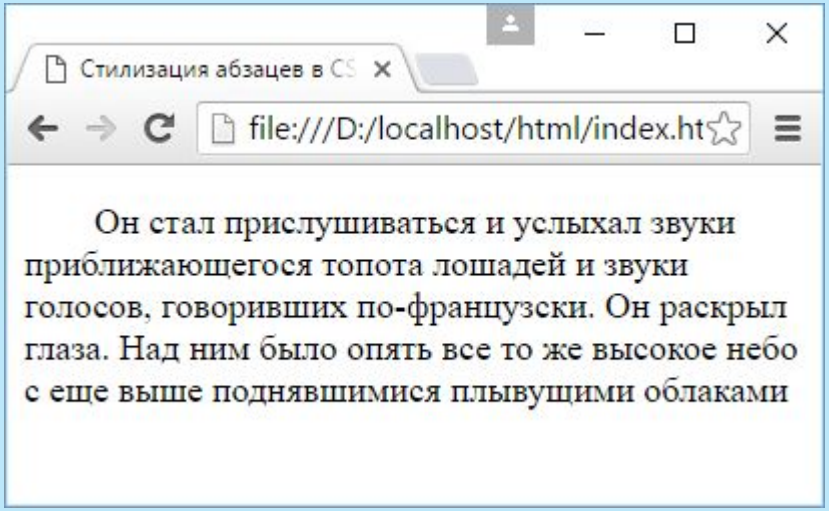
- left: текст выравнивается по левой стороне
- right: текст выравнивается по правой
- justify: выравнивание по ширине
- center: выравнивание по центру



Основы CSS3

Свойство **text-indent** задает отступ первой строки абзаца. Для установки отступа могут применяться стандартные единицы измерения, например, em или пиксели:

```
p{
    text-indent: 35px;
}
```



Основы CSS3

CSS предоставляет специальные свойства по стилизации списков. Одним из таких свойств является **list-style-type**.

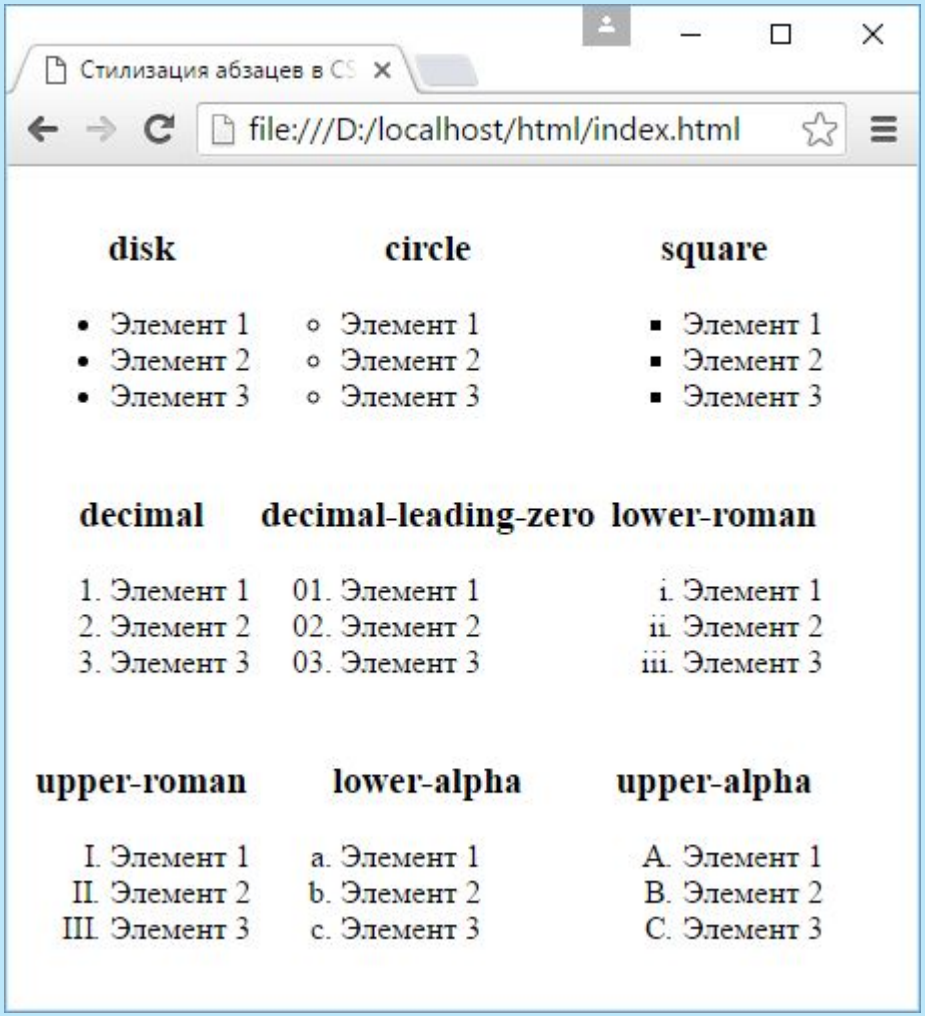
Значения для нумерованных списков:

- decimal: десятичные числа, отсчет идет от 1
- decimal-leading-zero: десятичные числа, которые предваряются нулем, например, 01, 02, 03, … 98, 99
- lower-roman: строчные латинские цифры, например, i, ii, iii, iv, v
- upper-roman: заглавные латинские цифры, например, I, II, III, IV, V…
- lower-alpha: строчные латинские буквы, например, a, b, c..., z
- upper-alpha: заглавные латинские буквы, например, A, B, C, … Z

Основы CSS3

Для ненумерованных списков:

- disc: черный диск
- circle: пустой кружочек
- square: черный квадратик



Основы CSS3

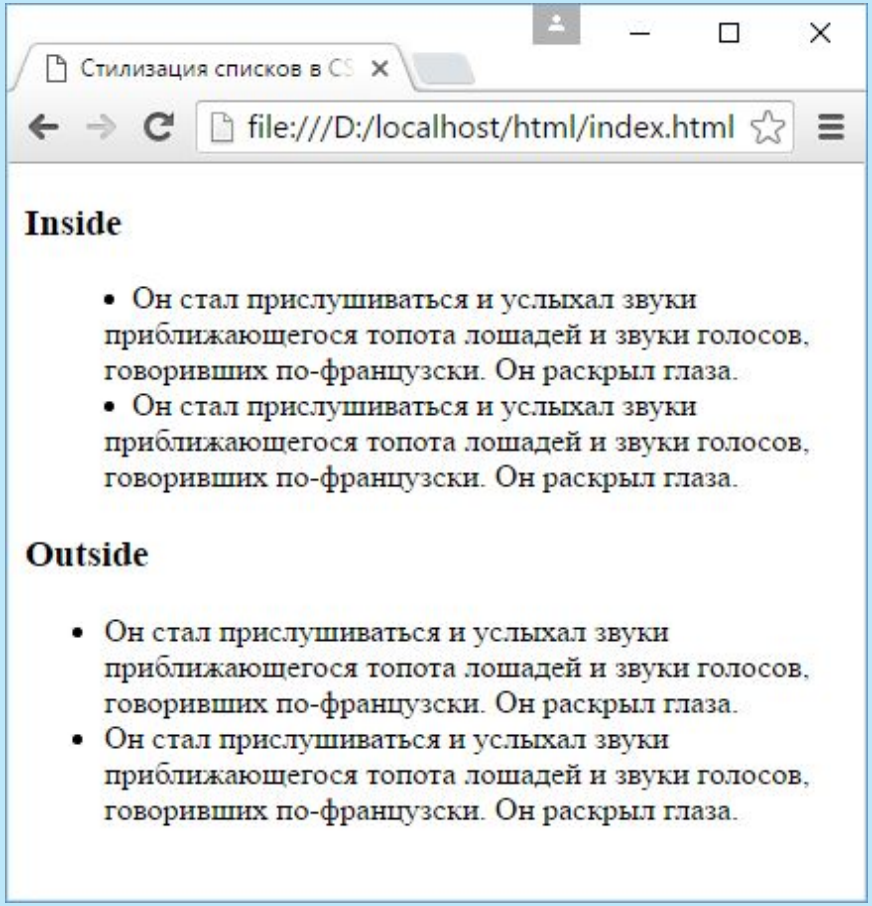
Свойство **list-style-position** отвечает за позиционирование списка.

Это свойство принимает два значения:

- outside (по умолчанию)
- inside (обеспечивает равномерное распределение по ширине).

Основы CSS3

```
<title>Стилизация списков в CSS3</title>
<style>
    ul.outside{
      list-style-position: outside;
    }
     ul.inside{
       list-style-position: inside;
     }
</style>
```



Основы CSS3

CSS предоставляет ряд свойств, которые помогают стилизовать таблицу:

- **border-collapse**: устанавливает, как будет стилизоваться граница смежных ячеек
- **border-spacing**: устанавливает промежутки между границами смежных ячеек
- **caption-side**: устанавливает положение элемента caption
- **empty-cells**: задает режим отрисовки для пустых ячеек
- **table-layout**: определяет размеры таблицы

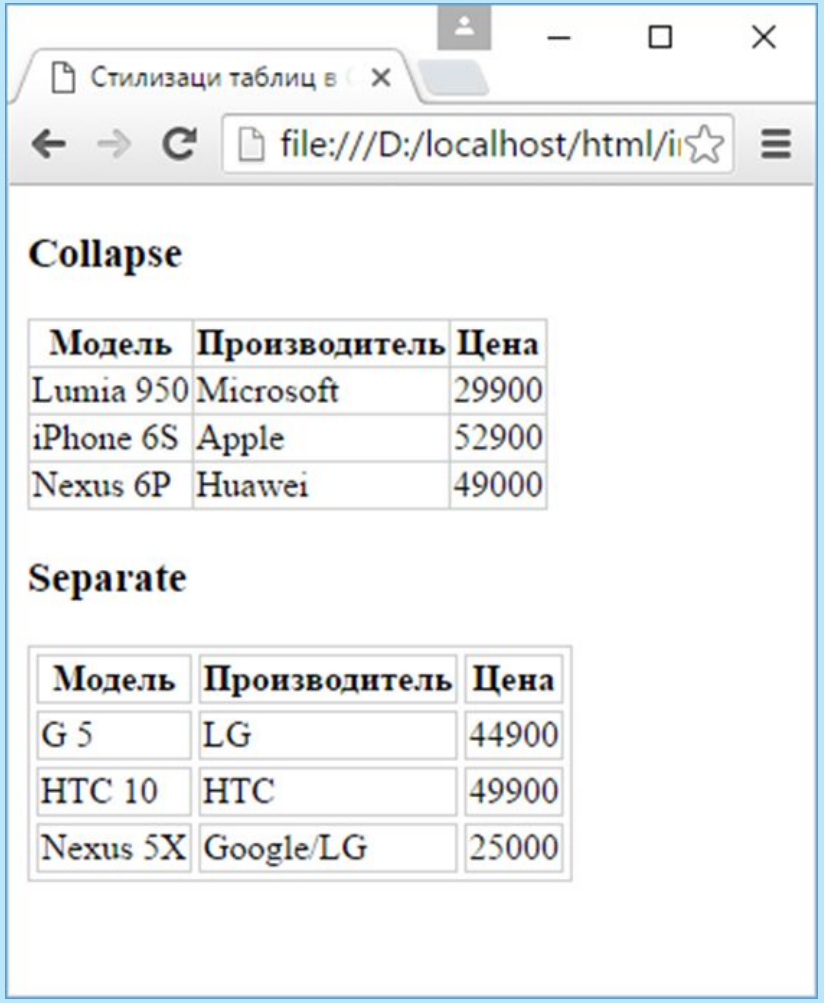
Основы CSS3

```
table {
    border: 1px solid #ccc;  /* граница всей таблицы */
}
tr {
    border: 1px solid #ccc;  /* границы между строками */
}
td, th {
    border: 1px solid #ccc;  /* границы между столбцами */
}
```

Основы CSS3

При установке границ между столбцами с помощью свойства **border-collapse** можно установить общую или раздельную границу между смежными ячейками:

- collapse: смежные ячейки имеют общую границу
- separate: смежные ячейки имеют отдельные границы, которые разделяются пространством



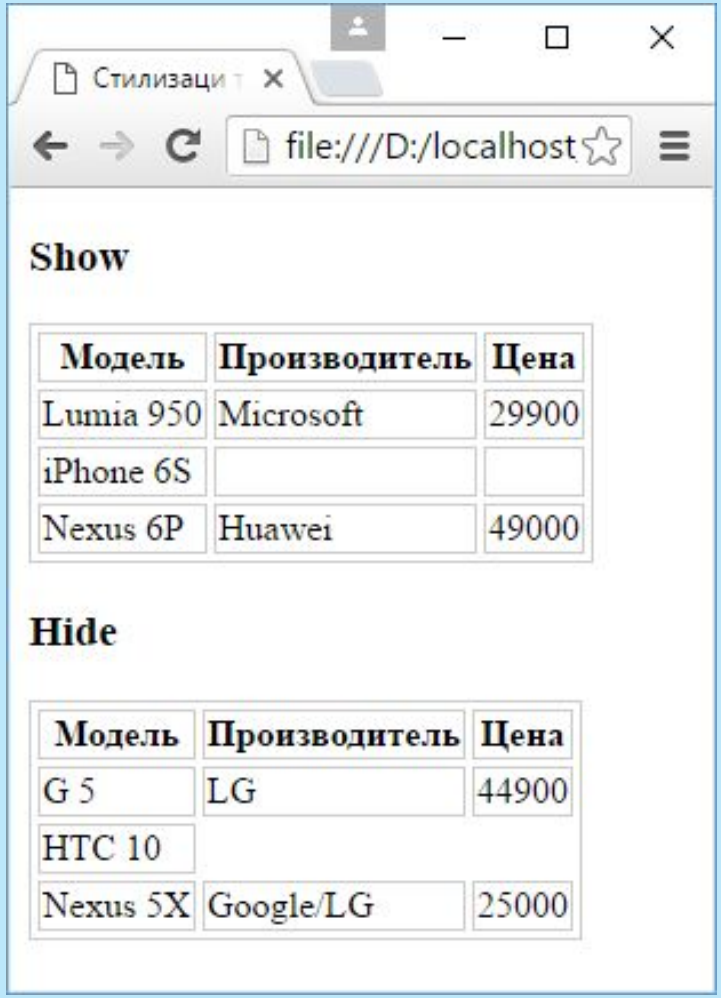
Основы CSS3

Свойство **empty-cells** позволяет стилизовать пустые ячейки с помощью одного из следующих значений:

- **show**: пустые ячейки отображаются, значение по умолчанию
- **hide**: пустые ячейки не отображаются

Основы CSS3

```
<title>Стилизаци таблиц в CSS3</title>
 <style>
   table {
      border: 1px solid #ccc;
      border-spacing: 3px;
    }
    td, th{
       border: solid 1px #ccc;
    }
    .hidden-empty-cells{
        empty-cells: hide;
    }
</style>
```



Основы CSS3

С помощью свойства **table-layout** можно управлять размером таблицы.

По умолчанию это свойство имеет значение auto, при котором браузер устанавливает ширину столбцов таблицы автоматически, исходя из ширины самой широкой ячейки в столбце.

С помощью другого значения - fixed можно установить фиксированную ширину:

```
table {
    border: 1px solid #ccc;
    border-spacing: 3px;
    table-layout: fixed;
    width:350px;
}
```

Основы CSS3

Вертикальное выравнивание ячеек

Как правило, содержимое ячеек таблицы выравнивается по центру ячейки. С помощью свойства **vertical-align** это поведение можно переопределить. Значения:

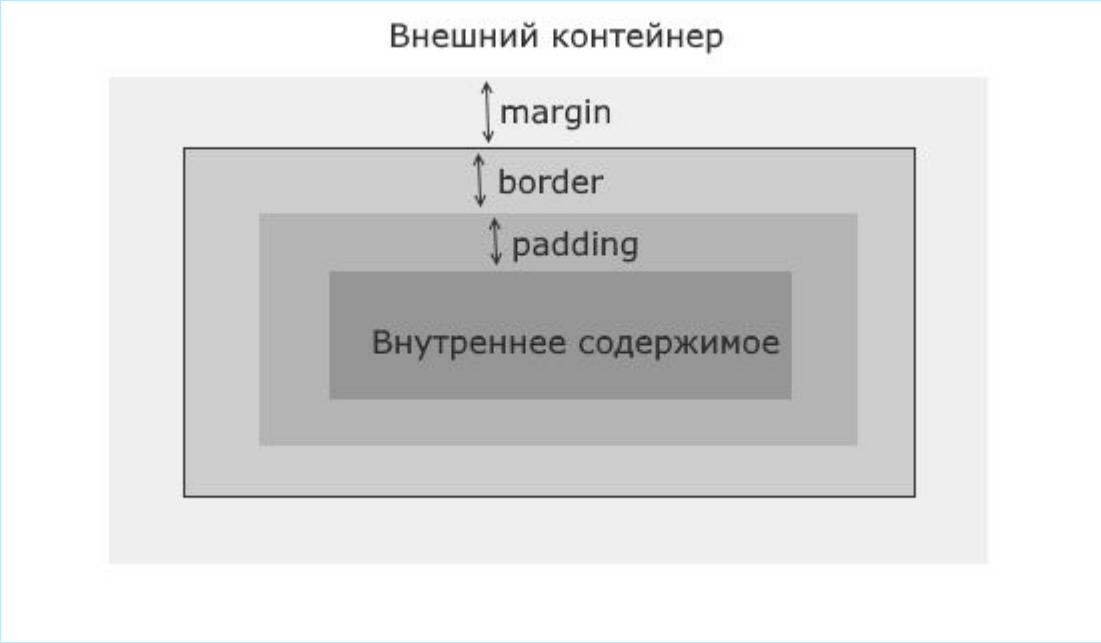
- top: выравнивание содержимого по верху ячейки
- baseline: выравнивание первой строки текста по верху ячейки
- middle: выравнивание по центру (значение по умолчанию)
- bottom: выравнивание по низу

Свойство **vertical-align** применяется только к элементам <th> и <td>:

```
td, th{
    border: solid 1px #ccc;
    vertical-align: bottom;
    height: 30px;
}
```

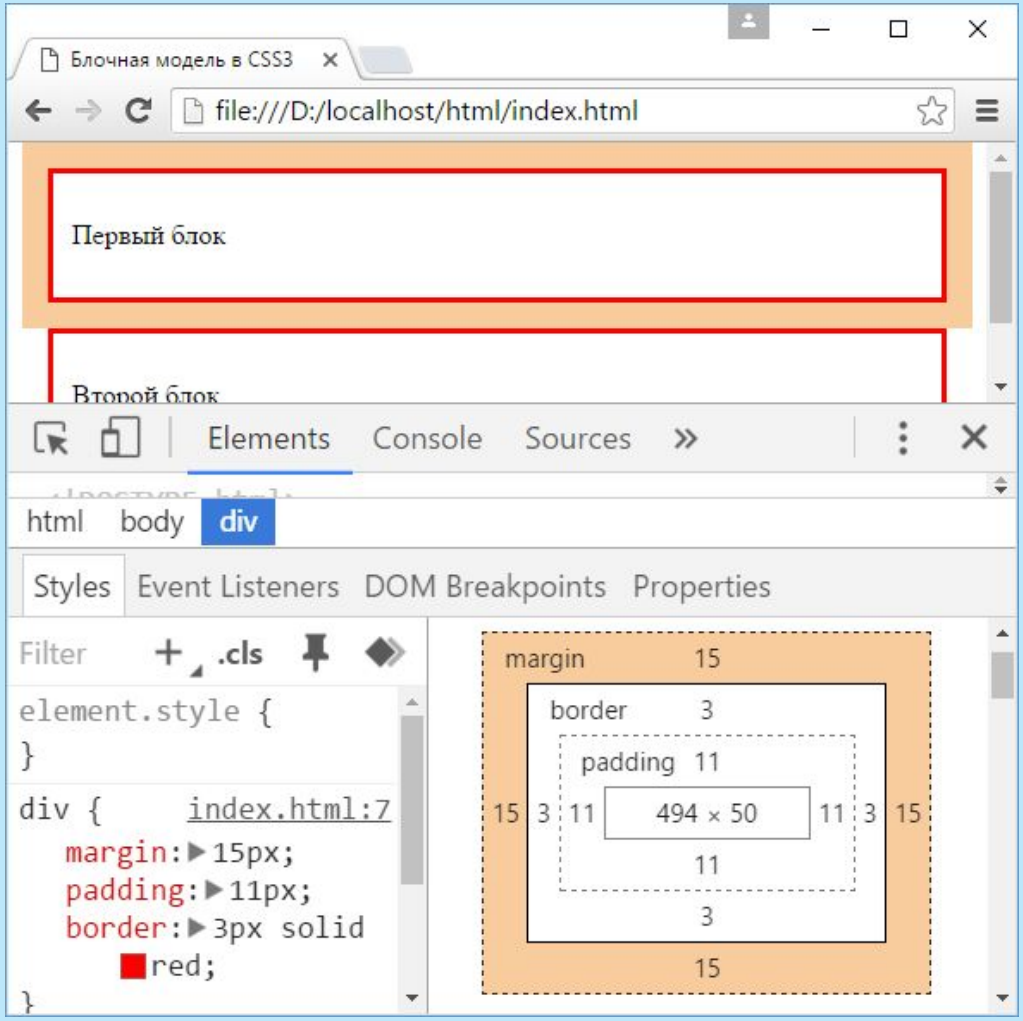
Основы CSS3

Для веб-браузера элементы страницы представляют небольшие контейнеры или блоки



Основы CSS3

```
<title>Блочная модель в CSS3</title>
 <style>
    div{
       margin: 15px; /* внешний отступ */
       padding: 11px; /* внутренний отступ */
       border: 3px solid red; /*
 границы шириной в 3 пикселя
 сплошной красной линией */
    }
</style>
   </head>
   <body>
      <div>
         <p>Первый блок</p>
      </div>
      <div>
         <p>Второй блок</p>
      </div>
   </body>
</html>
```



Основы CSS3

Свойство **margin** определяет отступ элемента от других элементов или границы внешнего контейнера. Существуют специальные свойства CSS для задания отступов для каждой стороны:

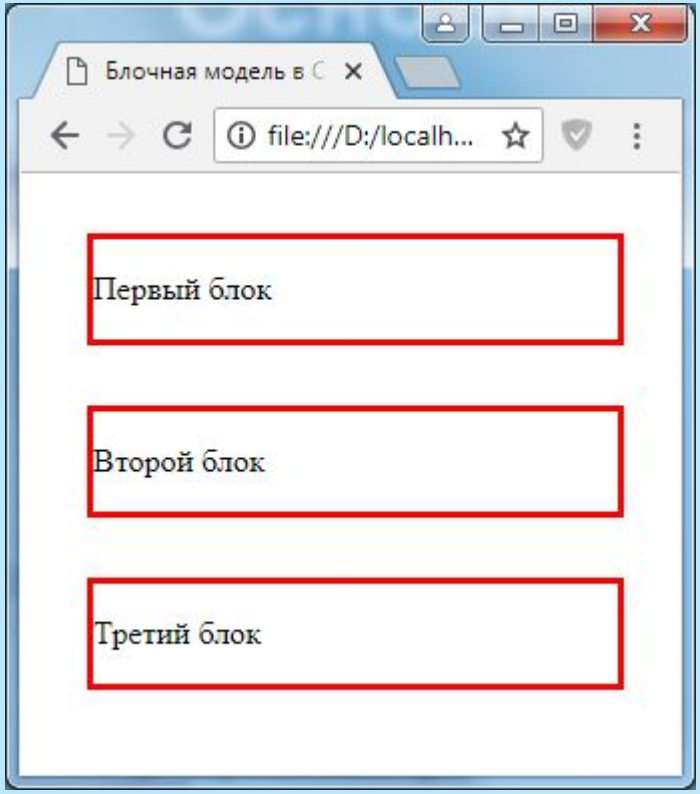
- **margin-top**: отступ сверху
- **margin-bottom**: отступ снизу
- **margin-left**: отступ слева
- **margin-right**: отступ справа

Свойство задается в формате:

margin: отступ_сверху отступ_справа отступ_снизу отступ_слева;

Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Блочная модель в CSS3</title>
      <style>
  div{
   margin: 30px 20px 15px 25px;
   border: 3px solid red;
}      </style>
   </head>
   <body>
      <div>
         <p>Первый блок</p>
      </div>
      <div>
         <p>Второй блок</p>
      </div>
      <div>
         <p>Третий блок</p>
      </div>
   </body>
</html>
```



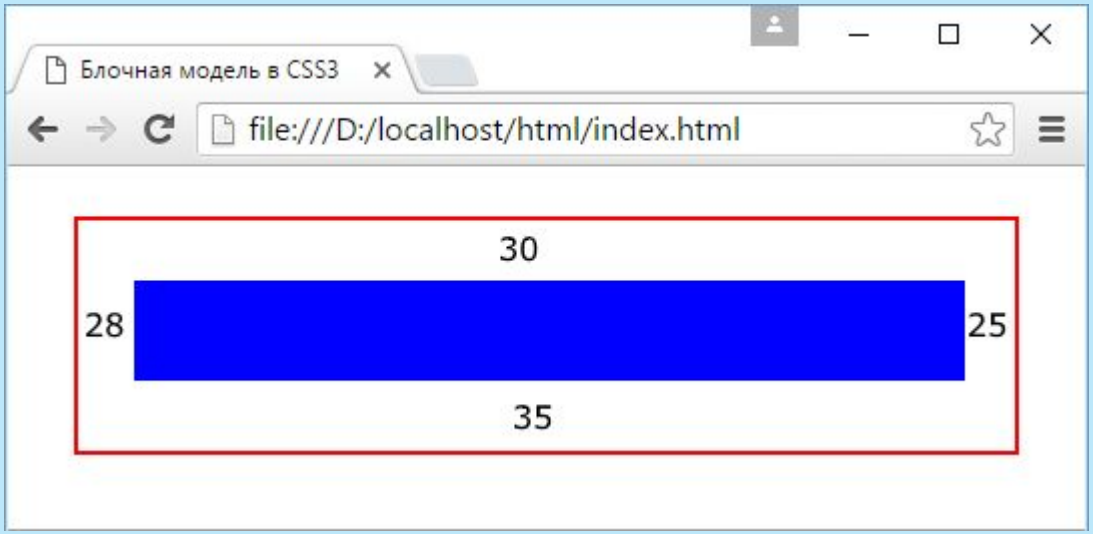
Основы CSS3

Свойство **padding** задает внутренние отступы от границы элемента до его внутреннего содержимого. Как и для свойство margin, в CSS имеются четыре свойства, которые устанавливают отступы для каждой из сторон:

- **padding-top**: отступ сверху
- **padding-bottom**: отступ снизу
- **padding-left**: отступ слева
- **padding-right**: отступ справа

Основы CSS3

```
<title>Блочная модель в CSS3</title>
    <style>
      div.outer{
        margin: 25px;
        padding-top:30px;
        padding-right: 25px;
        padding-bottom: 35px;
        padding-left: 28px;
        border: 2px solid red; }
      div.inner{
        height: 50px;
        background-color:blue;
      }
    </style>
  </head>
  <body>
    <div class="outer">
      <div class="inner"></div>
    </div>
  </body>
```



Основы CSS3

Граница элемента

Для настройки границы могут использоваться сразу несколько свойств:

- border-width: устанавливает ширину границы
- border-style: задает стиль линии границы
- border-color: устанавливает цвет границы

Основы CSS3

Свойство border-width может принимать следующие типы значений:

Значения в единицах измерения, таких как em, px или cm

border-width: 2px;

Одно из константных значений:

- thin (тонкая граница - 1px),
- medium (средняя по ширине - 3px), t
- hick (толстая - 5px)

border-width: medium;

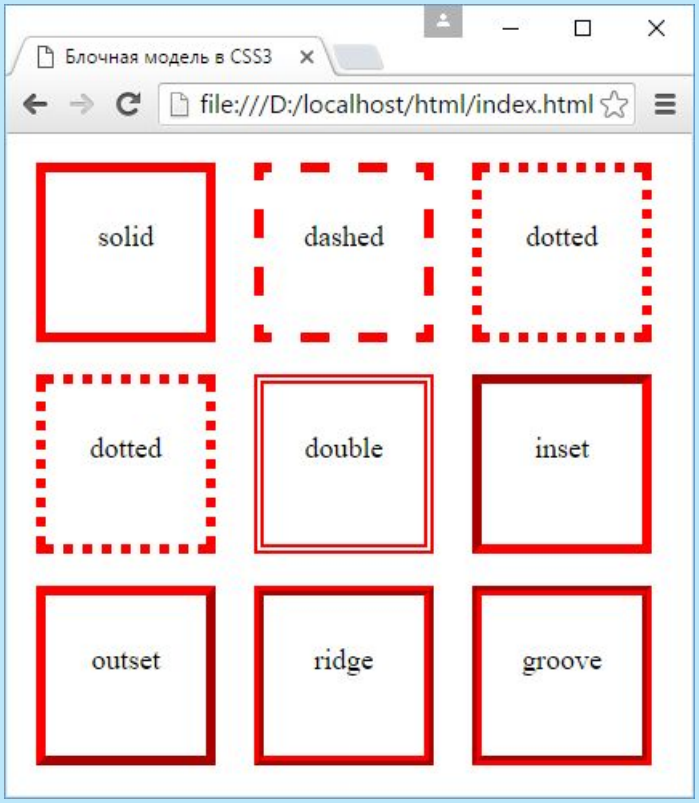
Свойство **border-color** в качестве значения принимает цвет CSS:

border-color: red;

Основы CSS3

Свойство **border-style** оформляет тип линии границы и может принимать одно из следующих значений:

- none: граница отсутствует
- solid: граница в виде обычной линии
- dashed: штриховая линия
- dotted: линия в виде последовательности точек
- double: граница в виде двух параллельных линий
- groove: граница имеет трехмерный эффект
- inset: граница как бы вдавливается во внутрь
- outset: аналогично inset, только граница как бы выступает наружу
- ridge: граница также реализует трехмерный эффект

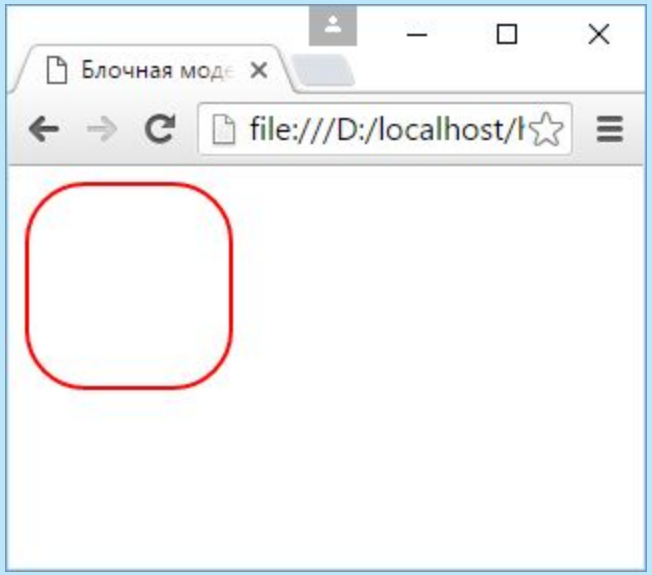


Основы CSS3

Радиус границы

Свойство **border-radius** позволяет округлить границу. Это свойство принимает значение радиуса в пикселях или единицах em.

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Блочная модель в CSS3</title>
      <style>
         div{
            width: 100px;
            height:100px;
            border: 2px solid red;
            border-radius: 30px;
         }
      </style>
   </head>
   <body>
      <div></div>
   </body>
</html>
```



Основы CSS3

Можно указать четыре значения для установки радиуса у каждого углов:

border-radius: 15px 30px 5px 40px;

Вместо общей установки радиусов для всех углов, можно их устанавливать по отдельности:

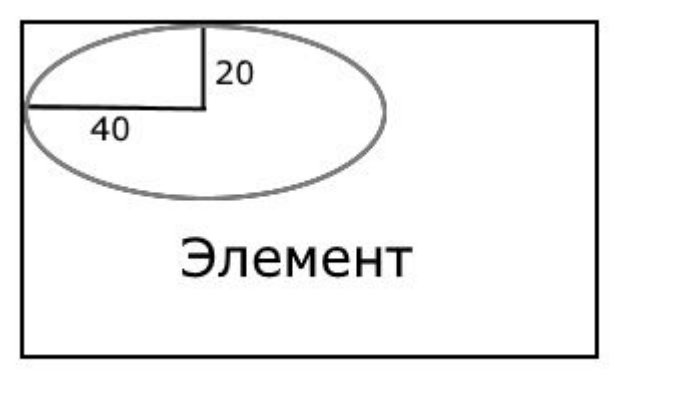
- border-top-left-radius: 15px; /* радиус для верхнего левого угла */
- border-top-right-radius: 30px; /* радиус для верхнего правого угла */
- border-bottom-right-radius: 5px; /* радиус для нижнего левого угла */
- border-bottom-left-radius: 40px; /* радиус для нижнего правого угла */

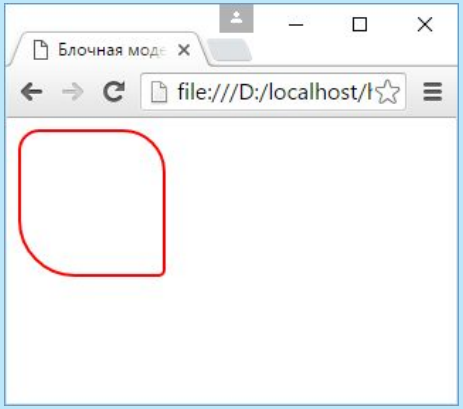
Основы CSS3

Также **border-radius** поддерживает возможность создания эллиптических углов. То есть угол не просто скругляется, а использует два радиуса, образуя в итоге душу эллипса:

border-radius: 40px/20px;

В данном случае полагается, что радиус по оси X будет иметь значение 40 пикселей, а по оси Y - 20 пикселей.





Основы CSS3

Размеры элементов задаются с помощью свойств **width** (ширина) и **height** (высота).

Значение по умолчанию для этих свойств - auto, то есть браузер сам определяет ширину и высоту элемента.

Можно также явно задать размеры с помощью единиц измерения (пикселей, em) или с помощью процентов:

width: 150px;

width: 75%;

height: 15em;

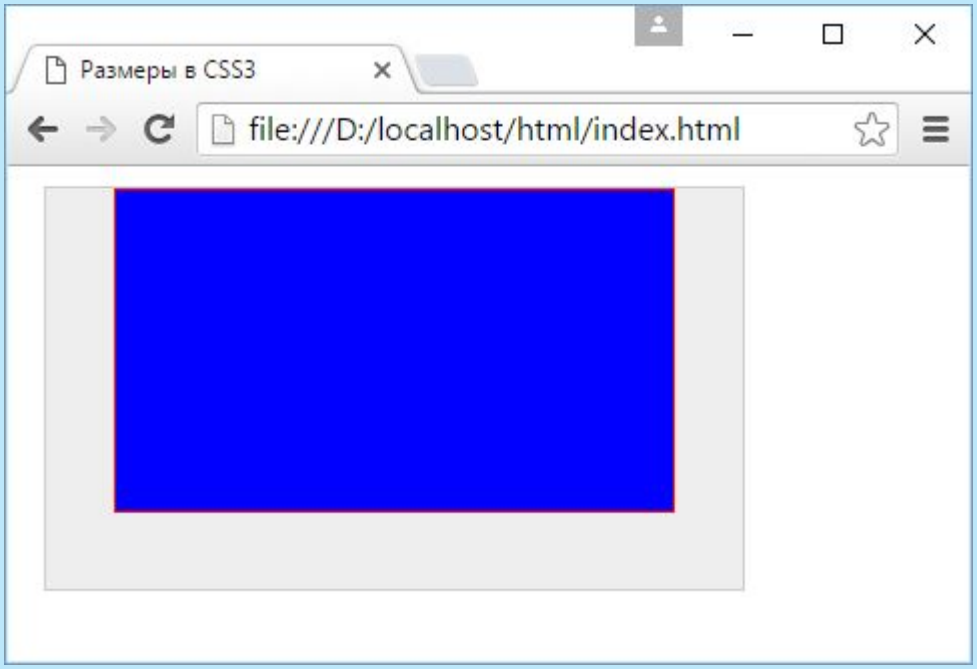
Основы CSS3

С помощью дополнительного набора свойств можно установить минимальные и максимальные размеры:

- **min-width**: минимальная ширина
- **max-width**: максимальная ширина
- **min-height**: минимальная высота
- **max-height**: максимальная высота

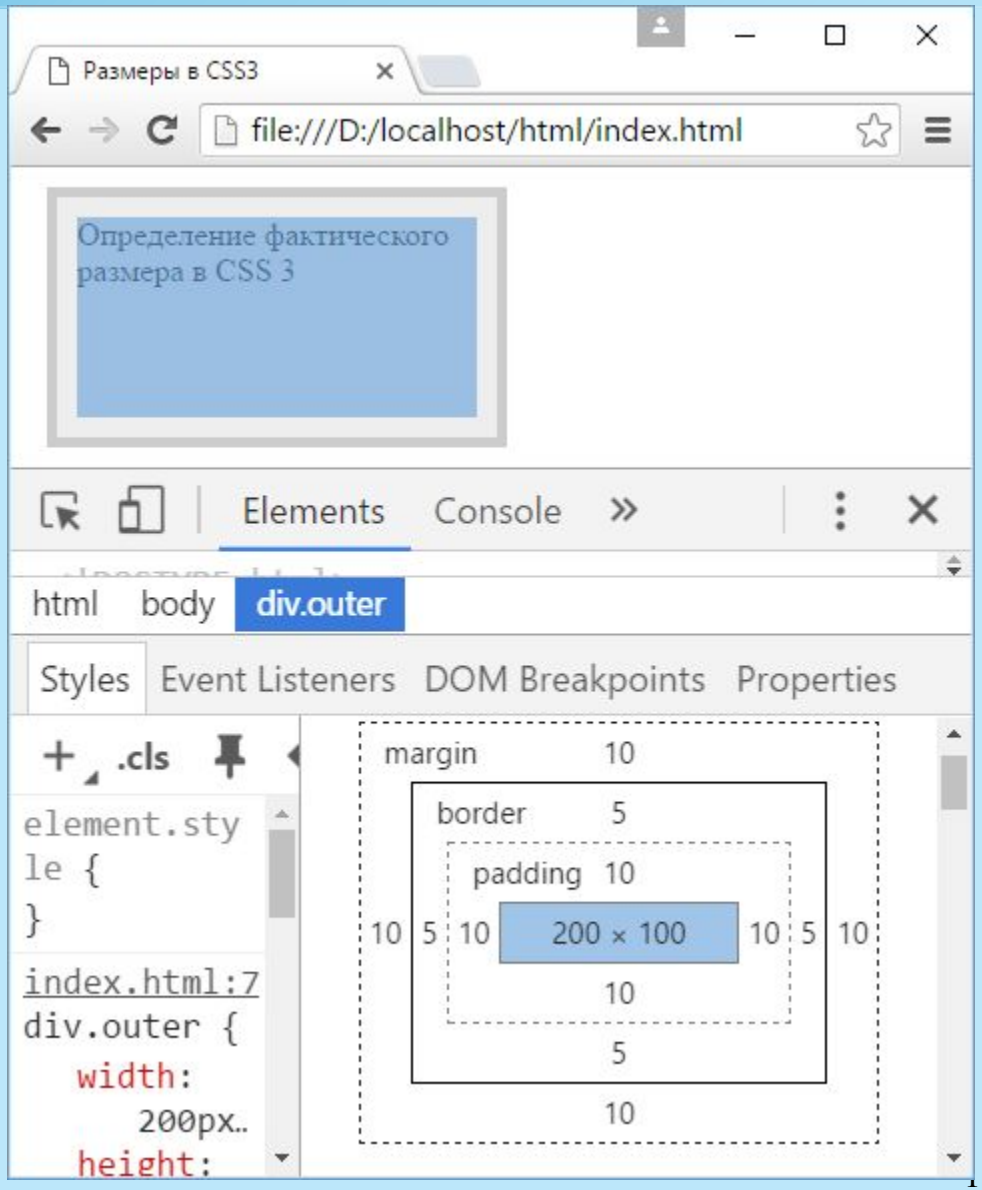
Основы CSS3

```
<style>
    div.outer{
        width: 75%;
        height: 200px;
        margin: 10px;
        border: 1px solid #ccc;
        background-color: #eee;
    }
    div.inner{
        width: 80%;
        height: 80%;
        margin: auto;
        border: 1px solid red;
        background-color: blue;
    }
</style>
```



Основы CSS3

```
<style>
    div.outer{
        width: 200px;
        height: 100px;
        margin: 10px;
        padding: 10px;
        border: 5px solid #ccc;
        background-color: #eee;
    }
</style>
```



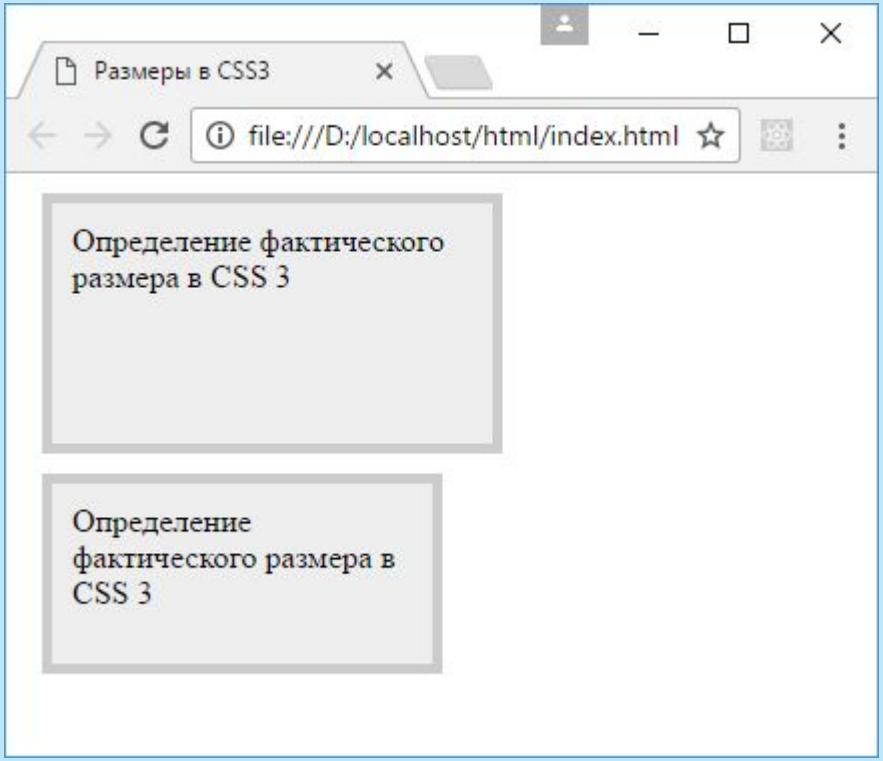
Основы CSS3

Свойство **box-sizing** позволяет переопределить установленные размеры элементов. Оно может принимать одно из следующих значений:

- content-box: значение свойства по умолчанию, при котором браузер для определения реальных ширины и высоты элементов добавляет к значениям свойств width и height ширину границы и внутренние отступы
- padding-box: указывает веб-браузеру, что ширина и высота элемента должны включать внутренние отступы как часть своего значения.

Основы CSS3

```
<style>
    div{
        width: 200px;
        height: 100px;
        margin: 10px;
        padding: 10px;
        border: 5px solid #ccc;
        background-color: #eee;
    }
    div.outer1{
        box-sizing: content-box;
    }
    div.outer2{
        box-sizing: border-box;
    }
    </style>
```



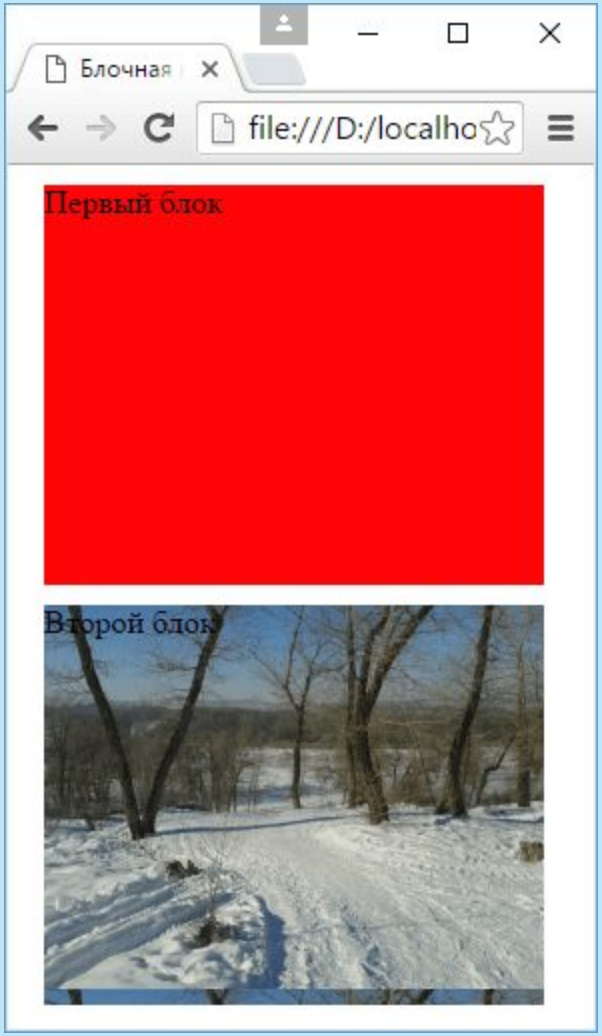
Основы CSS3

Фон элемента описывается в CSS свойством **background**.

- background-color: устанавливает цвет фона
- background-image: в качестве фона устанавливается изображение
- background-repeat: устанавливает режим повторения фонового изображения по всей поверхности элемента
- background-size: устанавливает размер фонового изображения
- background-position: указывает позицию фонового изображения
- background-attachment: устанавливает стиль прикрепления фонового изображения к элементу
- background-clip: определяет область, которая вырезается из изображения и используется в качестве фона
- background-origin: устанавливает начальную позицию фонового изображения

Основы CSS3

```
<style>
    div{
        width: 250px;
        height: 200px;
        margin: 10px;
    }
    .colored{
        background-color: #ff0507;
    }
    .imaged{
        background-image: url(forest.png);
    }
</style>
```



Основы CSS3

С помощью свойства **background-repeat** можно изменить механизм повторения. Оно может принимать следующие значения:

- repeat-x: повторение по горизонтали
- repeat-y: повторение по вертикали
- repeat: повторение по обеим сторонам (действие по умолчанию)
- space: изображение повторяется для заполнения всей поверхности элемента, но без создания фрагментов
- round: изображение должным образом масштабируется для полного заполнения всего пространства
- no-repeat: изображение не повторяется

Основы CSS3

Свойство **background-size** позволяет установить размер фонового изображения.

Для установки размера можно использовать либо единицы измерения, например, пиксели, либо проценты, либо одно из предустановленных значений:

- contain: масштабирует изображение по наибольшей стороне, сохраняя аспектное отношение
- cover: масштабирует изображение по наименьшей стороне, сохраняя аспектное отношение
- auto: значение по умолчанию, изображение отображается в полный размер

Основы CSS3

Свойство **background-position** управляет позицией фонового изображения внутри элемента.
background-position: отступ_по_оси_X отступ_по_оси_Y;

Данное свойство может принимать одно из следующих значений:

- top: выравнивание по верхнему краю элемента
- left: выравнивание по левому краю элемента
- right: выравнивание по правому краю элемента
- bottom: выравнивание по нижнему краю элемента
- center: изображение располагается по центру элемента

Например: background-position: top right;

Основы CSS3

Свойство **box-shadow** позволяет создать у элемента тень:

box-shadow: hoffset voffset blur spread color inset

- hoffset: горизонтальное смещение тени относительно элемента. При положительном значении тень смещается вправо, а при отрицательном - влево
- voffset: вертикальное смещение тени относительно элемента. При положительном значении тень смещается вниз, а при отрицательном - вверх
- blur: необязательное значение, которое определяет радиус размытия тени. Чем больше это значение, тем более размытыми будут края тени. По умолчанию имеет значение 0.

Основы CSS3

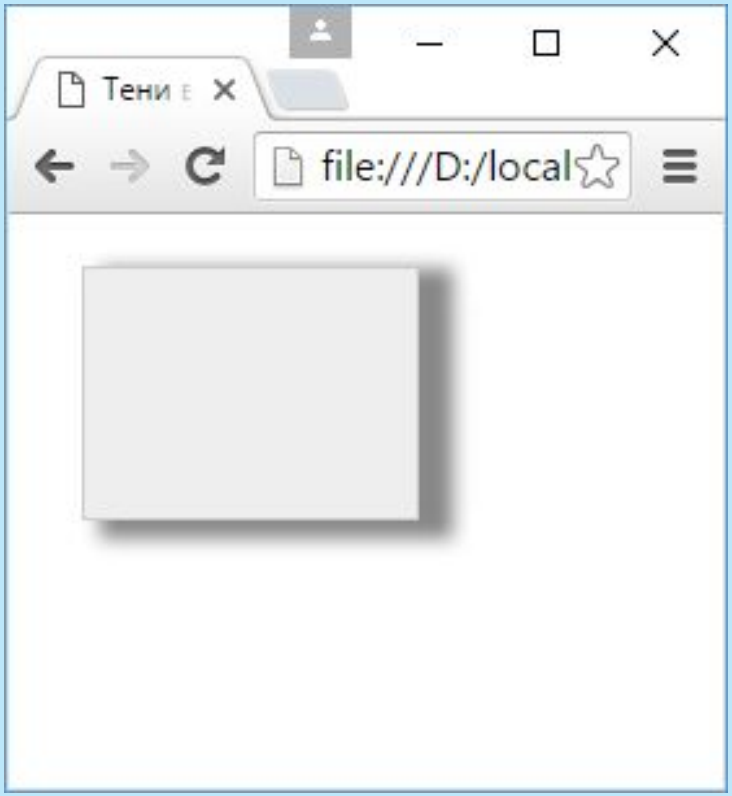
Свойство **box-shadow**

box-shadow: hoffset voffset blur spread color inset

- spread: необязательное значение, которое определяет направление тени. Положительное значение распространяет тень во вне во всех направлениях от элемента, а отрицательное значение направляет тень к элементу
- color: необязательное значение, которое устанавливает цвет тени
- inset: необязательное значение, которое заставляет рисовать тень внутри блока элемента

Основы CSS3

```
<style>
    div{
        width: 128px;
        height: 96px;
        margin: 20px;
        border: 1px solid #ccc;
        background-color: #eee;
        box-shadow: 10px 4px 10px 3px #888;
    }
</style>
```



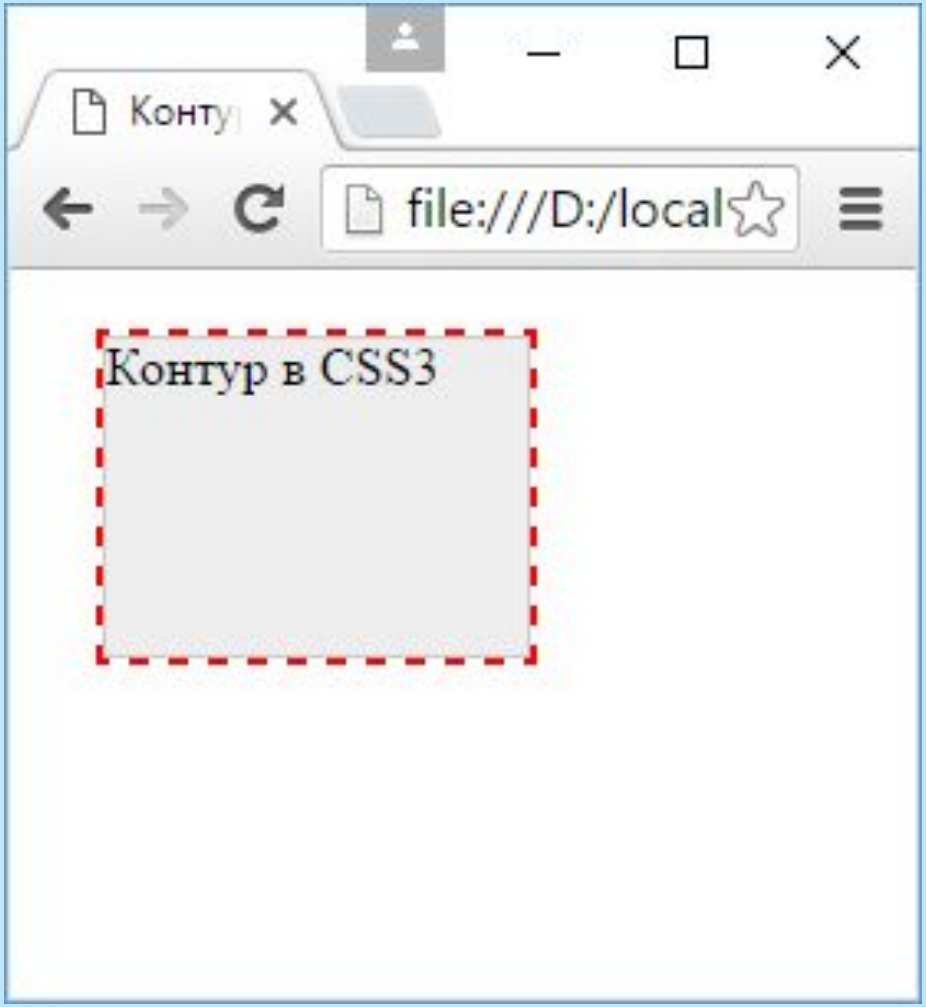
Основы CSS3

Контур в CSS 3 представлен свойством **outline**:

- outline-color: цвет контура
- outline-offset: смещение контура
- outline-style: стиль контура. Оно принимает те же значения, что и border-style:
- none: контур отсутствует
- solid: контур в виде обычной линии
- dashed: штриховая линия
- dotted: линия в виде последовательности точек
- double: контур в виде двух параллельных линий
- outline-width: толщина контура

Основы CSS3

```
<style>
    div{
        width: 128px;
        height: 96px;
        margin: 20px;
        border: 1px solid #ccc;
        background-color: #eee;
        outline-color: red;
        outline-style: dashed;
        outline-width: 2px;
    }
</style>
```



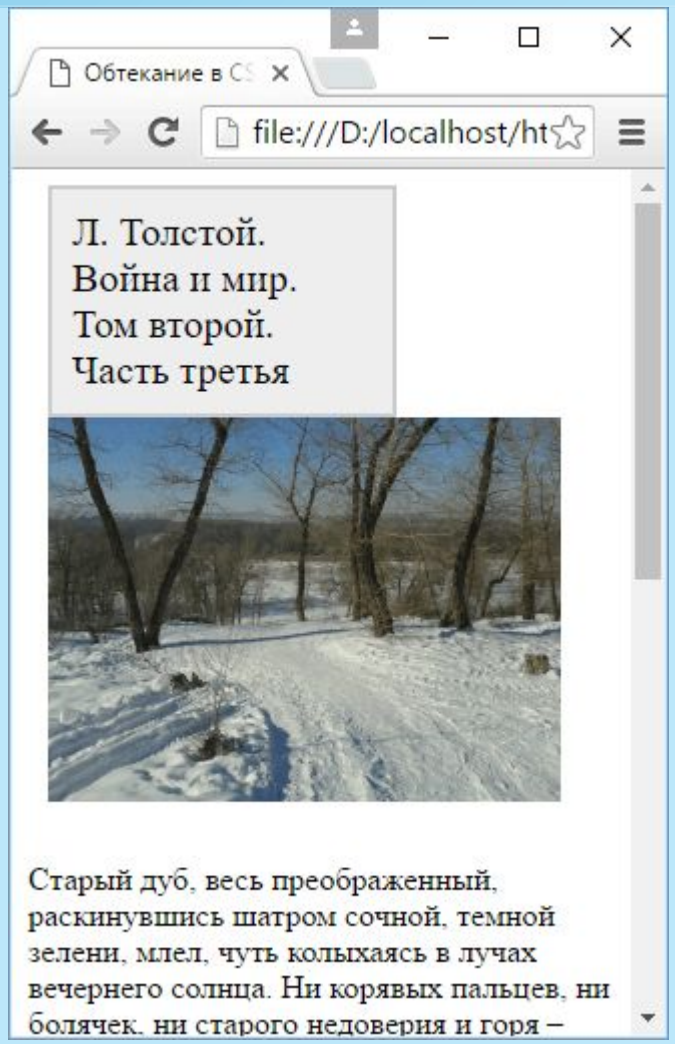
Основы CSS3

Свойство **float** позволяет установить обтекание элементов.

- left: элемент перемещается влево, а все содержимое, которое идет ниже его, обтекает правый край элемента
- right: элемент перемещается вправо
- none: отменяет обтекание и возвращает объект в его обычную позицию

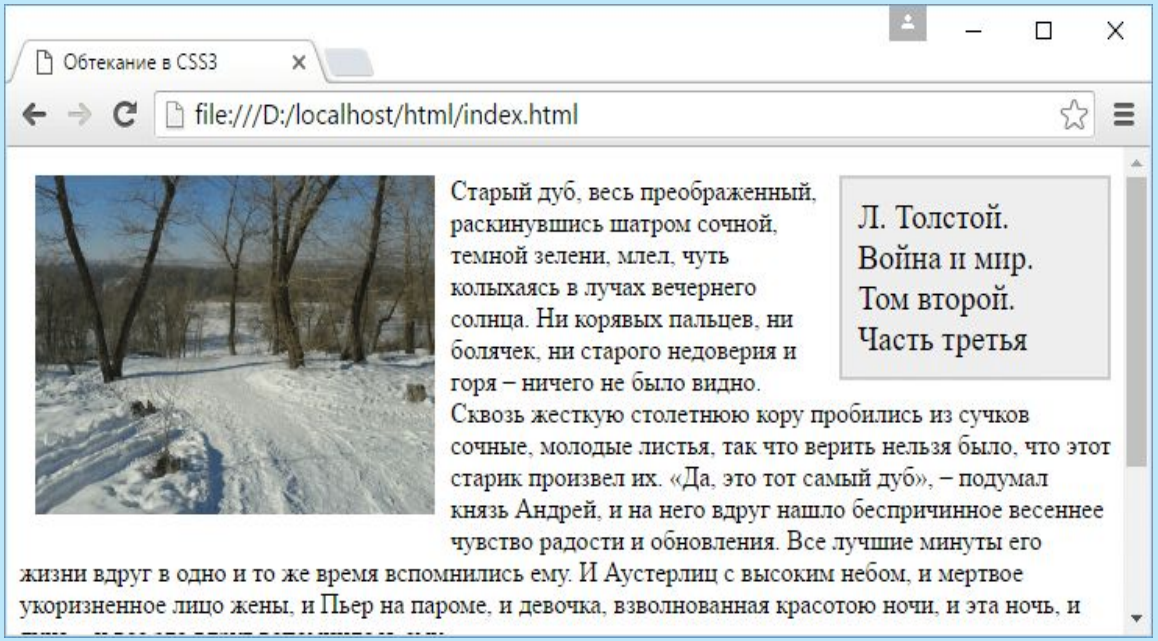
Основы CSS3

```
<style>
    .image {
        margin:10px;
        margin-top:0px;
    }
    .sidebar{
        border: 2px solid #ccc;
        background-color: #eee;
        width: 150px;
        padding: 10px;
        margin-left:10px;
        font-size: 20px;
    }
</style>
```



Основы CSS3

```
.image {
    float:left;
    margin:10px;
    margin-top:0px;
}
.sidebar{
    border: 2px solid #ccc;
    background-color: #eee;
    width: 150px;
    padding: 10px;
    margin-left:10px;
    font-size: 20px;
    float: right;
}
```



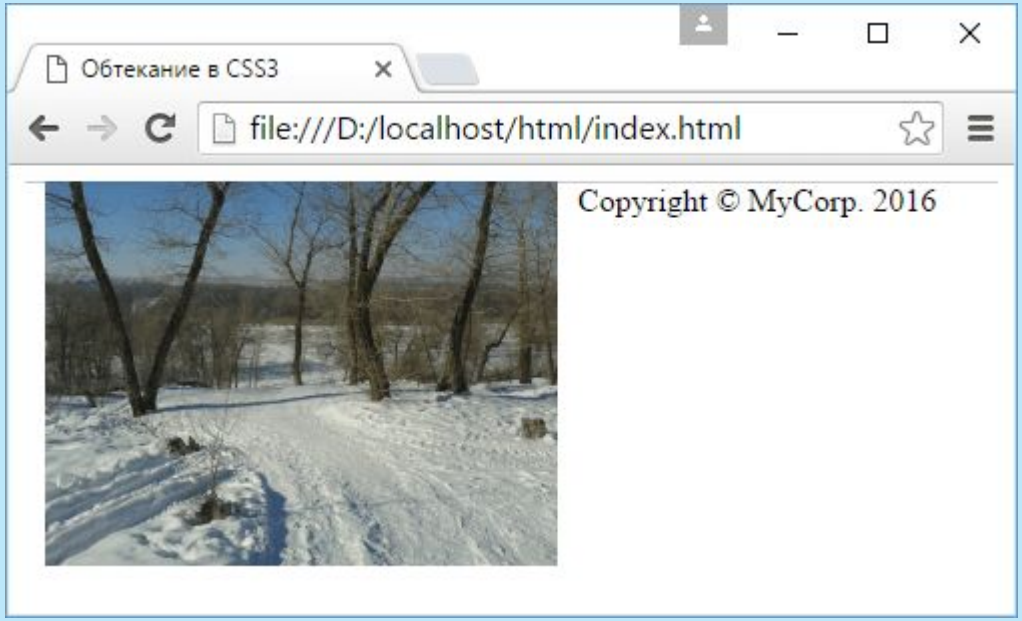
Основы CSS3

Запрет обтекания. Свойство **clear**:

- left: стилизуемый элемент может обтекать плавающий элемент справа. Слева же обтекание не работает
- right: стилизуемый элемент может обтекать плавающий элемент только слева. А справа обтекание не работает
- both: стилизуемый элемент может обтекать плавающие элементы и относительно них смещается вниз
- none: стилизуемый элемент ведет себя стандартным образом, то есть принимает участие в обтекании справа и слева

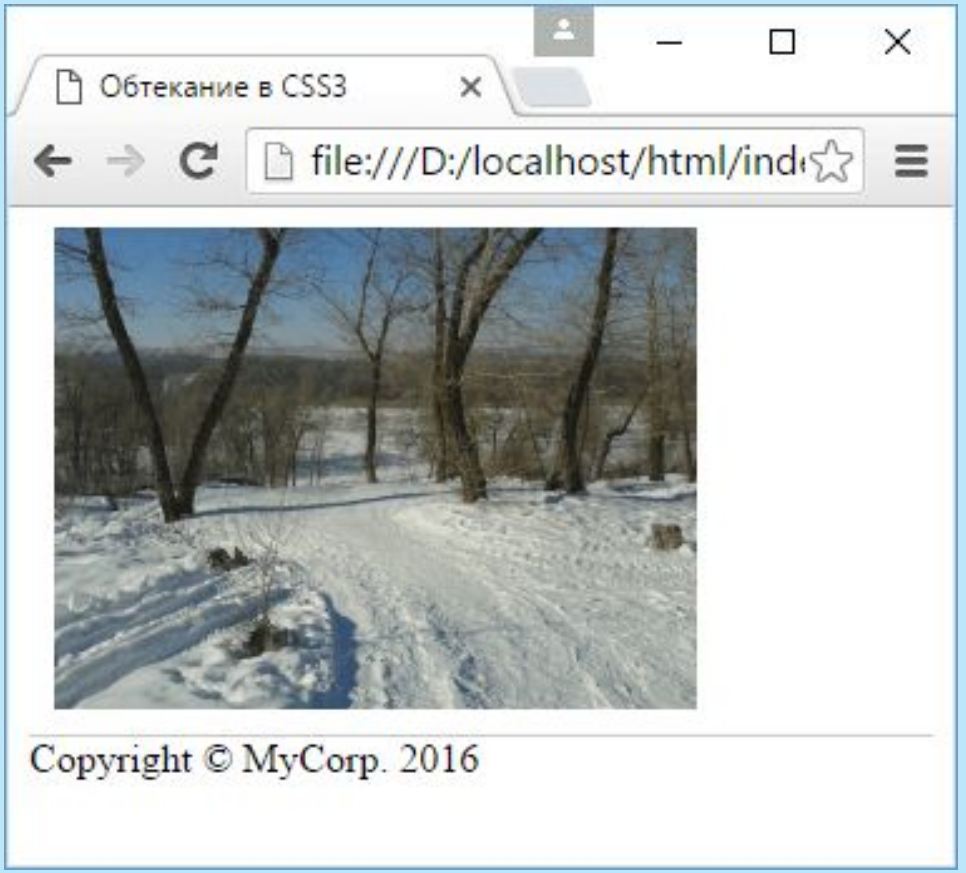
Основы CSS3

```
<style>
        .image {
            float:left;
            margin:10px;
            margin-top:0px;
        }
        .footer{
            border-top: 1px solid #ccc;
        }
</style>
```



Основы CSS3

```
.footer{
    border-top: 1px solid #ccc;
    clear: both;
}
```



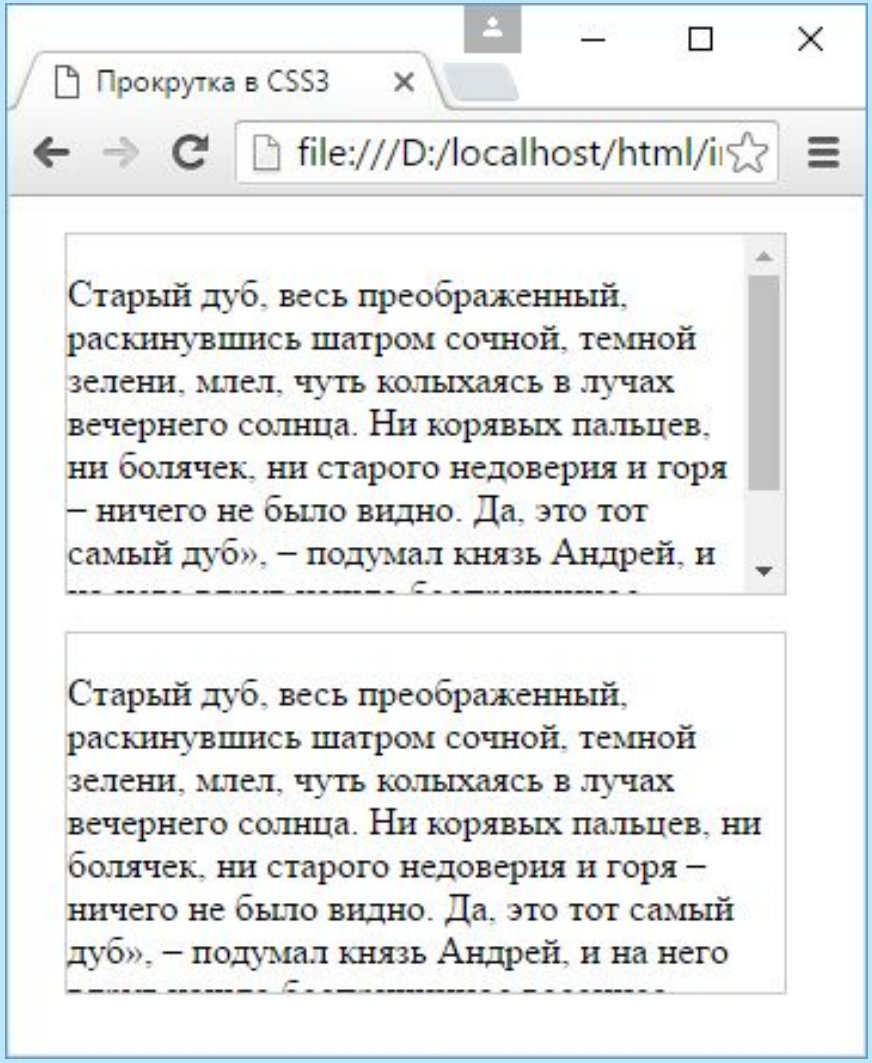
Основы CSS3

Свойство **overflow** позволяет настроить поведение блока и добавить возможность прокрутки.

- auto: если контент выходит за границы блока, то создается прокрутка. В остальных случаях полосы прокрутки не отображаются
- hidden: отображается только видимая часть контента. Контент, который выходит за границы блока, не отображается, а полосы прокрутки не создаются
- scroll: в блоке отображаются полосы прокрутки, даже если контент весь помещается в границах блока, и таких полос прокрутки не требуется
- visible: значение по умолчанию, контент отображается, даже если он выходит за границы блока

Основы CSS3

```
<style>
    .article1{
        width: 300px;
        height: 150px;
        margin:15px;
        border: 1px solid #ccc;
        overflow: auto;
    }
    .article2{
        width: 300px;
        height: 150px;
        margin:15px;
        border: 1px solid #ccc;
        overflow: hidden;
    }
</style>
```



Основы CSS3

Для создания градиента нужно указать его начало и несколько цветов, например:

background-image: linear-gradient(left,black,white);

В данном случае началом градиента будет левый край элемента, который обозначается значением left. Цвета градиента: черный (black) и белый (white). То есть начиная с левого края элемента до правого будет плавно идти переход из черного цвета в белый.

В использовании градиентов есть один недостаток - многообразие браузеров вынуждает использовать префикс вендора:

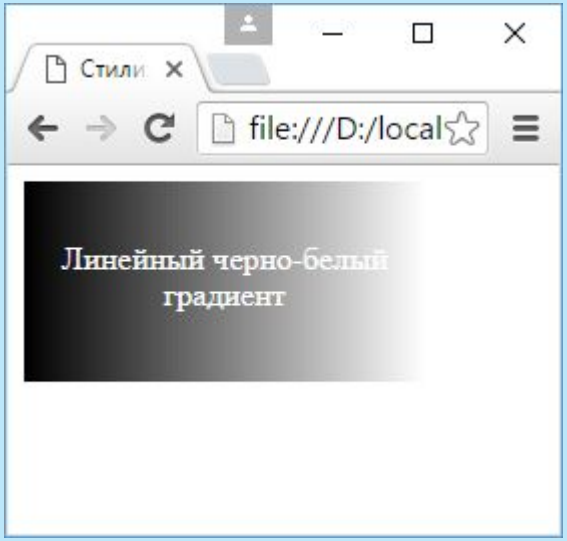
-webkit- /* Для Google Chrome, Safari, Microsoft Edge, Opera выше 15 версии */

-moz- /* Для Mozilla Firefox */

-o- /* Для Opera старше 15 версии (Opera 12) */

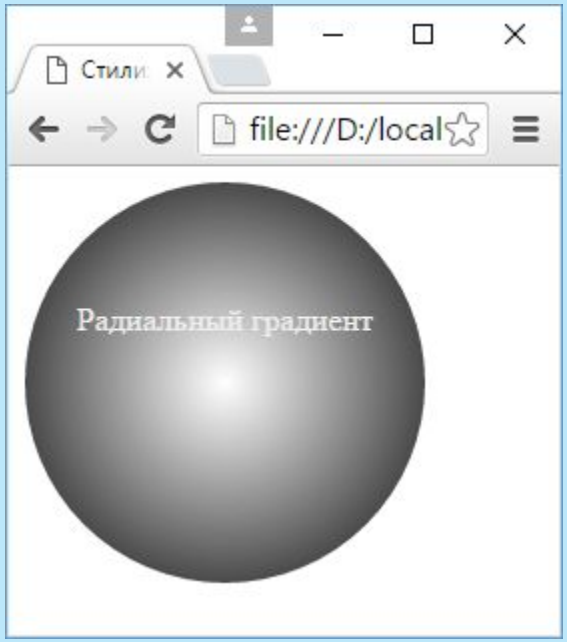
Основы CSS3

```
<style>
    div {
        width: 200px;
        height: 100px;
        background-color: #eee;
        background-image: linear-gradient(left,black,white);
        background-image: -o-linear-gradient(left,black,white);
        background-image: -moz-linear-gradient(left,black,white);
        background-image: -webkit-linear-gradient(left,black,white);
    }
    p{
        margin: 0;
        padding-top: 30px;
        text-align: center;
        color:white;
    }
</style>
```



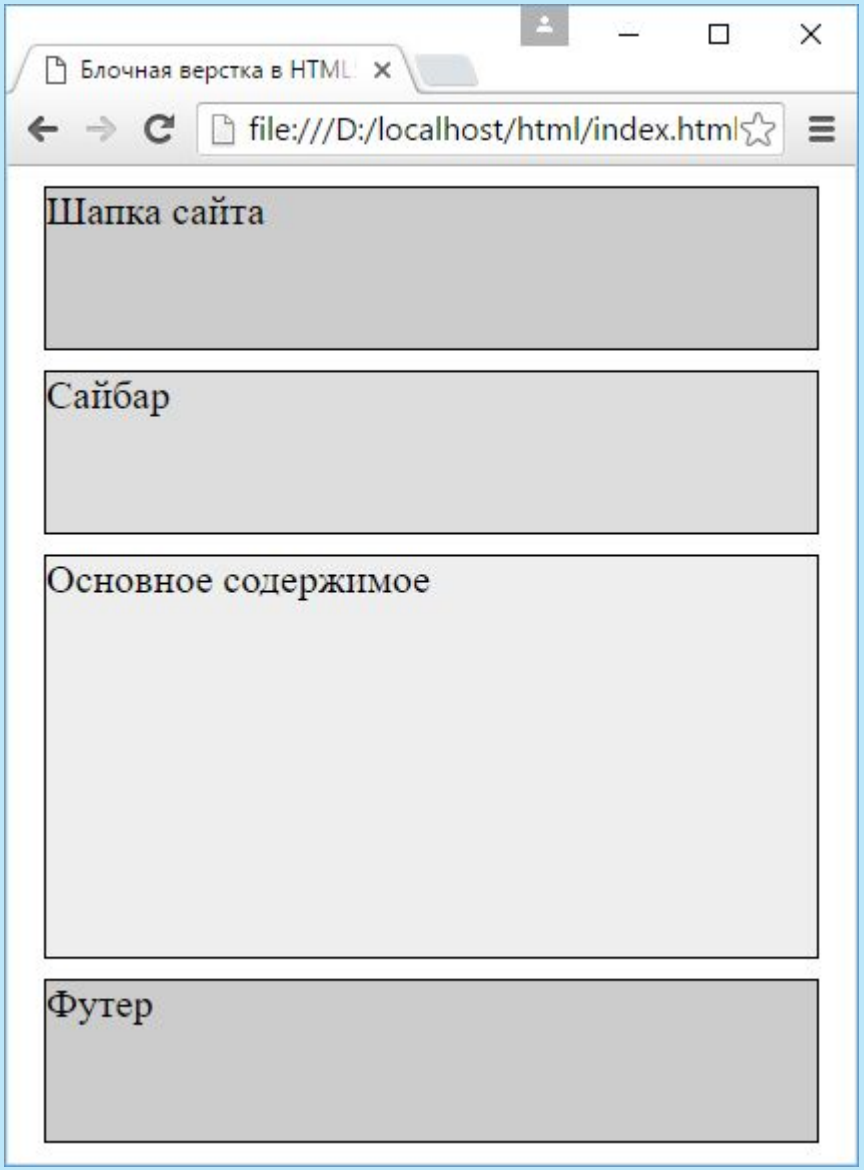
Основы CSS3

```
<style>
    div {
        width: 200px;
        height: 200px;
        border-radius: 100px;
        background-color: #eee;
        background-image: radial-gradient(white, black);
        background-image: -moz-radial-gradient(white, black);
        background-image: -webkit-radial-gradient(white, black);
    }
    p{
        margin: 0;
        padding-top: 60px;
        text-align: center;
        color: #eee;
    }
    </style>
```



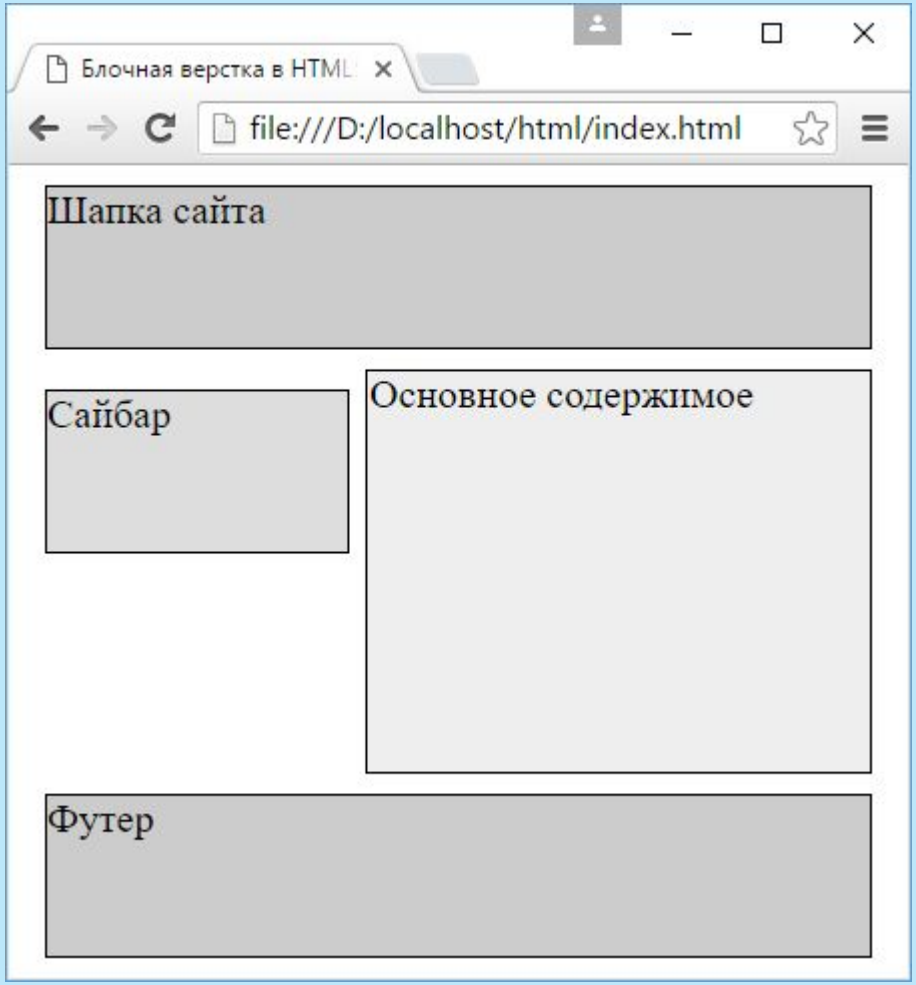
Основы CSS3

```
<style>
        div{
            margin: 10px;
            border: 1px solid black;
            font-size: 20px;
            height: 80px;
        }
        #header{
            background-color: #ccc;
        }
        #sidebar{
            background-color: #ddd;
        }
        #main{
            background-color: #eee;
            height: 200px;
        }
        #footer{
            background-color: #ccc;
        }
</style>
```



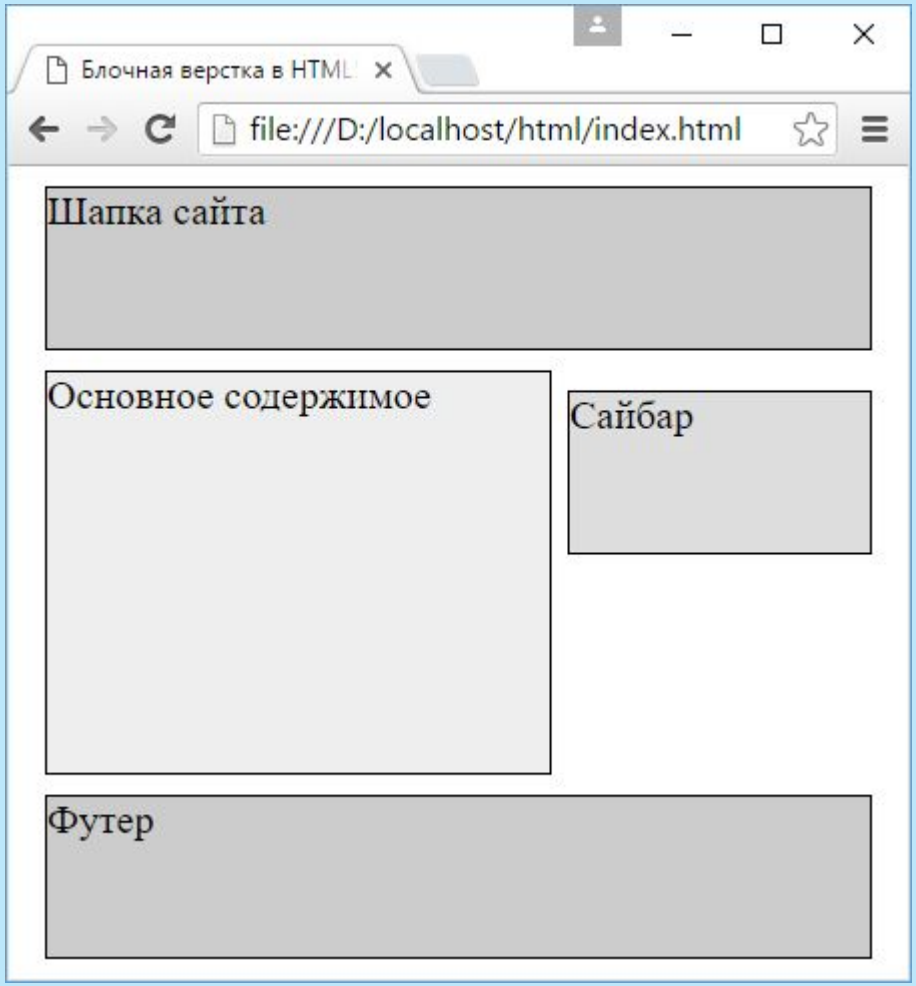
Основы CSS3

```
#sidebar{
    background-color: #ddd;
    float: left;
    width: 150px;
}
#main{
    background-color: #eee;
    height: 200px;
    margin-left: 170px;
}
```



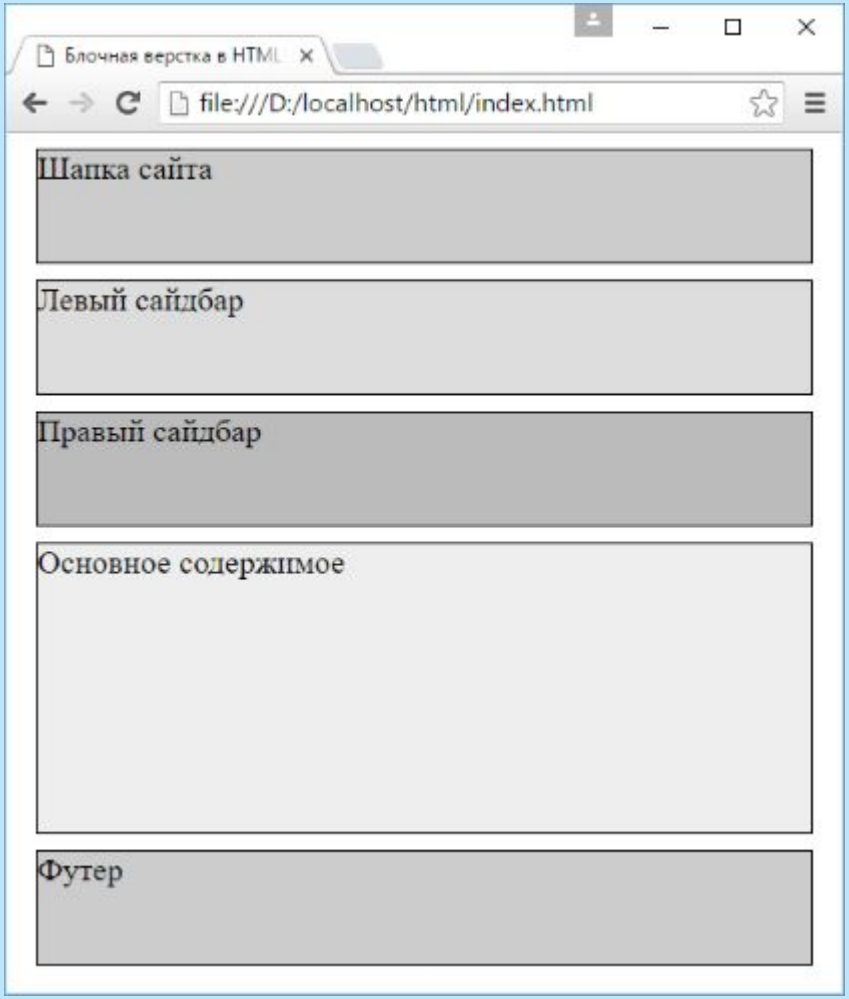
Основы CSS3

```
#sidebar{
    background-color: #ddd;
    float: right;
    width: 150px;
}
#main{
    background-color: #eee;
    height: 200px;
    margin-right: 170px;
}
```



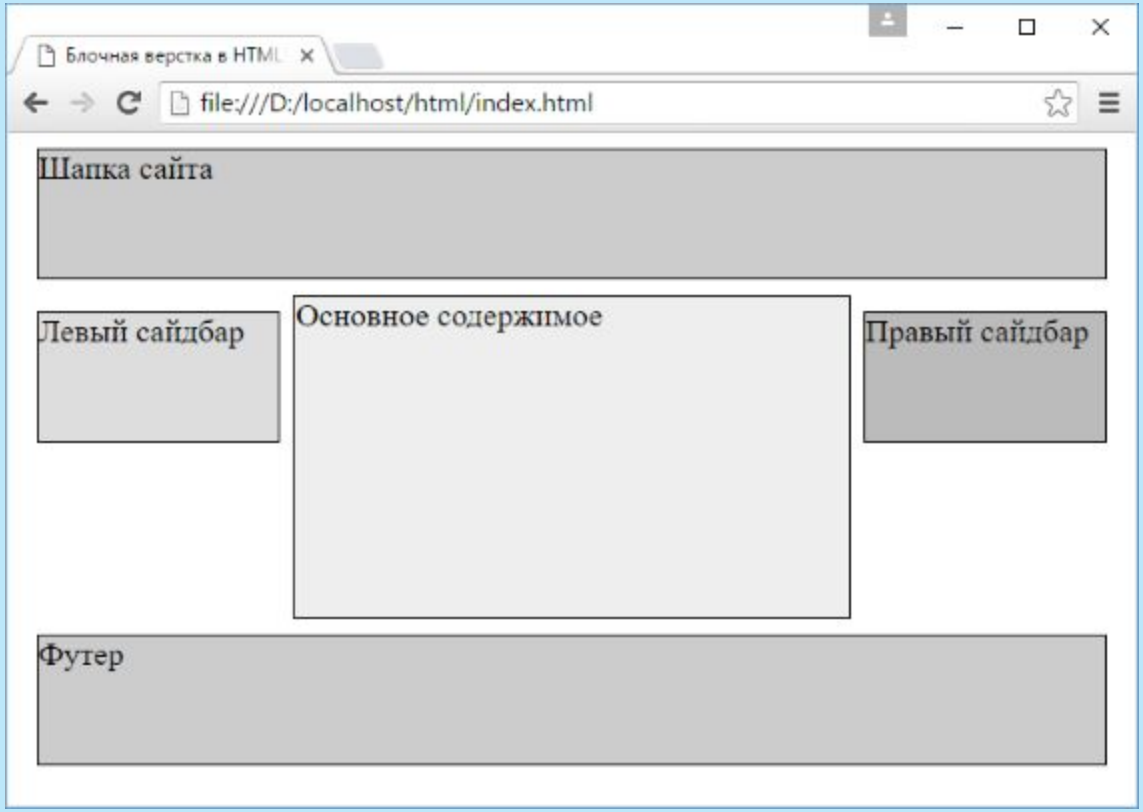
Основы CSS3

```
<style>
        div{
            margin: 10px;
            border: 1px solid black;
            font-size: 20px;
            height: 80px;
        }
        #header{
            background-color: #ccc;
        }
        #leftSidebar{
            background-color: #ddd;
        }
        #rightSidebar{
            background-color: #bbb;
        }
        #main{
            background-color: #eee;
            height: 200px;
        }
        #footer{
            background-color: #ccc;
        }
</style>
```



Основы CSS3

```
#leftSidebar{
    background-color: #ddd;
    float: left;
    width: 150px;
}
#rightSidebar{
    background-color: #bbb;
    float: right;
    width: 150px;
}
#main{
    background-color: #eee;
    height: 200px;
    margin-left: 170px;
    margin-right: 170px;
}
```



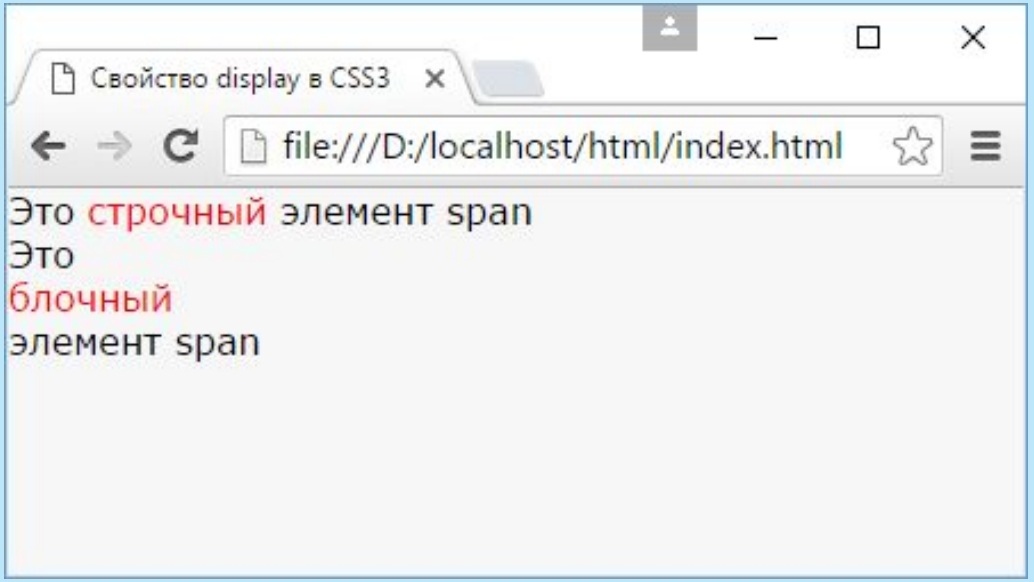
Основы CSS3

Свойство **display** позволяет управлять блоком элемента и также влиять на его позиционирование относительно соседних элементов.

- inline: элемент становится строчным, подобно словам в строке текста
- block: элемент становится блочным, как параграф
- inline-block: элемент располагается как строка текста
- list-item: элемент позиционируется как элемент списка обычно с добавление маркера виде точки или порядкового номера
- run-in: тип блока элемента зависит от окружающих элементов
- flex: позволяет осуществлять гибкое позиционирование элментов
- table, inline-table: позволяет расположить элементы в виде таблицы
- none: элемент не виден и удален из разметки html

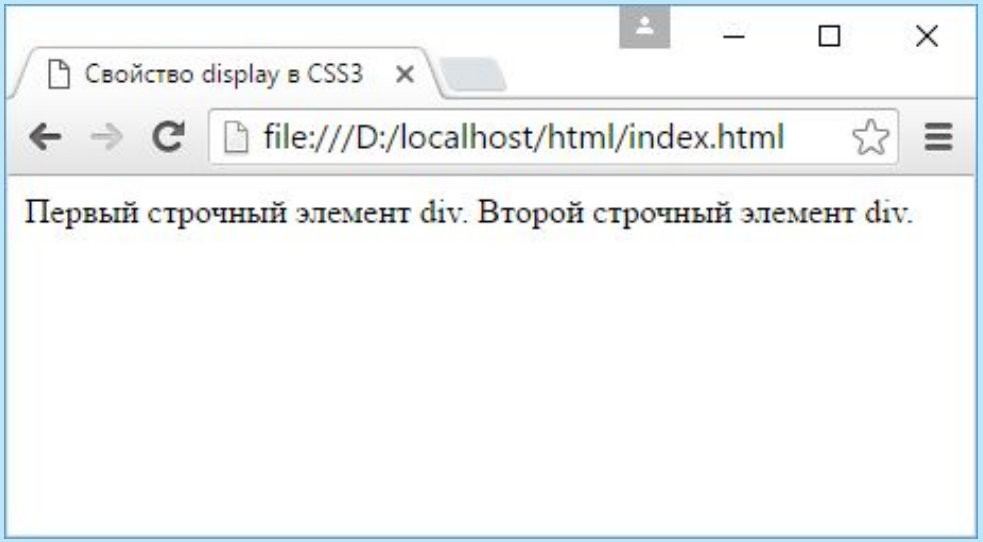
Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <link href="styles.css" rel="stylesheet">
      <title>Свойство display в CSS3</title>
      <style>
         span{
            color: red;
         }
         .blockSpan{
            display: block;
         }
      </style>
   </head>
   <body>
      <div>Это <span>строчный</span> элемент span</div>
      <div>Это <span class="blockSpan">блочный</span> элемент span</div>
   </body>
</html>
```



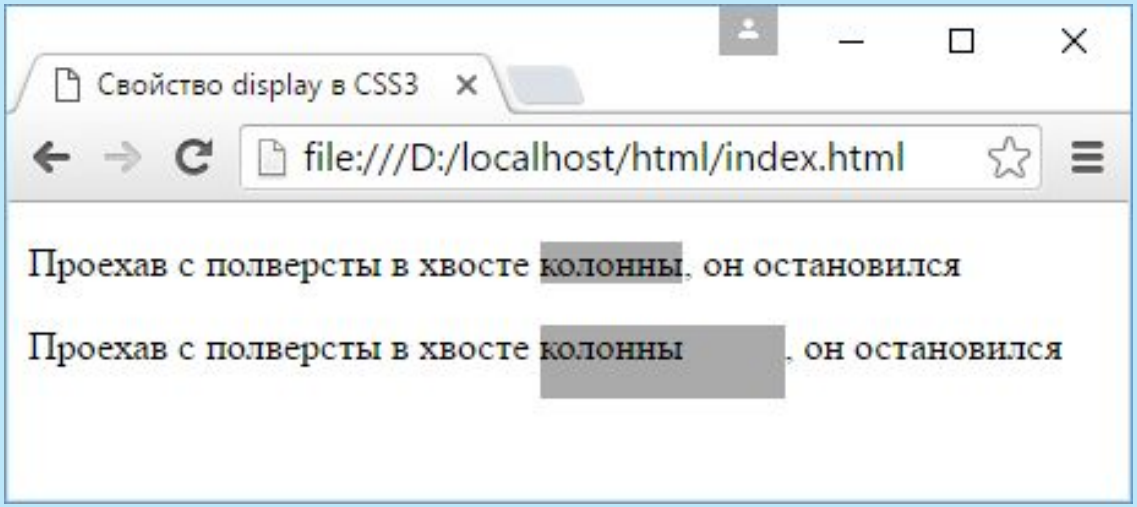
Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Свойство display в CSS3</title>
      <style>
         div{
            display: inline;
         }
      </style>
   </head>
   <body>
      <div>Первый строчный элемент div.</div>
      <div>Второй строчный элемент div.</div>
   </body>
</html>
```



Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Свойство display в CSS3</title>
      <style>
         span{
            width: 100px;
            height: 30px;
            background-color: #aaa;
         }
         .inineBlockSpan{
            display: inline-block;
         }
      </style>
   </head>
   <body>
      <p>Проехав с полверсты в хвосте <span>колонны</span>, он остановился</p>
      <p>Проехав с полверсты в хвосте <span class="inineBlockSpan">колонны</span>, он становился</p>
   </body>
</html>
```



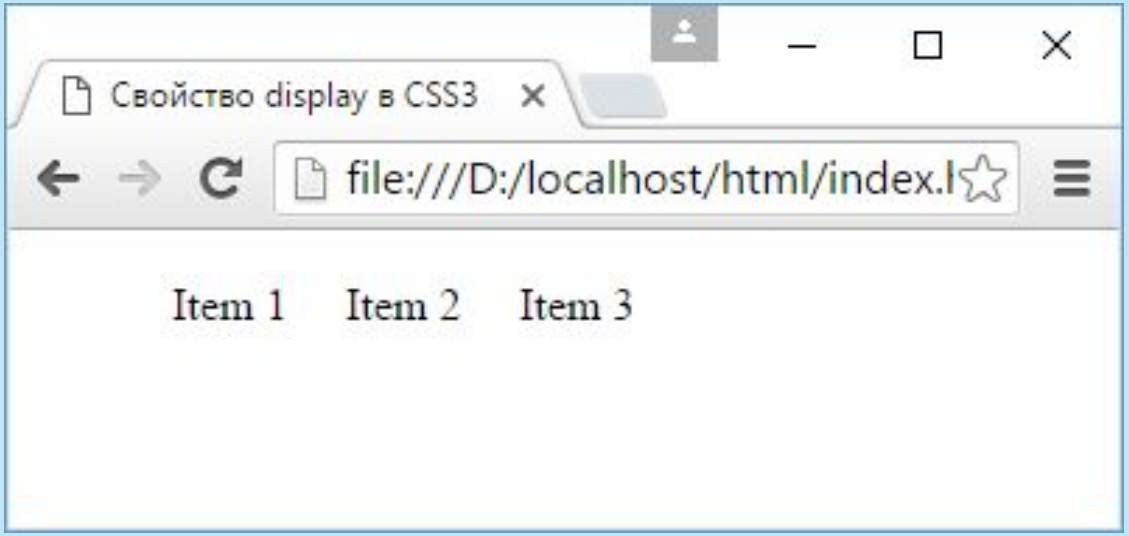
Основы CSS3

Значение **run-in** определяет элемент, который зависит от соседних элементов. И здесь есть три возможных варианта:

- Элемент окружен блочными элементами, тогда фактически он имеет стиль display: block, то есть сам становится блочным
- Элемент окружен строчными элементами, тогда фактически он имеет стиль display: inline, то есть сам становится строчным
- Во всех остальных случаях элемент считается блочным

Основы CSS3

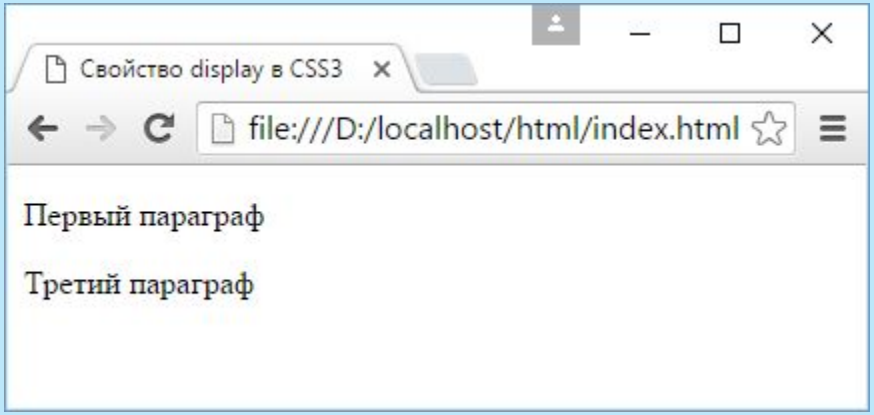
```
<style>
        ul{
            display: table;
            margin: 0;
        }
        li{
            list-style-type: none;
            display: table-cell;
            padding: 10px;
        }
</style>
<body>
      <ul>
          <li>Item 1</li>
          <li>Item 2</li>
          <li>Item 3</li>
      </ul>
</body>
```



Основы CSS3

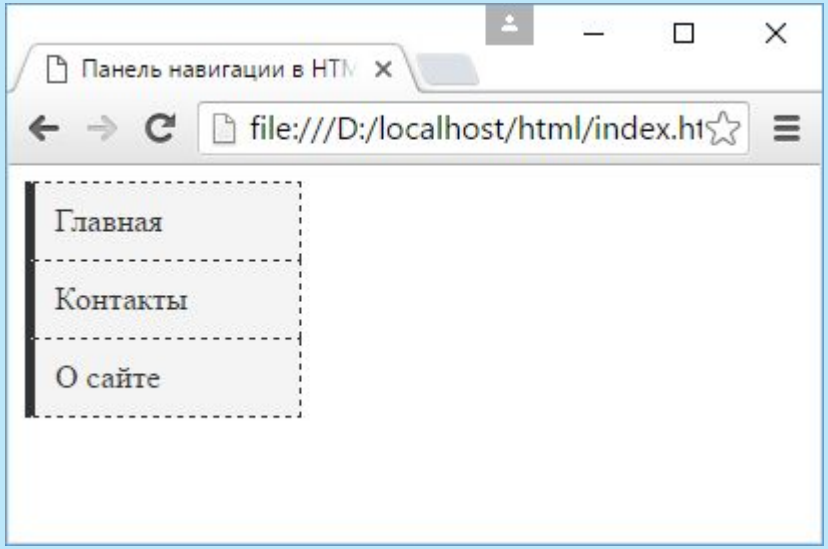
Значение **none** позволяет скрыть элемент, которого как-будто нет на веб-странице:

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Свойство display в CSS3</title>
      <style>
         .invisible{
            display: none;
         }
      </style>
   </head>
   <body>
      <p>Первый параграф</p>
      <p class="invisible">Второй параграф</p>
      <p>Третий параграф</p>
   </body>
</html>
```



Основы CSS3

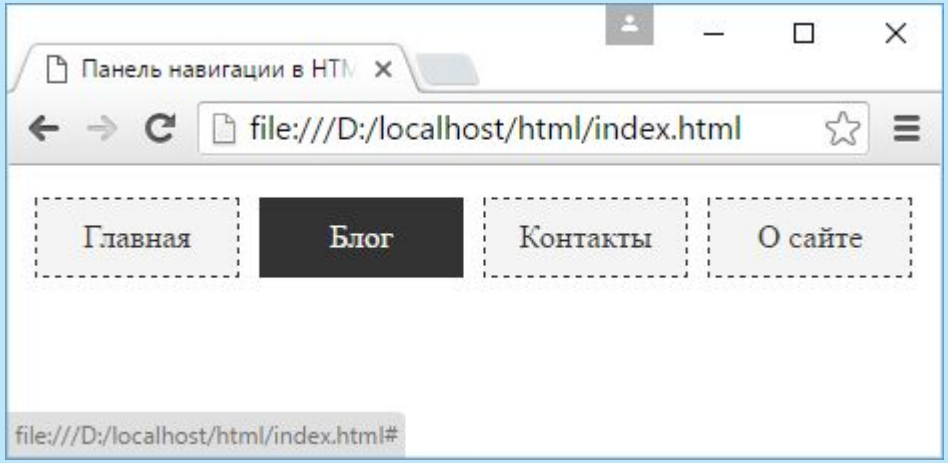
```
<style>
        ul.nav{
            margin-left: 0px;
            padding-left: 0px;
            list-style: none;
        }
        ul.nav a {
            display: block;
            width: 7em;
            padding:10px;
            background-color: #f4f4f4;
            border-top: 1px dashed #333;
            border-right: 1px dashed #333;
            border-left: 5px solid #333;
            text-decoration: none;
            color: #333;
        }
        ul.nav li:last-child a {
            border-bottom: 1px dashed #333;
        }
</style>
```




```
<body>
    <ul class="nav">
        <li><a href="#">Главная</a></li>
        <li><a href="#">Контакты</a></li>
        <li><a href="#">О сайте</a></li>
    </ul>
</body>
```

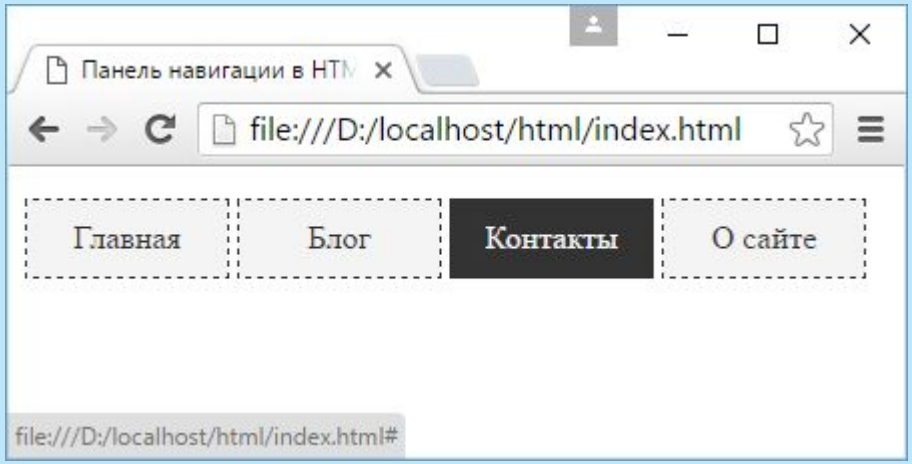
Основы CSS3

```
<style>
        ul.nav{
            margin-left: 0px;
            padding-left: 0px;
            list-style: none;
        }
        .nav li {
            float: left;
        }
        ul.nav a {
            display: block;
            width: 5em;
            padding:10px;
            margin: 0 5px;
            background-color: #f4f4f4;
            border: 1px dashed #333;
            text-decoration: none;
            color: #333;
            text-align: center;
        }
        ul.nav a:hover{
            background-color: #333;
            color: #f4f4f4;
        }
 </style>
```



Основы CSS3

```
<style>
        ul.nav{
            margin-left: 0px;
            padding-left: 0px;
            list-style: none;
        }
        .nav li {
            display: inline;
        }
        ul.nav a {
            display: inline-block;
            width: 5em;
            padding:10px;
            background-color: #f4f4f4;
            border: 1px dashed #333;
            text-decoration: none;
            color: #333;
            text-align: center;
        }
        ul.nav a:hover{
            background-color: #333;
            color: #f4f4f4;
        }
    </style>
```



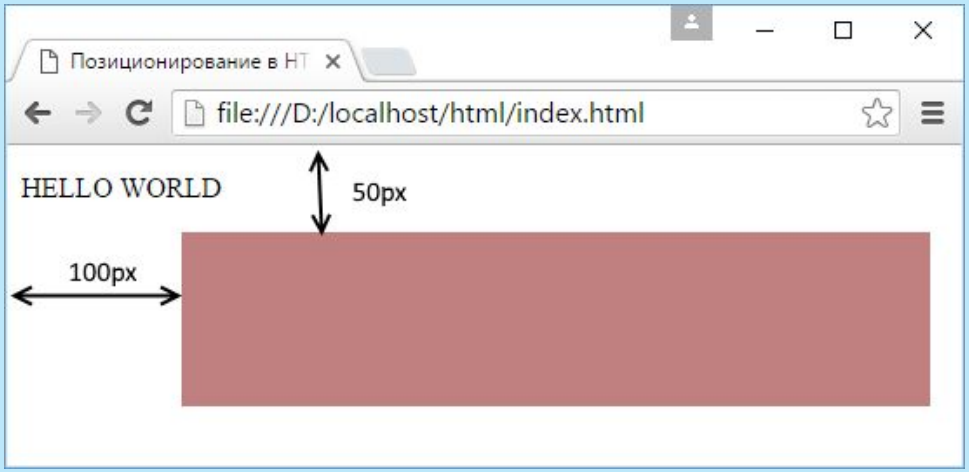
Основы CSS3

Свойство **position**. Это свойство может принимать одно из следующих значений:

- static: стандартное позиционирование элемента, значение по умолчанию
- absolute: элемент позиционируется относительно границ элемента-контейнера, если у того свойство position не равно static
- relative: элемент позиционируется относительно его позиции по умолчанию. Как правило, основная цель относительного позиционирования заключается не в том, чтобы переместить элемент, а в том, чтобы установить новую точку привязки для абсолютного позиционированния вложенных в него элементов
- fixed: элемент позиционируется относительно окна бразуера, это позволяет создать фиксированные элементы, которые не меняют положения при прокрутке

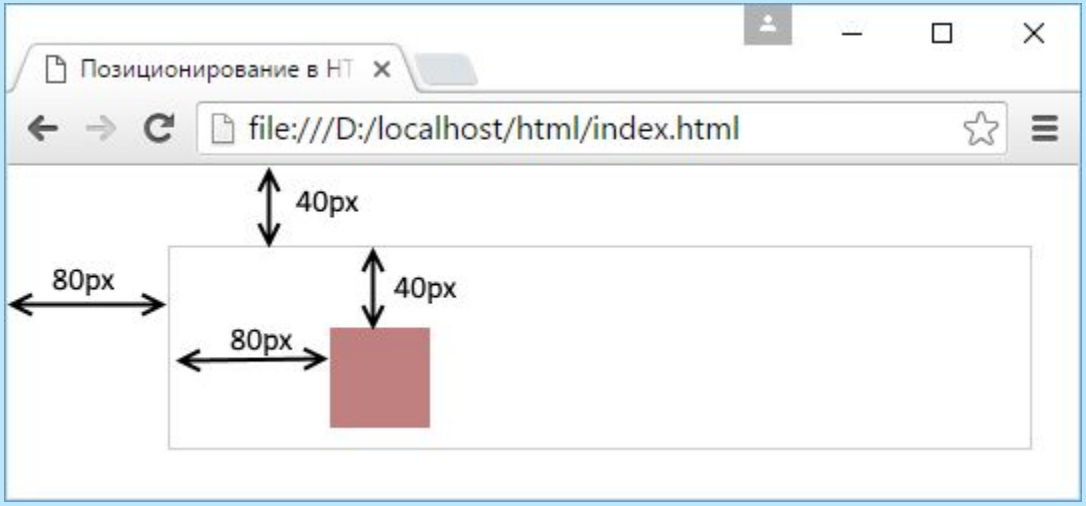
Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Блочная верстка в HTML5</title>
      <style>
         .header {
            position: absolute;
            left: 100px;
            top: 50px;
            width: 430px;
            height: 100px;
            background-color: rgba(128, 0, 0, 0.5);
         }
      </style>
   </head>
   <body>
      <div class="header"></div>
      <p>HELLO WORLD</p>
   </body>
</html>
```



Основы CSS3

```
<style>
        .outer {
            position: absolute;
            left: 80px;
            top: 40px;
            width: 430px;
            height: 100px;
            border: 1px solid #ccc;
        }
        .inner{
            position: absolute;
            left: 80px;
            top: 40px;
            width: 50px;
            height: 50px;
            background-color: rgba(128, 0, 0, 0.5);
        }
</style>
```



Основы CSS3

```
<style>
        .outer {
            position: relative;
            left: 80px;
            top: 40px;
            width: 300px;
            height: 100px;
            border: 1px solid #ccc;
        }
        .inner{
            position: absolute;
            left: 80px;
            top: 40px;
            width: 50px;
            height: 50px;
            background-color: rgba(128, 0, 0, 0.5);
        }
</style>
```

Основы CSS3

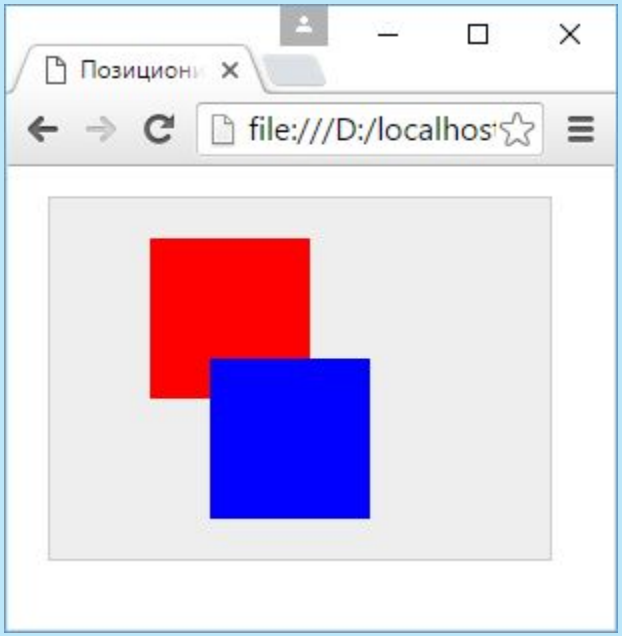
По умолчанию при совпадении у двух элементов границ, поверх другого отображается тот элемент, который определен в разметке html последним.

Однако свойство **z-index** позволяет изменить порядок следования элементов при их наложении.

В качестве значения свойство принимает число. Элементы с большим значением этого свойства будут отображаться поверх элементов с меньшим значением z-index.

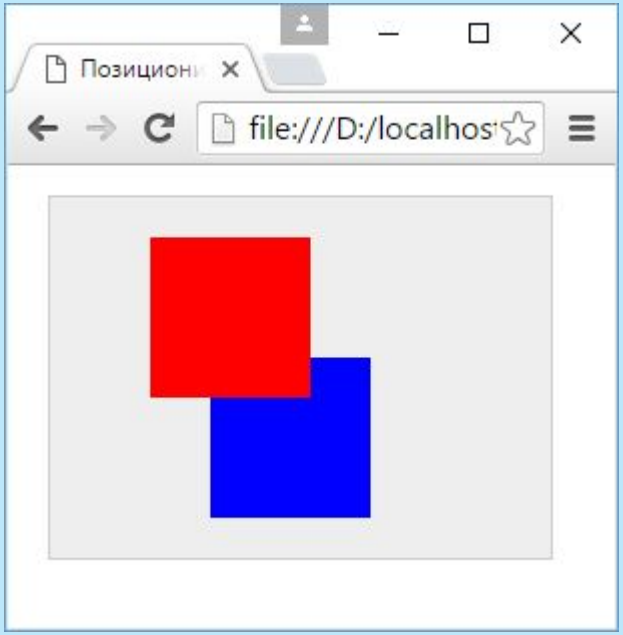
Основы CSS3

```
<style>
        body{
            margin:0;
            padding:0;
        }
        .content{
            position: relative;
            top: 15px; left: 20px;
            width: 250px; height: 180px;
            background-color: #eee;
            border: 1px solid #ccc;
        }
        .redBlock{
            position: absolute;
            top: 20px;              left:50px;
            width: 80px;              height: 80px;
            background-color: red;
        }
        .blueBlock{
            position: absolute;
            top: 80px;              left: 80px;
            width: 80px;              height: 80px;
            background-color: blue;
        }
</style>
```



Основы CSS3

```
.redBlock{
    z-index: 100;
    position: absolute;
    top: 20px;
    left:50px;
    width: 80px;
    height: 80px;
    background-color: red;
}
```



Основы CSS3

Одним из нововведений CSS3 по сравнению с предыдущей версией является встроенная возможность трансформаций элемента.

К трансформациям относятся такие действия, как вращение элемента, его масштабирование, наклон или перемещение по вертикали или горизонтали.

Для создания трансформаций в CSS3 применяется свойство **transform**.

Вращение

Для поворота элемента свойство transform использует функцию **rotate**:

transform: rotate(угол_поворота deg);

Основы CSS3

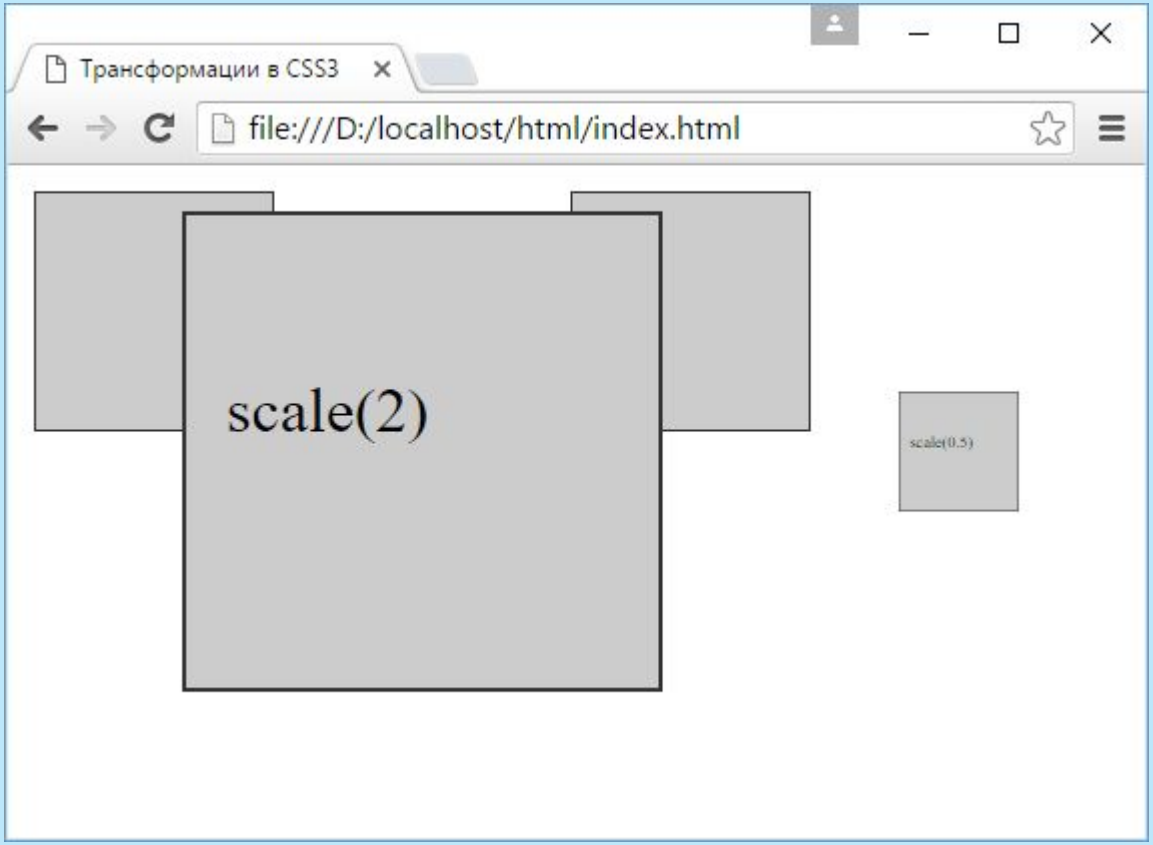
```
<style>
        div{
            background-color: #ccc;
            width: 120px;
            height: 120px;
            margin:5px;
            padding: 40px 10px;
            box-sizing: border-box;
            border: 1px solid #333;
            display: inline-block;
        }
        .rotated{
            transform: rotate(30deg);
        }
</style>
```

Основы CSS3

Применение масштабирования имеет следующую форму:

transform: scale(величина_масштабирования);

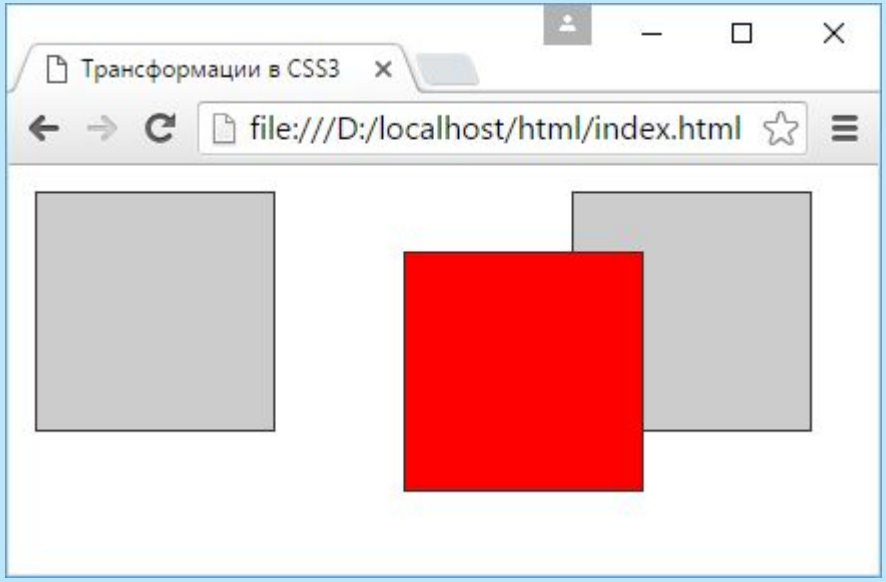
```
<style>
        div{
            background-color: #ccc;
            width: 120px;
            height: 120px;
            margin:5px;
            padding: 40px 10px;
            box-sizing: border-box;
            border: 1px solid #333;
            display: inline-block;
        }
        .halfScale{
            transform: scale(0.5);
        }
        .doubleScale{
            transform: scale(2);
        }
</style>
```



Основы CSS3

Для перемещения элемента используется функция **translate**:
transform: translate(offset_X, offset_Y);

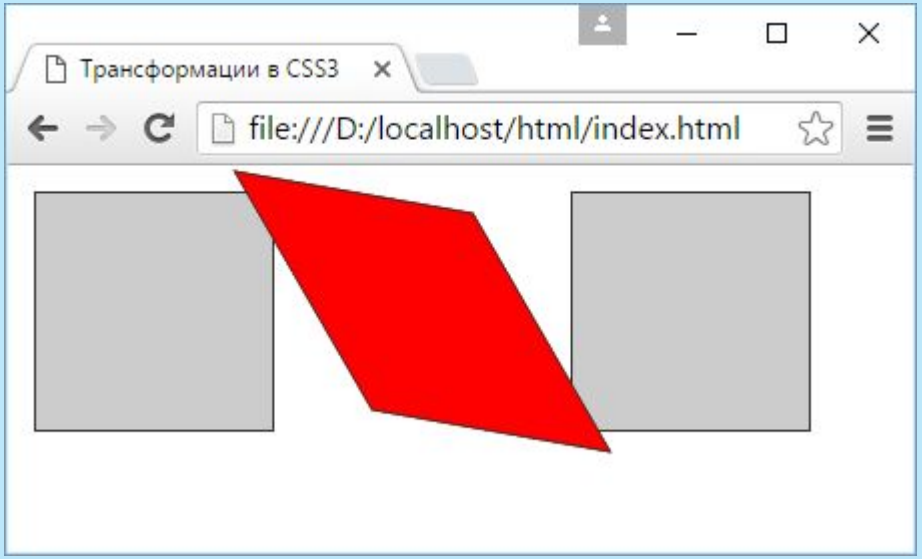
```
<style>
        div{
            background-color: #ccc;
            width: 120px;
            height: 120px;
            margin:5px;
            padding: 40px 10px;

            box-sizing: border-box;
            border: 1px solid #333;
            display: inline-block;
        }
        .translated{
            transform: translate(50px, 30px);
            background-color:red;
        }
</style>
```



Основы CSS3

Для наклона элемента применяется функция **skew()**:
transform: skew(X, Y);

```
<style>
        div{
            background-color: #ccc;
            width: 120px;
            height: 120px;
            margin:5px;
            padding: 40px 10px;
            box-sizing: border-box;
            border: 1px solid #333;
            display: inline-block;
        }
        .skewed{
            transform: skew(30deg, 10deg);
            background-color:red;
        }
</style>
```



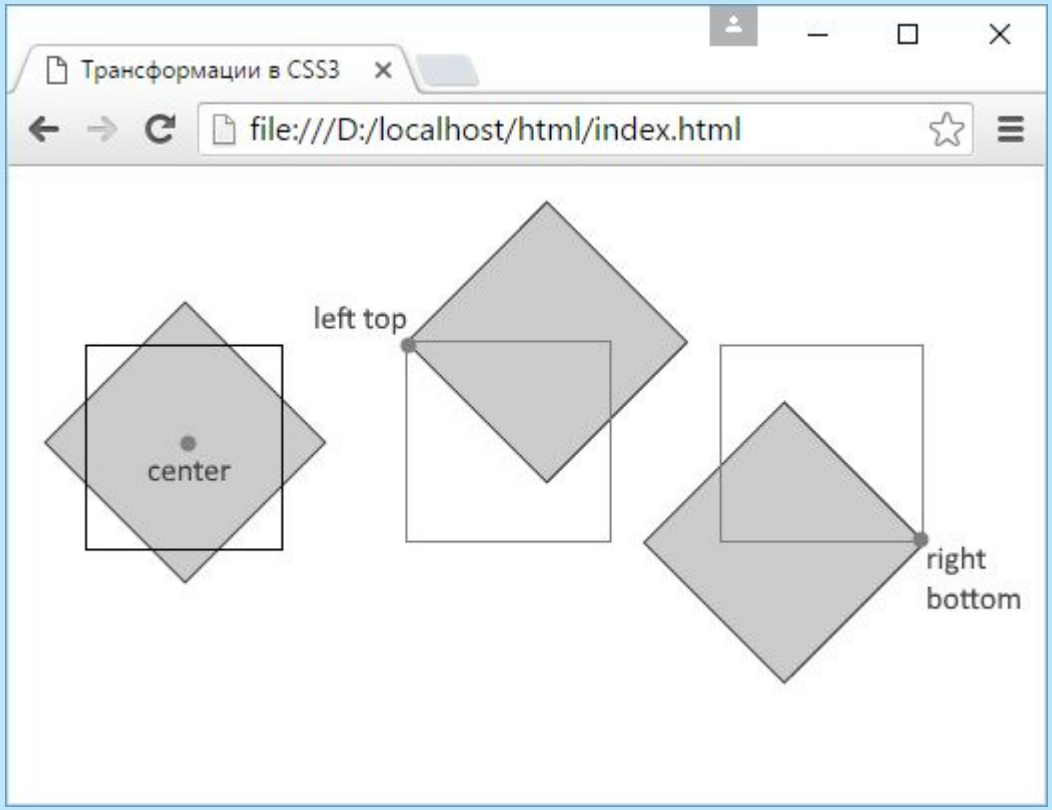
Основы CSS3

По умолчанию при применении трансформаций браузер в качестве точки начала преобразования использует центр элемента. Но с помощью свойства **transform-origin** можно изменить исходную точку:

- left top: левый верхний угол элемента
- left bottom: левый нижний угол элемента
- right top: правый верхний угол элемента
- right bottom: правый нижний угол элемента

Основы CSS3

```
<style>
        div{
            background-color: #ccc;
            width: 100px;
            height: 100px;
            margin: 80px 30px;
            float: left;
            box-sizing: border-box;
            border: 1px solid #333;
        }
        .transform1{
            transform: rotate(-45deg);
        }
        .transform2{
            transform-origin: left top;
            transform: rotate(-45deg);
        }
        .transform3{
            transform-origin: right bottom;
            transform: rotate(-45deg);
        }
</style>
```



Метатег **viewport**

Параметр	Значения	Описание
width	Принимает целочисленное значение в пикселях или значение device-width	Устанавливает ширину области viewport
height	Принимает целочисленное значение в пикселях или значение device-height	Устанавливает высоту области viewport
initial-scale	Число с плавающей точкой от 0.1 и выше	Задает коэффициент масштабирования начального размера viewport. Значение 1.0 задает отсутствие масштабирования
user-scalable	no/yes	Указывает, может ли пользователь с помощью жестов масштабировать страницу
minimum-scale	Число с плавающей точкой от 0.1 и выше	Задает минимальный масштаб размера viewport. Значение 1.0 задает отсутствие масштабирования
maximum-scale	Число с плавающей точкой от 0.1 и выше	Задает максимальный масштаб размера viewport. Значение 1.0 задает отсутствие масштабирования

Основы CSS3

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width">
        <title>Обычная веб-страница</title>
    </head>
    <body>
        <h2>Обычная веб-страница</h2>
    </body>
</html>
```

Основы CSS3

Механизм **CSS3 Media Query**

```
<html>
 <head>
  <title>Адаптивная веб-страница</title>
  <meta name="viewport" content="width=device-width">
  <link rel="stylesheet" type="text/css" href="desctop.css" />
  <link rel="stylesheet" type="text/css" media="(max-device-width:480px)" href="mobile.css" />
 </head>
 <body>
```

Основы CSS3

С помощью ключевого слова and можно комбинировать условия, например:

<link rel="stylesheet" type="text/css" media="(min-width:481px) and (max-width:768px)" href="mobile.css" />

С помощью директивый @import можно определить один css-файл и импортировать в него стили для определенных устройств:

@import url(desctop.css);

@import url(tablet.css) (min-device-width:481px) and (max-device-width:768);

@import url(mobile.css) (max-device-width:480px);

Основы CSS3

Можно не разделять стили по файлам, а использовать правила CSS3 Media Query в одном файле css:

```
body {
    background-color: red;
}
/*Далее остальные стили*/
@media (max-device-width:480px){
    body {
        background-color: blue;
    }
    /*Далее остальные стили*/
```

Основы CSS3

Применяемые функции в **CSS3 Media Query**:

- aspect-ratio: отношение ширины к высоте области отображения (браузера)
- device-aspect-ratio: отношение ширины к высоте экрана устройства
- max-width/min-width и max-height/min-height: максимальная и минимальная ширина и высота области отображения (браузера)
- max-device-width/min-device-width и max-device-height/min-device-height: максимальная и минимальная ширина и высота экрана мобильного устройства
- orientation: ориентация (портретная или альбомная)

Основы CSS3

Например, можем задать разные стили для разных ориентаций мобильных устройств:

```
/*Стили для портретной ориентации*/
@media only screen and (orientation: portrait){


}
/*Стили альбомной ориентации*/
@media only screen and (orientation: landscape){


}
```

Основы CSS3

Для воспроизведения видео в HTML5 используется элемент **video**.

- src: источник видео, это может быть какой-либо видеофайл
- width: ширина элемента
- height: высота элемента
- controls: добавляет элементы управления воспроизведением
- autoplay: устанавливает автовоспроизведение
- loop: задает повторение видео
- muted: отключает звук по умолчанию

Основы CSS3

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Видео в HTML5</title>
    </head>
    <body>
        <video src= "video.mp4" width="400" height="300" controls ></video>
    </body>
</html>
```

Основы CSS3

Атрибут preload призван управлять загрузкой видео.

- auto: видео и связанные с ним метаданные будут загружаться до того, как видео начнет воспроизводиться.
- none: видео не будет загружаться в фоне, пока пользователь не нажмет на кнопку начала проигрывания
- metadata: в фоне до воспроизведения будут загружаться только метаданные (данные о формате, длительности и т.д), само видео не загружается

```
<video src= "video.mp4" width="400" height="300" controls preload="auto"></video>
```

Основы CSS3

Атрибут poster позволяет установить изображение, которое будет отображаться до запуска видео. Этому атрибуту в качестве значения передается путь к изображению:

```
<video src="tea.mp4" width="400" height="300" controls poster="tea.jpg"></video>
```

Основы CSS3

Поддержка форматов видео

Главной проблемой при использовании элемента **video** является поддержка различными веб-браузерами определенных форматов. С помощью вложенных элементов **source** можно задать несколько источников видео, один из которых будет использоваться:

```
<video width="400" height="300" controls>
    <source src="tea.mp4" type="video/mp4">
    <source src="tea.webm" type="video/webm">
    <source src="tea.ogv" type="video/ogg">
</video>
```

Элемент source использует два атрибута для установки источника видео:

- src: путь к видеофайлу
- type: тип видео (MIME-тип)

Основы CSS3

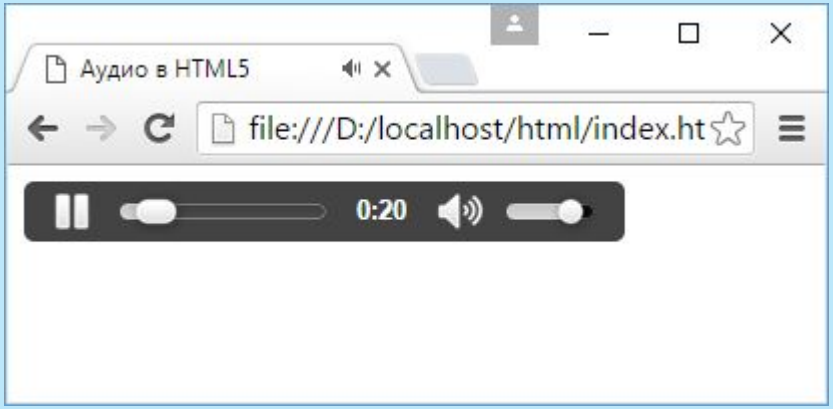
Для воспроизведения звука в HTML5 применяется элемент **audio**:

- src: путь к аудиофайлу
- controls: добавляет элементы управления воспроизведением
- autoplay: устанавливает автовоспроизведение
- loop: задает повторение аудиофайла
- muted: отключает звук по умолчанию
- preload: устанавливает режим загрузки файла

Действие всех этих атрибутов будет аналогично их действию в элементе video.

Основы CSS3

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>Аудио в HTML5</title>
   </head>
   <body>
      <audio src="mobile_phone.mp3" controls></audio>
   </body>
</html>
```



Поддержка форматов аудио

Можно использовать вложенный элемент source и указать аудио в иных форматах:

```
<audio width="400" height="300" controls>
    <source src="mobile_phone.mp3" type="audio/mpeg">
    <source src="mobile_phone.m4a" type="audio/aac">
    <source src="mobile_phone.ogg" type="audio/ogg">
</audio>
```

Как и в случае с элементом video, здесь у элемента source устанавливается атрибут src с ссылкой на файл и атрибут type - тип файла.