

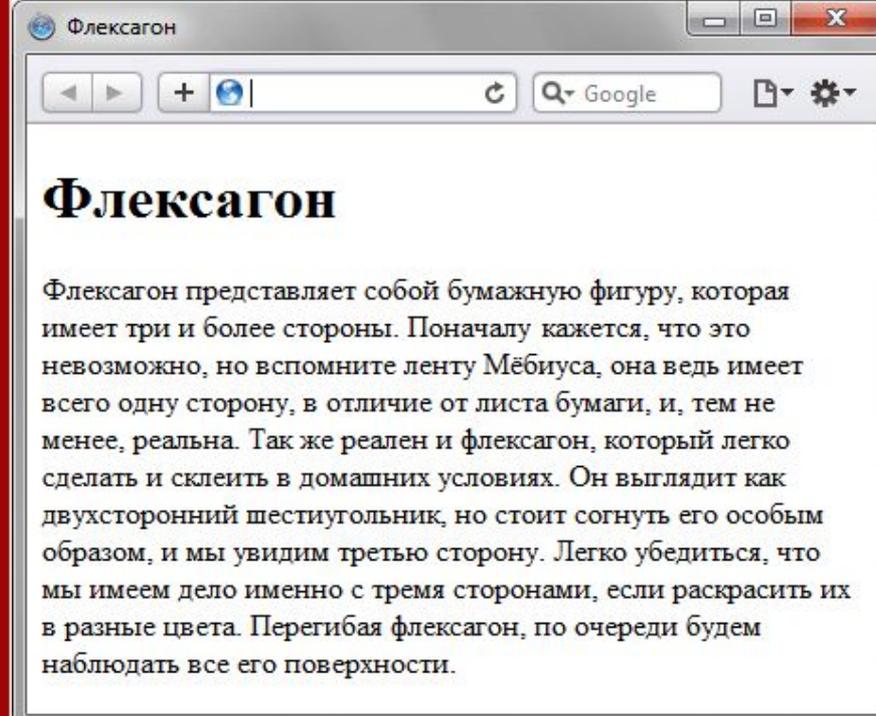
Тема 4-1. Каскадные таблицы стилей



Каскадные таблицы стилей

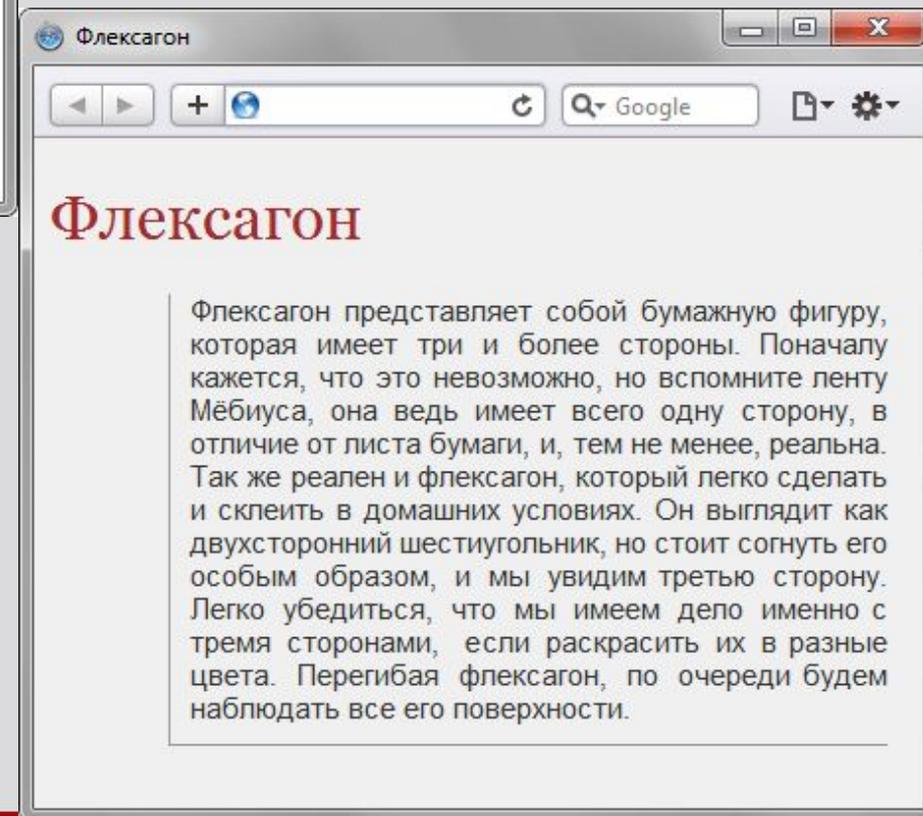
- Стилем или CSS (Cascading Style Sheets, каскадные таблицы стилей) называется набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид.
- Возможность работы со стилями издавна включают в развитые издательские системы и текстовые редакторы, тем самым позволяя одним нажатием кнопки придать тексту заданный, заранее установленный вид.
- Теперь это доступно и создателям сайта, когда цвет, размеры текста и другие параметры хранятся в определенном месте и легко «прикручиваются» к любому тегу.
- Еще одним преимуществом стилей является то, что они предлагают намного больше возможностей для форматирования, чем обычный HTML.
- CSS представляет собой мощную систему, расширяющую возможности дизайна и верстки веб-страниц.





Веб-страница, созданная только на HTML

Веб-страница, созданная на HTML и CSS

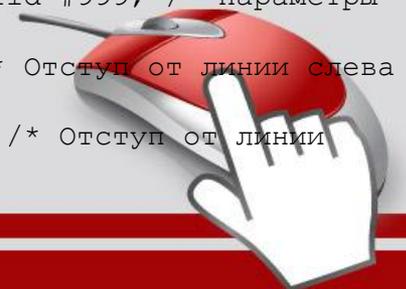


Исходный код документа

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Флексагон</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Флексагон</h1>
    <p>Флексагон представляет собой бумажную фигуру, которая имеет три и более стороны. Поначалу кажется, что это невозможно, но вспомните ленту Мёбиуса, она ведь имеет всего одну сторону, в отличие от листа бумаги, и, тем не менее, реальна. Так же реален и флексагон, который легко сделать и склеить в домашних условиях. Он выглядит как двухсторонний шестиугольник, но стоит согнуть его особым образом, и мы увидим третью сторону. Легко убедиться, что мы имеем дело именно с тремя сторонами, если раскрасить их в разные цвета. Перегибая флексагон, по очереди будем наблюдать все его поверхности.</p>
  </body>
</html>
```

Содержимое стилевого файла style.css

```
body {
  font-family: Arial, Verdana, sans-serif; /* Семейство шрифтов */
  font-size: 11pt; /* Размер основного шрифта в пунктах */
  background-color: #f0f0f0; /* Цвет фона веб-страницы */
  color: #333; /* Цвет основного текста */
}
h1 {
  color: #a52a2a; /* Цвет заголовка */
  font-size: 24pt; /* Размер шрифта в пунктах */
  font-family: Georgia, Times, serif; /* Семейство шрифтов */
  font-weight: normal; /* Нормальное начертание текста */
}
p {
  text-align: justify; /* Выравнивание по ширине */
  margin-left: 60px; /* Отступ слева в пикселах */
  margin-right: 10px; /* Отступ справа в пикселах */
  border-left: 1px solid #999; /* Параметры линии слева */
  border-bottom: 1px solid #999; /* Параметры линии снизу */
  padding-left: 10px; /* Отступ от линии слева до текста */
  padding-bottom: 10px; /* Отступ от линии снизу до текста */
}
```



ПОНЯТИЕ CSS СТИЛЕЙ



Типы стилей

- Различают несколько типов стилей, которые могут совместно применяться к одному документу. Это стиль браузера, стиль автора и стиль пользователя.
 - **Стиль браузера**
Оформление, которое по умолчанию применяется к элементам веб-страницы браузером. Это оформление можно увидеть в случае «голового» HTML, когда к документу не добавляется никаких стилей. Например, заголовок страницы, формируемый тегом **<H1>**, в большинстве браузеров выводится шрифтом с засечками размером 24 пункта.
 - **Стиль автора**
Стиль, который добавляет к документу его разработчик.
 - **Стиль пользователя**
Это стиль, который может включить пользователь сайта через настройки браузера. Такой стиль имеет более высокий приоритет и переопределяет исходное оформление документа. В браузере Internet Explorer подключение стиля пользователя делается через меню Сервис > Свойство обозревателя > Кнопка «Оформление»,.



Преимущества стилей...

- Стили являются удобным, практичным и эффективным инструментом при вёрстке веб-страниц и оформлении текста, ссылок, изображений и других элементов. Несмотря на явные плюсы применения стилей, рассмотрим все преимущества CSS, в том числе и незаметные на первый взгляд.
 - **Разграничение кода и оформления**
Идея о том, чтобы код HTML был свободен от элементов оформления вроде установки цвета, размера шрифта и других параметров, стара как мир. В идеале, веб-страница должна содержать только теги логического форматирования, а вид элементов задаётся через стили. При подобном разделении работа над дизайном и версткой сайта может вестись параллельно.
 - **Разное оформление для разных устройств**
С помощью стилей можно определить вид веб-страницы для разных устройств вывода: монитора, принтера, смартфона, планшета и др. Например, на экране монитора отображать страницу в одном оформлении, а при её печати — в другом. Эта возможность также позволяет скрывать или показывать некоторые элементы документа при отображении на разных устройствах.
 - **Расширенные по сравнению с HTML способы оформления элементов**
В отличие от HTML стили имеют гораздо больше возможностей по оформлению элементов веб-страниц. Простыми средствами можно изменить цвет фона элемента, добавить рамку, установить шрифт, определить размеры, положение и многое другое.



Преимущества стилей

- **Ускорение загрузки сайта**

При хранении стилей в отдельном файле, он кэшируется и при повторном обращении к нему извлекается из кэша браузера. За счёт кэширования и того, что стили хранятся в отдельном файле, уменьшается код веб-страниц и снижается время загрузки документов.

Кэшем называется специальное место на локальном компьютере пользователя, куда браузер сохраняет файлы при первом обращении к сайту. При следующем обращении к сайту эти файлы уже не скачиваются по сети, а берутся с локального диска. Такой подход позволяет существенно повысить скорость загрузки веб-страниц.

- **Единое стилевое оформление множества документов**

Сайт это не просто набор связанных между собой документов, но и одинаковое расположение основных блоков, и их вид. Применение единообразного оформления заголовков, основного текста и других элементов создает преемственность между страницами и облегчает пользователям работу с сайтом и его восприятие в целом. Разработчикам же использование стилей существенно упрощает проектирование дизайна.

- **Централизованное хранение**

Стили, как правило, хранятся в одном или нескольких специальных файлах, ссылка на которые указывается во всех документах сайта. Благодаря этому удобно править стиль в одном месте, при этом оформление элементов автоматически меняется на всех страницах, которые связаны с указанным файлом. Вместо того чтобы модифицировать десятки HTML-файлов, достаточно отредактировать один файл со стилем и оформление нужных документов сразу же поменяется.



Способы добавления стилей на страницу...

- Связанные стили
 - При использовании связанных стилей описание селекторов и их значений располагается в отдельном файле, как правило, с расширением `css`, а для связывания документа с этим файлом применяется тег `<link>`. Данный тег помещается в контейнер `<head>`



Подключение связанных стилей

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Стили</title>
    <link rel="stylesheet"
href="http://htmlbook.ru/mysite.css">
    <link rel="stylesheet"
href="http://www.htmlbook.ru/main.css"
>
  </head>
  <body>
    <h1>Заголовок</h1>
    <p>Текст</p>
  </body>
</html>
```

Файл со стилем

```
H1 {
  color: #000080;
  font-size: 200%;
  font-family: Arial, Verdana,
sans-serif;
  text-align: center;
}
P {
  padding-left: 20px;
}
```



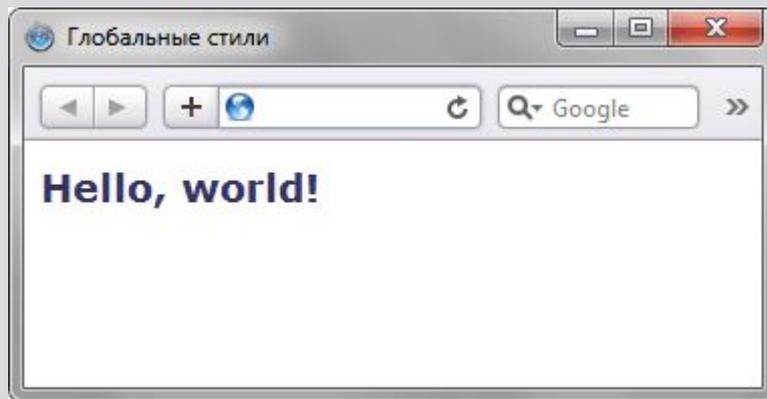
Способы добавления стилей на страницу...

- Глобальные стили
- При использовании глобальных стилей свойства CSS описываются в самом документе и располагаются в заголовке веб-страницы. По своей гибкости и возможностям этот способ добавления стиля уступает предыдущему, но также позволяет хранить стили в одном месте, в данном случае прямо на той же странице с помощью контейнера **<style>**



Использование глобального стиля

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Глобальные стили</title>
    <style>
      H1 {
        font-size: 120%;
        font-family: Verdana, Arial, Helvetica, sans-serif;
        color: #333366;
      }
    </style>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```



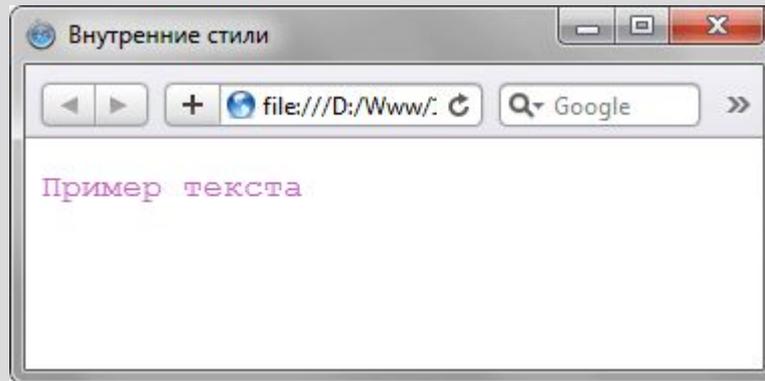
Способы добавления стилей на страницу...

- Внутренние стили
- Внутренний или встроенный стиль является по существу расширением для одиночного тега используемого на текущей веб-странице. Для определения стиля используется атрибут `style`, а его значением выступает набор стилевых правил



Использование внутреннего стиля

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Внутренние стили</title>
  </head>
  <body>
    <p style="font-size: 120%; font-family: monospace; color:
#cd66cc">Пример текста</p>
  </body>
</html>
```



Способы добавления стилей на страницу...

- Импорт CSS
- В текущую стилевую таблицу можно импортировать содержимое CSS-файла с помощью команды `@import`. Этот метод допускается использовать совместно со связанными или глобальными стилями, но никак не с внутренними стилями. Общий синтаксис следующий.

`@import url("имя файла")` типы носителей;

`@import "имя файла"` типы носителей;

- После ключевого слова `@import` указывается путь к стилевому файлу одним из двух приведенных способов — с помощью `url` или без него.



Импорт CSS

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Импорт</title>
    <style>
      @import
url("style/header.css");
      H1 {
        font-size: 120%;
        font-family: Arial, Helvetica,
sans-serif;
        color: green;
      }
    </style>
  </head>
  <body>
    <h1>Заголовок 1</h1>
    <h2>Заголовок 2</h2>
  </body>
</html>
```

Импорт в таблице связанных стилей

```
@import "/style/print.css";
@import "/style/palm.css";
BODY {
  font-family: Arial, Verdana,
Helvetica, sans-serif;
  font-size: 90%;
  background: white;
  color: black;
}
```



ТИПЫ НОСИТЕЛЕЙ



- Широкое развитие различных платформ и устройств заставляет разработчиков делать под них специальные версии сайтов, что достаточно трудоёмко и проблематично. Вместе с тем, времена и потребности меняются, и создание сайта для различных устройств является неизбежным и необходимым звеном его развития.
- С учетом этого в CSS введено понятие типа носителя, когда стиль применяется только для определённого устройства.

Тип	Описание
all	Все типы. Это значение используется по умолчанию.
aural	Речевые синтезаторы, а также программы для воспроизведения текста вслух. Сюда, например, можно отнести речевые браузеры.
braille	Устройства, основанные на системе Брайля, которые предназначены для слепых людей.
handheld	Наладонные компьютеры и аналогичные им аппараты.
print	Печатающие устройства вроде принтера.
projection	Проектор.
screen	Экран монитора.
tv	Телевизор.

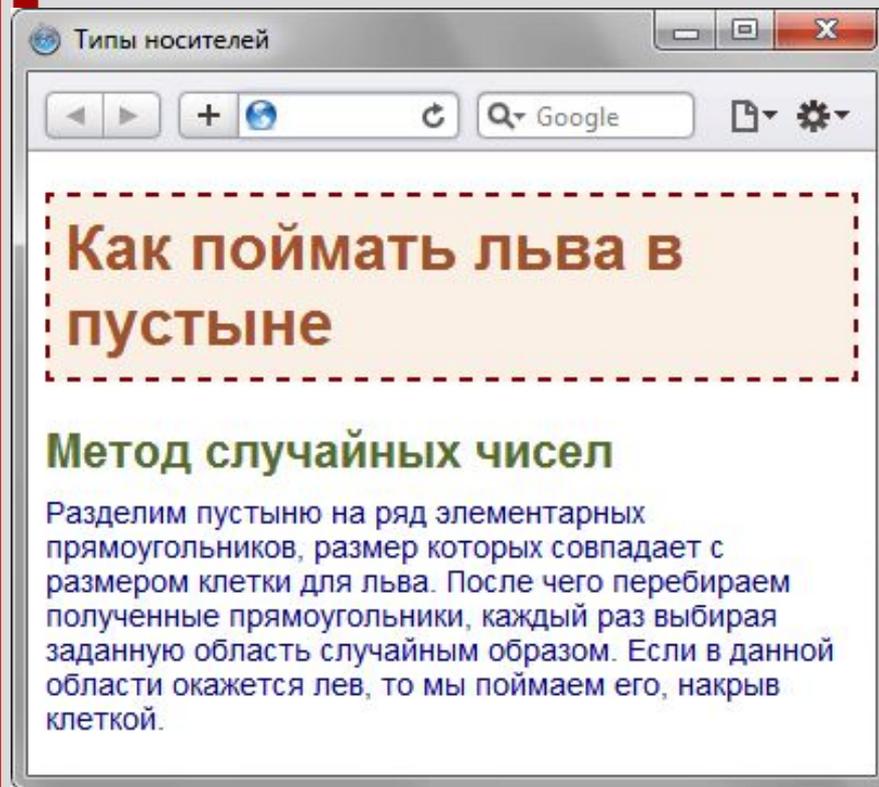
В CSS для указания типа носителей применяются команды `@import` и `@media`, с помощью которых можно определить стиль для элементов в зависимости от того, выводится документ на экран или на принтер.



Стили для разных типов носителей

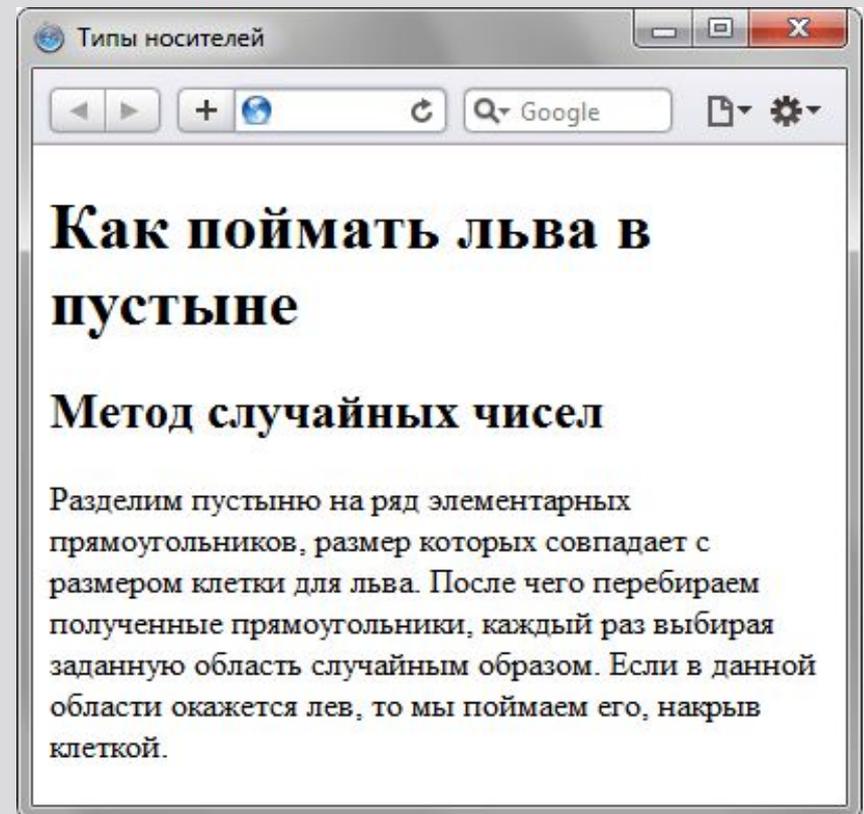
```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Типы носителей</title>
    <style>
      @media screen { /* Стил ь для отображения в браузере */
        BODY {
          font-family: Arial, Verdana, sans-serif; /* Рубленый шрифт */
          font-size: 90%; /* Размер шрифта */
          color: #000080; /* Цвет текста */
        }
        H1 {
          background: #faf0e6; /* Цвет фона */
          border: 2px dashed maroon; /* Рамка вокруг заголовка */
          color: #a0522d; /* Цвет текста */
          padding: 7px; /* Поля вокруг текста */
        }
        H2 {
          color: #556b2f; /* Цвет текста */
          margin: 0; /* Убираем отступы */
        }
        P {
          margin-top: 0.5em; /* Отступ сверху */
        }
      }
      @media print { /* Стил ь для печати */
        BODY {
          font-family: Times, 'Times New Roman', serif; /* Шрифт с засечками */
        }
        H1, H2, P {
          color: black; /* Чёрный цвет текста */
        }
      }
    </style>
  </head>
  <body>
    <h1>Как поймать льва в пустыне</h1>
    <h2>Метод случайных чисел</h2>
    <p>Разделим пустыню на ряд элементарных прямоугольников, размер которых совпадает с размером клетки для льва. После чего перебираем полученные прямоугольники, каждый раз выбирая заданную область случайным образом. Если в данной области окажется лев, то мы поймаем его, накрыв клеткой.</p>
  </body>
</html>
```





Страница для отображения в окне браузера

Страница, предназначенная для печати



БАЗОВЫЙ СИНТАКСИС CSS



Основным понятием выступает селектор — это некоторое имя стиля, для которого добавляются параметры форматирования. В качестве селектора выступают теги, классы и идентификаторы. Общий способ записи имеет следующий вид.

селектор	свойство	значение
┌──────────┐	┌──────────┐	┌──────────┐
body	{ background:	#ffc910; }

Вначале пишется имя селектора, например, **TABLE**, это означает, что все стилевые параметры будут применяться к тегу **<table>**, затем идут фигурные скобки, в которых записывается стилевое свойство, а его значение указывается после двоеточия. Силевые свойства разделяются между собой точкой с запятой, в конце этот символ можно опустить.

CSS не чувствителен к регистру, переносу строк, пробелам и символам табуляции, поэтому форма записи зависит от желания разработчика.



Форма записи

- Для селектора допускается добавлять каждое стилевое свойство и его значение по **отдельности**

Расширенная форма записи

```
td { background: olive; }  
td { color: white; }  
td { border: 1px solid  
black; }
```

Компактная форма записи

```
td  
{  
background: olive;   color: white;  
border: 1px solid black; }
```



Комментарии

- Комментарии нужны, чтобы делать пояснения по поводу использования того или иного стилевого свойства, выделять разделы или писать свои заметки.
- Комментарии позволяют легко вспоминать логику и структуру селекторов, и повышают разборчивость кода.
- Вместе с тем, добавление текста увеличивает объём документов, что отрицательно сказывается на времени их загрузки.
- Поэтому комментарии обычно применяют в отладочных или учебных целях, а при выкладывании сайта в сеть их стирают.
- Чтобы пометить, что текст является комментарием, применяют следующую конструкцию `/* ... */`



Комментарии в CSS-файле

```
/*  
  Стил ь для сайта htmlbook.ru  
  Сделан для ознакомительных целей  
*/  
  
div {  
  width: 200px; /* Ширина блока */  
  margin: 10px; /* Поля вокруг элемента */  
  float: left; /* Обтекание по правому краю */  
}
```



ЗНАЧЕНИЯ СТИЛЕВЫХ СВОЙСТВ



Строки

- Любые строки необходимо брать в двойные или одинарные кавычки.
- Если внутри строки требуется оставить одну или несколько кавычек, то можно комбинировать типы кавычек или добавить перед кавычкой слэш

- **Допустимые строки**

‘Гостиница “Турист”’

“Гостиница ‘Турист’”

“Гостиница \”Турист\””

- В данном примере в первой строке применяются одинарные кавычки, а слово «Турист» взято в двойные кавычки.
- Во второй строке всё с точностью до наоборот, в третьей же строке используются только двойные кавычки, но внутренние экранированы с помощью слэша.



Числа

- Значением может выступать целое число, содержащее цифры от 0 до 9 и десятичная дробь, в которой целая и десятичная часть разделяются точкой

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Числа</title>
    <style>
      p {
        font-weight: 600; /* Жирное
начертание */
        line-height: 1.2; /*
Межстрочный интервал */
      }
    </style>
  </head>
  <body>
    <p>Пример текста</p>
  </body>
</html>
```



Проценты

- Процентная запись обычно применяется в тех случаях, когда надо изменить значение относительно родительского элемента или когда размеры зависят от внешних условий.
- Так, ширина таблицы 100% означает, что она будет подстраиваться под размеры окна браузера и меняться вместе с шириной окна

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ширина в процентах</title>
    <style>
      TABLE {
        width: 100%; /* Ширина таблицы
в процентах */
        background: #f0f0f0; /* Цвет
фона */
      }
    </style>
  </head>
  <body>
    <table>
      <tr><td>Содержимое
таблицы</td></tr>
    </table>
  </body>
</html>
```



Размеры

- Для задания размеров различных элементов, в CSS используются абсолютные и относительные единицы измерения.
- Абсолютные единицы не зависят от устройства вывода, а относительные единицы определяют размер элемента относительно значения другого размера.



Относительные единицы

- Относительные единицы обычно используют для работы с текстом, либо когда надо вычислить процентное соотношение между элементами.

Единица	Описание
em	Размер шрифта текущего элемента
ex	Высота символа x
px	Пиксел
%	Процент

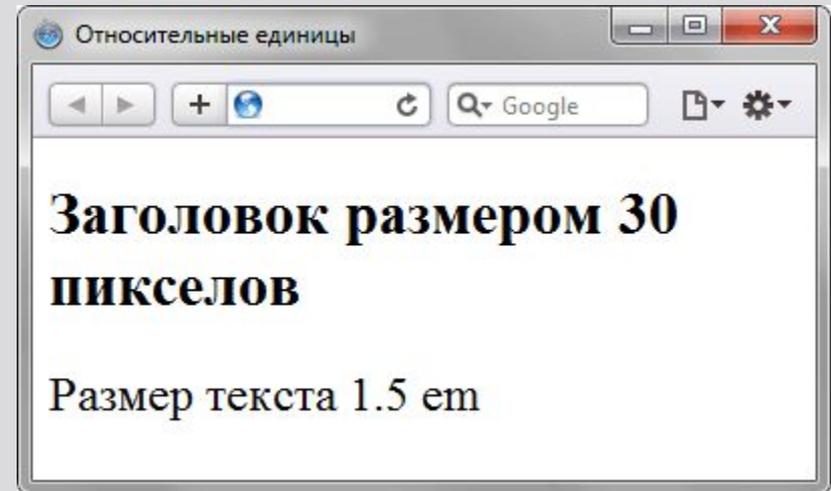
Единица em это изменяемое значение, которое зависит от размера шрифта текущего элемента (размер устанавливается через стилевое свойство font-size). В каждом браузере заложен размер текста, применяемый в том случае, когда этот размер явно не задан. Поэтому изначально 1em равен размеру шрифта, заданного в браузере по умолчанию или размеру шрифта родительского элемента. Процентная запись идентична em, в том смысле, что значения 1em и 100% равны.

Единица ex определяется как высота символа «x» в нижнем регистре. На ex распространяются те же правила, что и для em, а именно, он привязан к размеру шрифта, заданного в браузере по умолчанию, или к размеру шрифта родительского элемента.



Использование относительных единиц

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Относительные
единицы</title>
    <style>
      H1 { font-size: 30px; }
      P { font-size: 1.5em; }
    </style>
  </head>
  <body>
    <h1>Заголовок размером 30
пикселей</h1>
    <p>Размер текста 1.5em</p>
  </body>
</html>
```



Абсолютные единицы

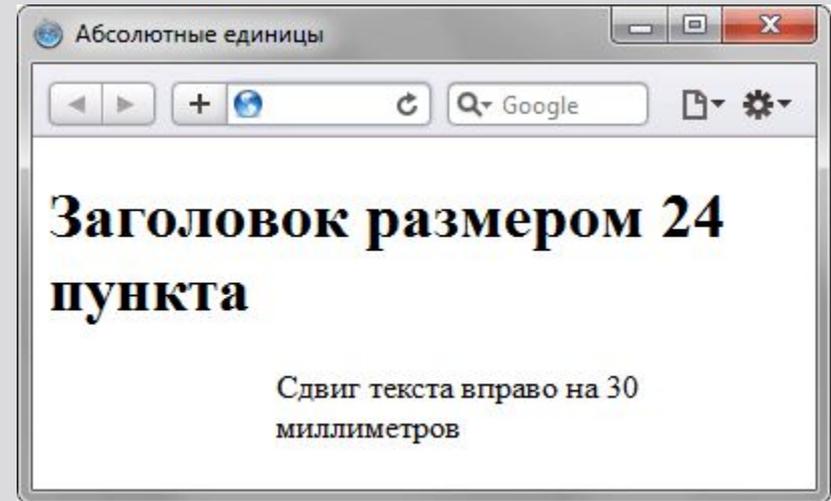
- Абсолютные единицы применяются реже, чем относительные и обычно при работе с текстом.
- Самой, пожалуй, распространенной единицей является пункт, который используется для указания размера шрифта.
- Хотя мы привыкли измерять все в миллиметрах и подобных единицах, пункт, пожалуй, единственная величина из не метрической системы измерения, которая используется у нас повсеместно.
- И все благодаря текстовым редакторам и издательским системам.

Единица	Описание
in	Дюйм (1 дюйм равен 2,54 см)
cm	Сантиметр
mm	Миллиметр
pt	Пункт (1 пункт равен 1/72 дюйма)
pc	Пика (1 пика равна 12 пунктам)



Использование абсолютных единиц

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Абсолютные единицы</title>
    <style>
      H1 { font-size: 24pt; }
      P { margin-left: 30mm; }
    </style>
  </head>
  <body>
    <h1>Заголовок размером 24 пункта</h1>
    <p>Сдвиг текста вправо на 30
миллиметров</p>
  </body>
</html>
```



Цвет

- Цвет в стилях можно задавать тремя способами: по шестнадцатеричному значению, по названию и в формате RGB.
- **По шестнадцатеричному значению**
- Для задания цветов используются числа в шестнадцатеричном коде. Допускается использовать сокращенную форму вида #rgb, где каждый символ следует удваивать (#rrggbb). К примеру, запись #fe0 расценивается как #ffee00.
- **По названию**
- Браузеры поддерживают некоторые цвета по их названию.
- **С помощью RGB**
- Можно определить цвет, используя значения красной, зелёной и синей составляющей в десятичном исчислении. Значение каждого из трех цветов может принимать значения от 0 до 255. Также можно задавать цвет в процентном отношении. Вначале указывается ключевое слово rgb, а затем в скобках, через запятую указываются компоненты цвета, например rgb(255, 0, 0) или rgb(100%, 20%, 20%).



Представление цвета

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Цвета</title>
    <style>
      BODY {
        background-color: #3366CC; /* Цвет фона веб-
страницы */
      }
      H1 {
        background-color: RGB(249, 201, 16); /* Цвет фо
под заголовком */
      }
      P {
        background-color: maroon; /* Цвет фона под
текстом абзаца */
        color: white; /* Цвет текста */
      }
    </style>
  </head>
  <body>
    <h1>Lorem ipsum dolor sit amet</h1>
    <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed diem
nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat.</p>
  </body>
</html>
```



Адреса

- Адреса (*URI*, Uniform Resource Identifiers, унифицированный идентификатор ресурсов) применяются для указания пути к файлу, например, для установки фоновой картинки на странице.
- Для этого применяется ключевое слово `url()`, внутри скобок пишется относительный или абсолютный адрес файла.
- При этом адрес можно задавать в необязательных одинарных или двойных кавычках



Адрес графического файла

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Добавление фона</title>
    <style>
      body {
        background: url('http://webimg.ru/images/156_1.png') no-repeat;
      }
      div {
        background: url(images/warning.png) no-repeat;
        padding-left: 20px;
        margin-left: 200px;
      }
    </style>
  </head>
  <body>
    <div>Внимание, запрашиваемая страница не найдена!</div>
  </body>
</html>
```



Ключевые слова

- В качестве значений активно применяются ключевые слова, которые определяют желаемый результат действия стилевых свойств.
- Ключевые слова пишутся без кавычек.

Правильно: `P { text-align: right; }`

Неверно: `P { text-align: "right"; }`



Селекторы тегов

- В качестве селектора может выступать любой тег HTML, для которого определяются правила форматирования, такие как: цвет, фон, размер и т. д. Правила задаются в следующем виде.

Тег { свойство1: значение; свойство2: значение; ... }

- Вначале указывается имя тега, оформление которого будет переопределено, заглавными или строчными символами не имеет значения.
- Внутри фигурных скобок пишется стилевое свойство, а после двоеточия — его значение.
- Набор свойств разделяется между собой точкой с запятой и может располагаться как в одну строку, так и в несколько



Изменение стиля тега абзаца

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Селекторы тегов</title>
    <style>
      P {
        text-align: justify; /* Выравнивание по ширине */
        color: green; /* Зеленый цвет текста */
      }
    </style>
  </head>
  <body>
    <p>Более эффективным способом ловли льва в пустыне является метод
золотого сечения. При его использовании пустыня делится на две
неравные части, размер которых подчиняется правилу золотого
сечения.</p>
  </body>
</html>
```



КЛАССЫ



Классы

- Классы применяют, когда необходимо определить стиль для индивидуального элемента веб-страницы или задать разные стили для одного тега.
- При использовании совместно с тегами синтаксис для классов будет следующий.

Тег.Имя класса { свойство1: значение; свойство2: значение; ... }

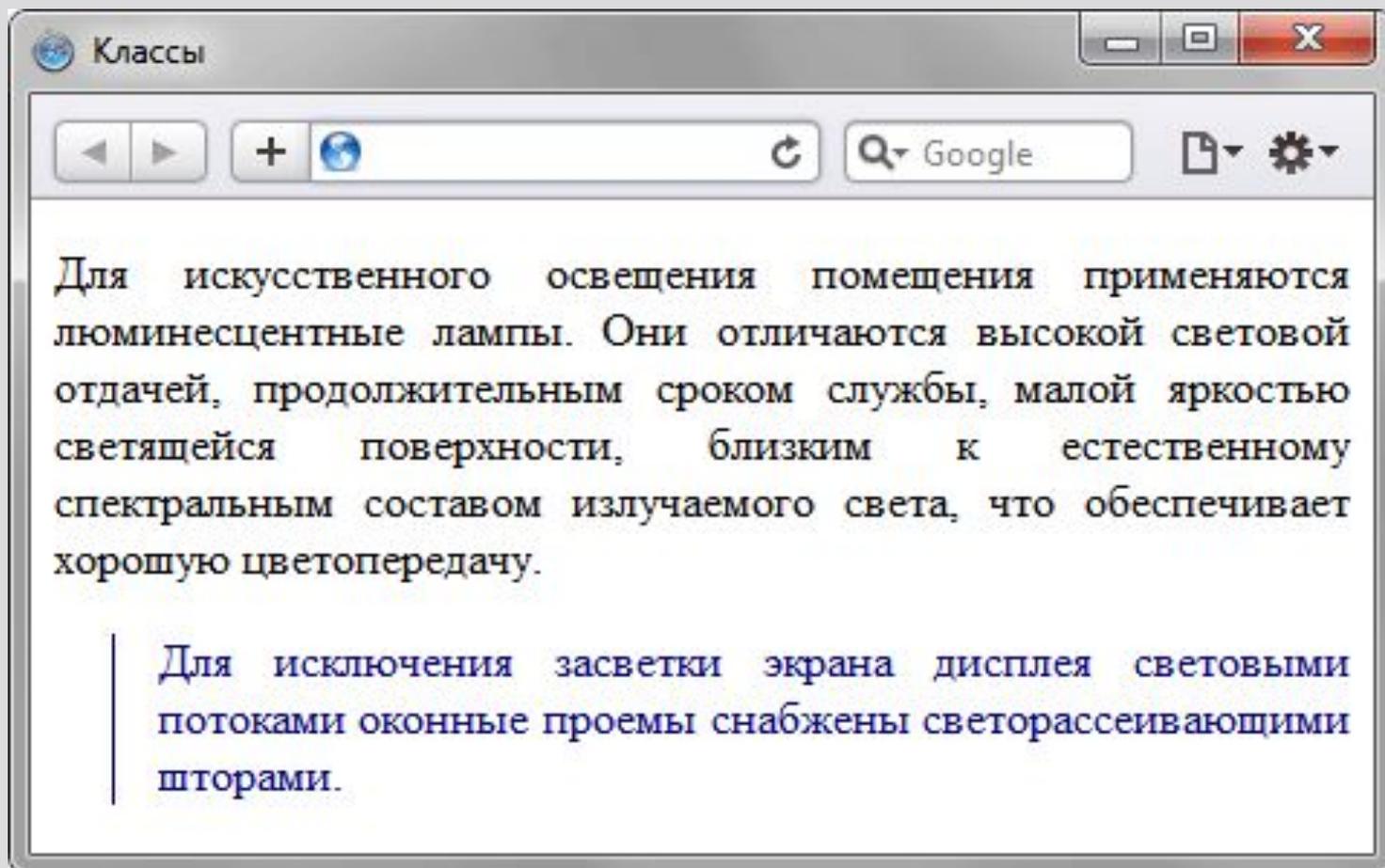
- Внутри стиля вначале пишется желаемый тег, а затем, через точку пользовательское имя класса.
- Имена классов должны начинаться с латинского символа и могут содержать в себе символ дефиса (-) и подчеркивания (_).
- Использование русских букв в именах классов недопустимо.
- Чтобы указать в коде HTML, что тег используется с определённым классом, к тегу добавляется атрибут `class="Имя класса"`



Использование классов

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Классы</title>
    <style>
      P { /* Обычный абзац */
        text-align: justify; /* Выравнивание текста по ширине */
      }
      P.cite { /* Абзац с классом cite */
        color: navy; /* Цвет текста */
        margin-left: 20px; /* Отступ слева */
        border-left: 1px solid navy; /* Граница слева от текста */
        padding-left: 15px; /* Расстояние от линии до текста */
      }
    </style>
  </head>
  <body>
    <p>Для искусственного освещения помещения применяются люминесцентные лампы. Они отличаются высокой световой отдачей, продолжительным сроком службы, малой яркостью светящейся поверхности, близким к естественному спектральным составом излучаемого света, что обеспечивает хорошую цветопередачу.</p>
    <p class="cite">Для исключения засветки экрана дисплея световыми потоками оконные проемы снабжены светорассеивающими шторами.</p>
  </body>
</html>
```





Для искусственного освещения помещения применяются люминесцентные лампы. Они отличаются высокой световой отдачей, продолжительным сроком службы, малой яркостью светящейся поверхности, близким к естественному спектральным составом излучаемого света, что обеспечивает хорошую цветопередачу.

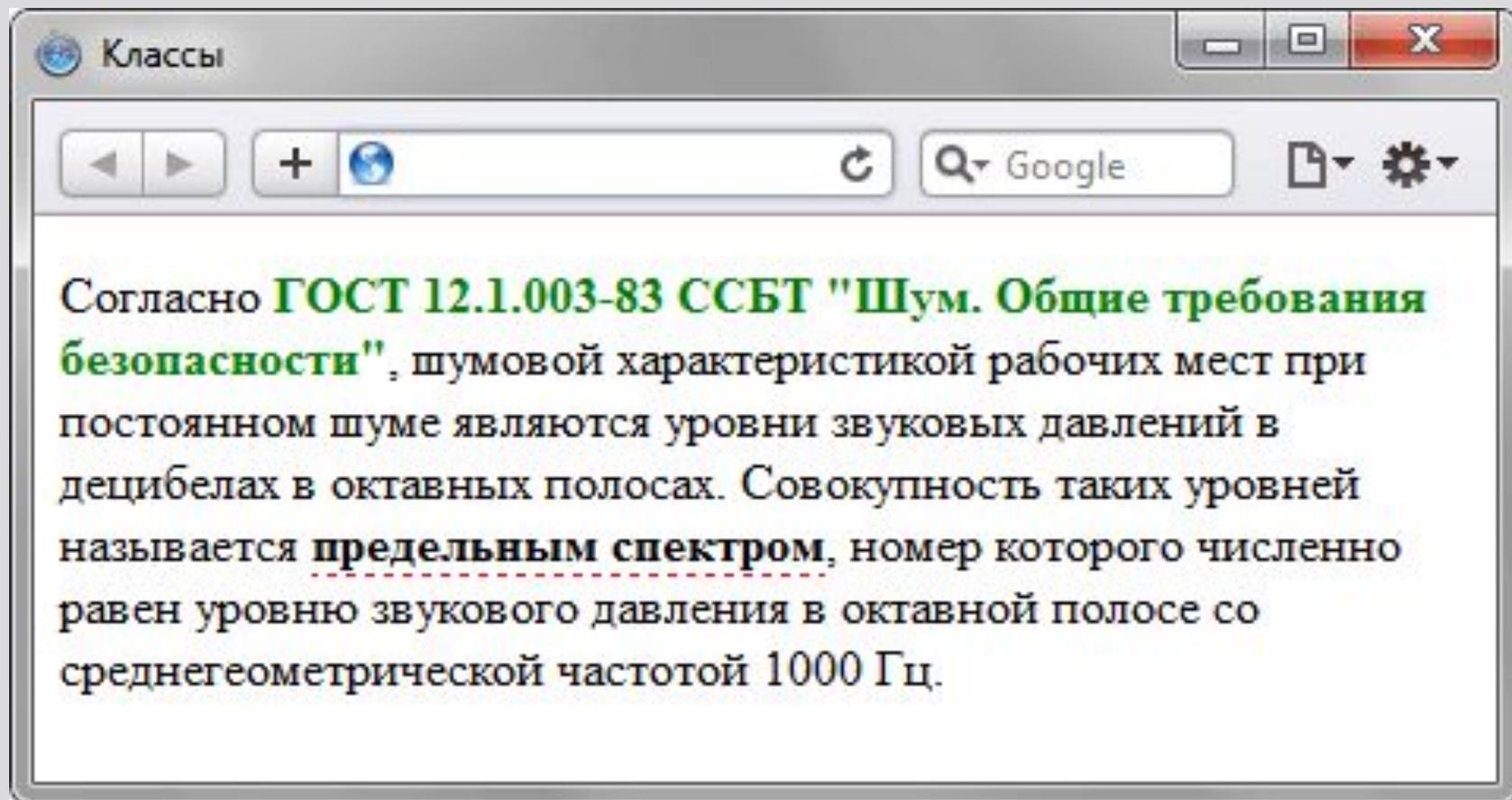
Для исключения засветки экрана дисплея световыми потоками оконные проемы снабжены светорассеивающими шторами.



Использование классов

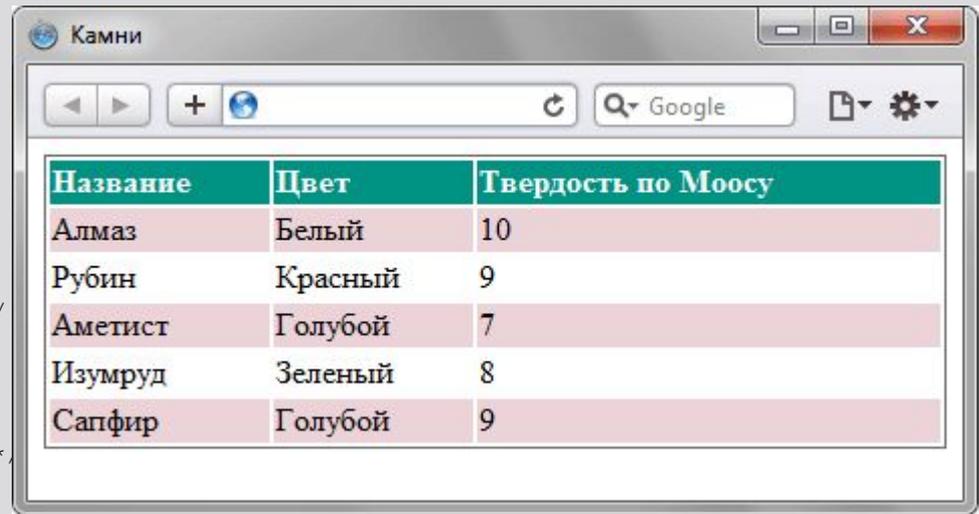
```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Классы</title>
    <style>
      .gost {
        color: green; /* Цвет текста */
        font-weight: bold; /* Жирное начертание */
      }
      .term {
        border-bottom: 1px dashed red; /* Подчеркивание под текстом */
      }
    </style>
  </head>
  <body>
    <p>Согласно <span class="gost">ГОСТ 12.1.003-83 ССБТ &quot;Шум. Общие
    требования безопасности&quot;</span>, шумовой характеристикой рабочих мест
    при постоянном шуме являются уровни звуковых давлений в децибелах в
    октавных полосах. Совокупность таких уровней называется <b
    class="term">предельным спектром</b>, номер которого численно равен уровню
    звукового давления в октавной полосе со среднегеометрической частотой
    1000 Гц.
    </p>
  </body>
</html>
```





Использование классов

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Камни</title>
<style>
table.jewel {
width: 100%; /* Ширина таблицы */
border: 1px solid #666; /* Рамка вокруг таблицы */
}
th {
background: #009383; /* Цвет фона */
color: #fff; /* Цвет текста */
text-align: left; /* Выравнивание по левому краю */
}
tr.odd {
background: #ebd3d7; /* Цвет фона */
}
</style>
</head>
<body>
<table class="jewel">
<tr>
<th>Название</th><th>Цвет</th><th>Твердость по
Моосу</th>
</tr>
<tr class="odd">
<td>Алмаз</td><td>Белый</td><td>10</td>
</tr>
<tr>
<td>Рубин</td><td>Красный</td><td>9</td>
</tr>
<tr class="odd">
<td>Аметист</td><td>Голубой</td><td>7</td>
</tr>
<tr>
<td>Изумруд</td><td>Зеленый</td><td>8</td>
</tr>
<tr class="odd">
<td>Сапфир</td><td>Голубой</td><td>9</td>
</tr>
</table>
</body>
</html>
```



The screenshot shows a web browser window with the title "Камни". The address bar contains "Google". The main content area displays a table with the following data:

Название	Цвет	Твердость по Моосу
Алмаз	Белый	10
Рубин	Красный	9
Аметист	Голубой	7
Изумруд	Зеленый	8
Сапфир	Голубой	9



Одновременное использование разных классов

- К любому тегу одновременно можно добавить несколько классов, перечисляя их в атрибуте class через пробел.
- В этом случае к элементу применяется стиль, описанный в правилах для каждого класса.
- Поскольку при добавлении нескольких классов они могут содержать одинаковые стилевые свойства, но с разными значениями, то берётся значение у класса, который описан в коде ниже.

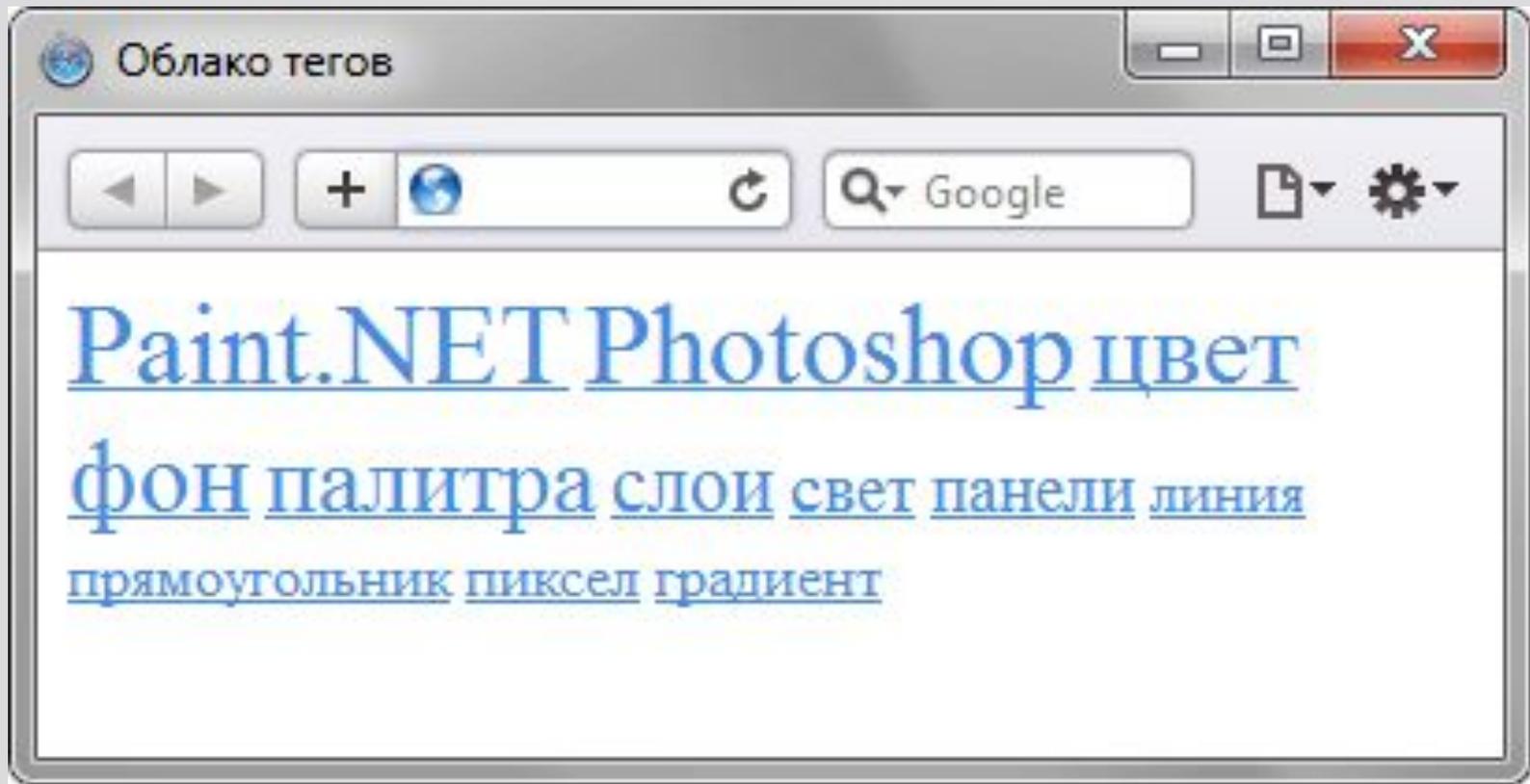


Сочетание разных классов

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Камни</title>
<style>
table.jewel {
width: 100%; /* Ширина таблицы */
border: 1px solid #666; /* Рамка вокруг
таблицы */
}
th {
background: #009383; /* Цвет фона */
color: #fff; /* Цвет текста */
text-align: left; /* Выравнивание по левому
краю */
}
tr.odd {
background: #ebd3d7; /* Цвет фона */
}
</style>
</head>
<body>
```

```
<table class="jewel">
<tr>
<th>Название</th><th>Цвет</th><th>Твердость
по Моосу</th>
</tr>
<tr class="odd">
<td>Алмаз</td><td>Белый</td><td>10</td>
</tr>
<tr>
<td>Рубин</td><td>Красный</td><td>9</td>
</tr>
<tr class="odd">
<td>Аметист</td><td>Голубой</td><td>7</td>
</tr>
<tr>
<td>Изумруд</td><td>Зеленый</td><td>8</td>
</tr>
<tr class="odd">
<td>Сапфир</td><td>Голубой</td><td>9</td>
</tr>
</table>
</body>
</html>
```





ИДЕНТИФИКАТОРЫ



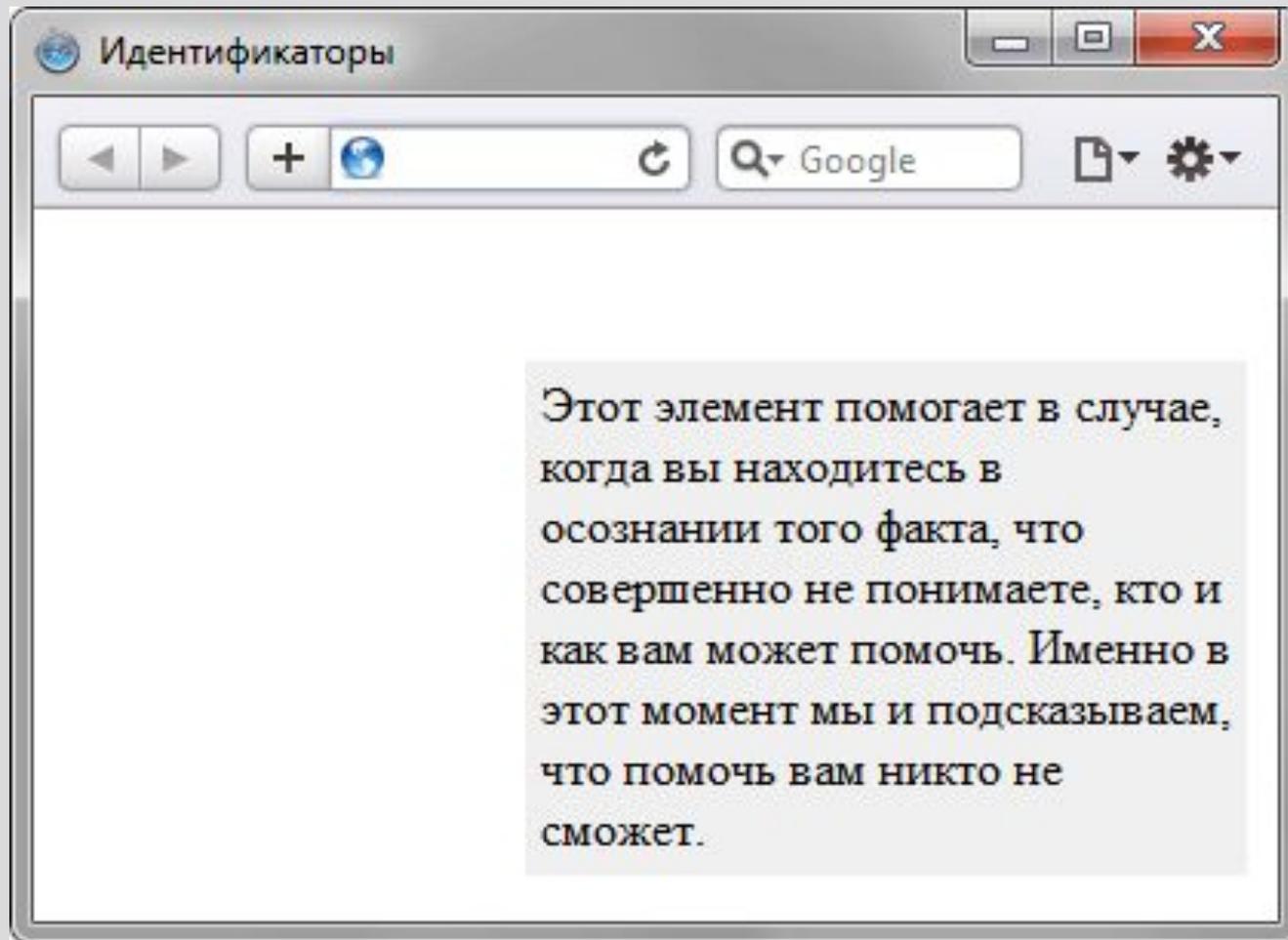
- Идентификатор (называемый также «ID селектор») определяет уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты.
- Синтаксис применения идентификатора следующий.
#Имя идентификатора { свойство1: значение; свойство2: значение; ... }
- При описании идентификатора вначале указывается символ решётки (#), затем идет имя идентификатора.
- Оно должно начинаться с латинского символа и может содержать в себе символ дефиса (-) и подчеркивания (_).
- Использование русских букв в именах идентификатора недопустимо.
- В отличие от классов идентификаторы должны быть уникальны, иными словами, встречаться в коде документа только один раз.
- Обращение к идентификатору происходит аналогично классам, но в качестве ключевого слова у тега используется атрибут id, значением которого выступает имя идентификатора



Использование идентификатора

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Идентификаторы</title>
    <style>
      #help {
        position: absolute; /* Абсолютное позиционирование */
        left: 160px; /* Положение элемента от левого края */
        top: 50px; /* Положение от верхнего края */
        width: 225px; /* Ширина блока */
        padding: 5px; /* Поля вокруг текста */
        background: #f0f0f0; /* Цвет фона */
      }
    </style>
  </head>
  <body>
    <div id="help">
      Этот элемент помогает в случае, когда вы находитесь в осознании того
      факта, что совершенно не понимаете, кто и как вам может помочь. Именно в
      этот момент мы и подсказываем, что помочь вам никто не сможет.
    </div>
  </body>
</html>
```





Идентификатор с тегом

- Как и при использовании классов, идентификаторы можно применять к конкретному тегу.
- Синтаксис при этом будет следующий.

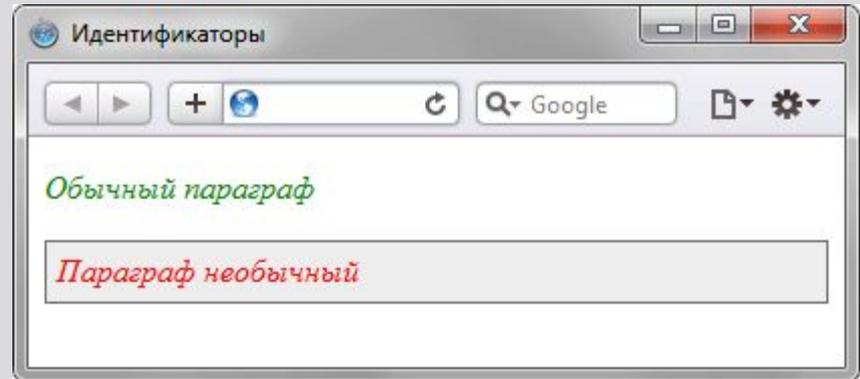
Тег#Имя идентификатора { свойство1: значение; свойство2: значение; ... }

- Вначале указывается имя тега, затем без пробелов символ решётки и название идентификатора.



Идентификатор совместно с тегом

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Идентификаторы</title>
    <style>
      P {
        color: green; /* Зеленый цвет текста */
        font-style: italic; /* Курсивное начертание
текста */
      }
      P#ора {
        color: red; /* Красный цвет текста */
        border: 1px solid #666; /* Параметры рамки */
        background: #eee; /* Цвет фона */
        padding: 5px; /* Поля вокруг текста */
      }
    </style>
  </head>
  <body>
    <p>Обычный параграф</p>
    <p id="ора">Параграф необычный</p>
  </body>
</html>
```



КОНТЕКСТНЫЕ СЕЛЕКТОРЫ



- При создании веб-страницы часто приходится вкладывать одни теги внутрь других.
- Чтобы стили для этих тегов использовались корректно, помогут селекторы, которые работают только в определённом контексте.
- Например, задать стиль для тега **** только когда он располагается внутри контейнера **<p>**.
- Таким образом можно одновременно установить стиль для отдельного тега, а также для тега, который находится внутри другого.
- Контекстный селектор состоит из простых селекторов разделённых пробелом. Так, для селектора тега синтаксис будет следующий.

Тег1 Тег2 { ... }

- В этом случае стиль будет применяться к Тегу2 когда он размещается внутри Тега1, как показано ниже.

<Тег1>

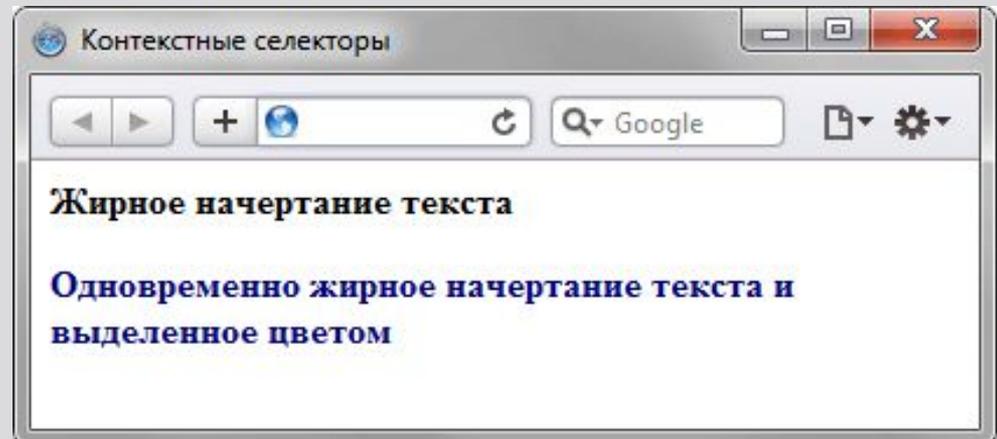
 <Тег2> ... </Тег2>

</Тег1>



Контекстные селекторы

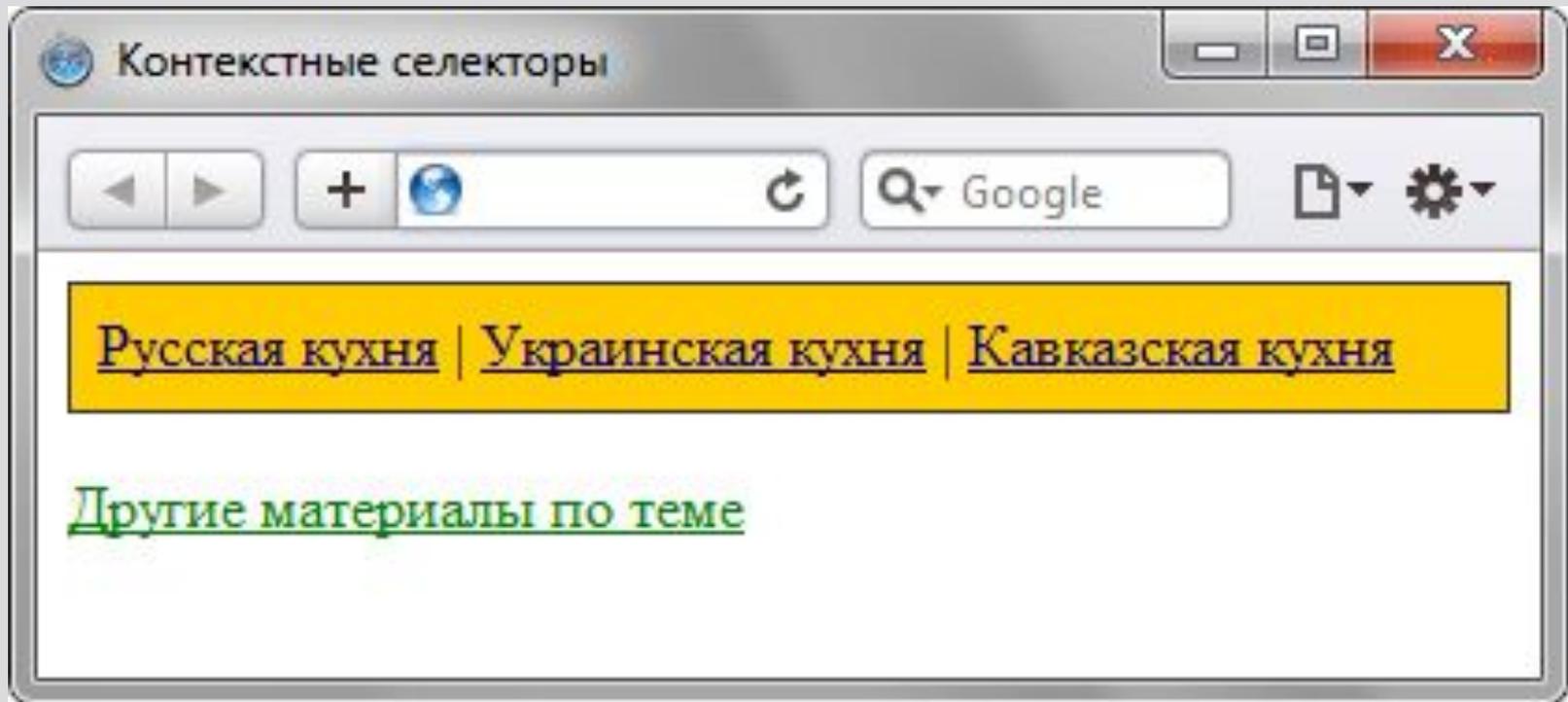
```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Контекстные селекторы</title>
    <style>
      P B {
        font-family: Times, serif; /* Семейство шрифта */
        color: navy; /* Синий цвет текста */
      }
    </style>
  </head>
  <body>
    <div><b>Жирное начертание текста</b></div>
    <p><b>Одновременно жирное начертание текста
и выделенное цветом</b></p>
  </body>
</html>
```



Использование классов

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Контекстные селекторы</title>
    <style>
      A {
        color: green; /* Зеленый цвет текста для всех ссылок */
      }
      .menu {
        padding: 7px; /* Поля вокруг текста */
        border: 1px solid #333; /* Параметры рамки */
        background: #fc0; /* Цвет фона */
      }
      .menu A {
        color: navy; /* Темно-синий цвет ссылок */
      }
    </style>
  </head>
  <body>
    <div class="menu">
      <a href="http://htmlbook.ru/1.html">Русская кухня</a> |
      <a href="http://htmlbook.ru/2.html">Украинская кухня</a> |
      <a href="http://htmlbook.ru/3.html">Кавказская кухня</a>
    </div>
    <p><a href="http://htmlbook.ru/text.html">Другие материалы по теме</a></p>
  </body>
</html>
```





СОСЕДНИЕ СЕЛЕКТОРЫ



- Соседними называются элементы веб-страницы, когда они следуют непосредственно друг за другом в коде документа. Рассмотрим несколько примеров отношения элементов.

```
<p>Lorem ipsum <b>dolor</b> sit amet.</p>
```

- В этом примере тег **** является дочерним по отношению к тегу **<p>**, поскольку он находится внутри этого контейнера. Соответственно **<p>** выступает в качестве родителя ****.

```
<p>Lorem ipsum <b>dolor</b> <var>sit</var> amet.</p>
```

- Здесь теги **<var>** и **** никак не перекрываются и представляют собой соседние элементы. То, что они расположены внутри контейнера **<p>**, никак не влияет на их отношение.

```
<p>Lorem <b>ipsum </b> dolor sit amet, <i>consectetur</i> adipiscing <tt>elit</tt>.</p>
```

- Соседними здесь являются теги **** и **<i>**, а также **<i>** и **<tt>**. При этом **** и **<tt>** к соседним элементам не относятся из-за того, что между ними расположен контейнер **<i>**.
- Для управления стилем соседних элементов используется символ плюса (+), который устанавливается между двумя селекторами. Общий синтаксис следующий.

Селектор 1 + Селектор 2 { Описание правил стиля }

- Пробелы вокруг плюса не обязательны, стиль при такой записи применяется к Селектору 2, но только в том случае, если он является соседним для Селектора 1 и следует сразу после него.



Использование соседних селекторов

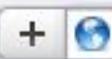
```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Соседние селекторы</title>
    <style>
      B + I {
        color: red; /* Красный цвет текста */
      }
    </style>
  </head>
  <body>
    <p>Lorem <b>ipsum </b> dolor sit amet,
<i>consectetuer</i> adipiscing elit.</p>
    <p>Lorem ipsum dolor sit amet, <i>consectetuer</i>
adipiscing elit.</p>
  </body>
</html>
```



Изменение стиля абзаца

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Изменение стиля абзаца</title>
    <style>
      H2.sic {
        font-size: 140%; /* Размер шрифта */
        color: maroon; /* Цвет текста */
        font-weight: normal; /* Нормальное начертание текста */
        margin-left: 30px; /* Отступ слева */
        margin-bottom: 0px; /* Отступ снизу */
      }
      H2.sic + P {
        background: #ddd; /* Цвет фона */
        margin-left: 30px; /* Отступ слева */
        margin-top: 0.5em; /* Отступ сверху */
        padding: 7px; /* Поля вокруг текста */
      }
    </style>
  </head>
  <body>
    <h1>Методы ловли льва в пустыне</h1>
    <h2>Метод последовательного перебора</h2>
    <p>Пусть лев имеет габаритные размеры  $L \times W \times H$ , где  $L$  - длина льва от кончика носа до кисточки хвоста,  $W$  - ширина льва, а  $H$  - его высота. После чего пустыню разбиваем на ряд элементарных прямоугольников, размер которых совпадает с шириной и длиной льва. Учитывая, что лев может находиться не строго на заданном участке, а одновременно на двух из них, клетку для ловли следует делать повышенной площади, а именно  $2L \times 2W$ . Благодаря этому мы избежим ошибки, когда в клетке окажется пойманным лишь половина льва или, что хуже, только его хвост.</p>
    <h2 class="sic">Важное замечание</h2>
    <p>Для упрощения расчетов хвост в качестве погрешности измерения можно отбросить и не принимать во внимание.</p>
    <p>Далее последовательно накрываем каждый из размеченных прямоугольников пустыни клеткой и проверяем, пойман лев или нет. Как только лев окажется в клетке, процедура поимки считается завершенной.</p>
  </body>
</html>
```





Google



Методы ловли льва в пустыне

Метод последовательного перебора

Пусть лев имеет габаритные размеры $L \times W \times H$, где L - длина льва от кончика носа до кисточки хвоста, W - ширина льва, а H - его высота. После чего пустыню разбиваем на ряд элементарных прямоугольников, размер которых совпадает с шириной и длиной льва. Учитывая, что лев может находиться не строго на заданном участке, а одновременно на двух из них, клетку для ловли следует делать повышенной площади, а именно $2L \times 2W$. Благодаря этому мы избежим ошибки, когда в клетке окажется пойманным лишь половина льва или, что хуже, только его хвост.

Важное замечание

Для упрощения расчетов хвост в качестве погрешности измерения можно отбросить и не принимать во внимание.

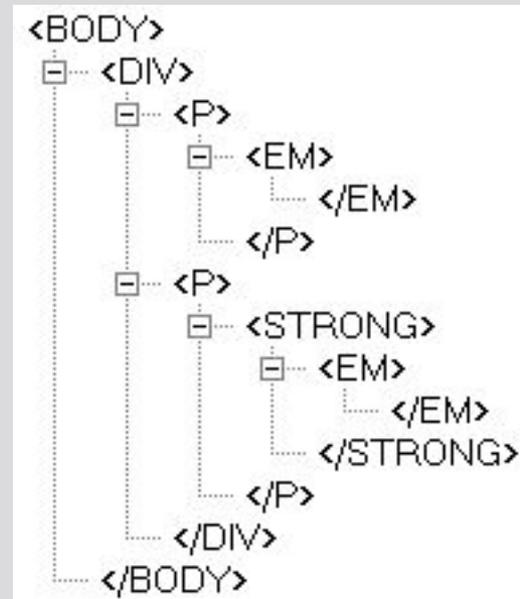


ДОЧЕРНИЕ СЕЛЕКТОРЫ



Вложенность элементов в документе

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Lorem ipsum</title>
  </head>
  <body>
    <div class="main">
      <p><em>Lorem ipsum dolor sit amet</em>,
consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet
dolore magna aliquam erat volutpat.</p>
      <p><strong><em>Ut wisis enim ad minim
veniam</em></strong>, quis nostrud exerci
tution ullamcorper suscipit lobortis nisl
ut aliquip ex ea commodo consequat.</p>
    </div>
  </body>
</html>
```



- Дочерним селектором считается такой, который в дереве элементов находится прямо внутри родительского элемента. Синтаксис применения таких селекторов следующий.

Селектор 1 > Селектор 2 { Описание правил стиля }

- Стиль применяется к Селектору 2, но только в том случае, если он является дочерним для Селектора 1.
- Если обратиться к примеру, то стиль вида

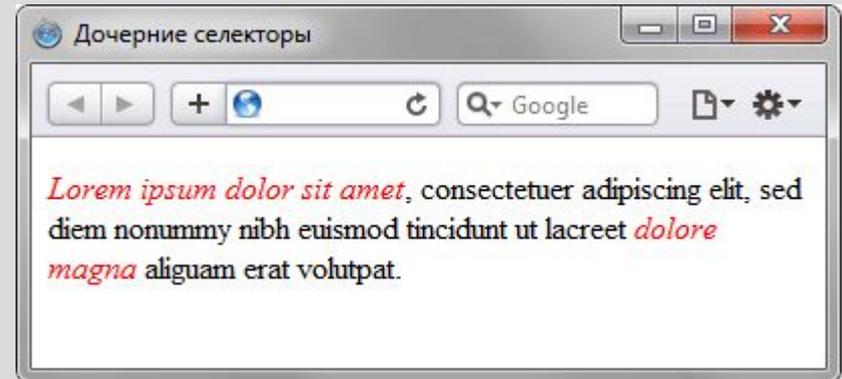
P > EM {color: red}

- будет установлен для первого абзаца документа, поскольку тег **** находится внутри контейнера **<p>**, и не даст никакого результата для второго абзаца. А все из-за того, что тег **** во втором абзаце расположен в контейнере ****, поэтому нарушается условие вложенности.
- По своей логике дочерние селекторы похожи на селекторы контекстные.
- Разница между ними следующая.
- Стиль к дочернему селектору применяется только в том случае, когда он является прямым потомком, иными словами, непосредственно располагается внутри родительского элемента.



Контекстные и дочерние селекторы

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Дочерние селекторы</title>
    <style>
      DIV I { /* Контекстный селектор */
        color: green; /* Зеленый цвет текста */
      }
      P > I { /* Дочерний селектор */
        color: red; /* Красный цвет текста */
      }
    </style>
  </head>
  <body>
    <div>
      <p><i>Lorem ipsum dolor sit amet</i>,
      consectetuer adipiscing elit, sed diem
      nonummy nibh euismod tincidunt ut laoreet
      <i>dolore magna</i> aliquam erat
      volutpat.</p>
    </div>
  </body>
</html>
```



На тег `<i>` в примере действуют одновременно два правила:

- контекстный селектор (тег `<i>` расположен внутри`<div>`) и
- дочерний селектор (тег `<i>` является дочерним по отношению к `<p>`).



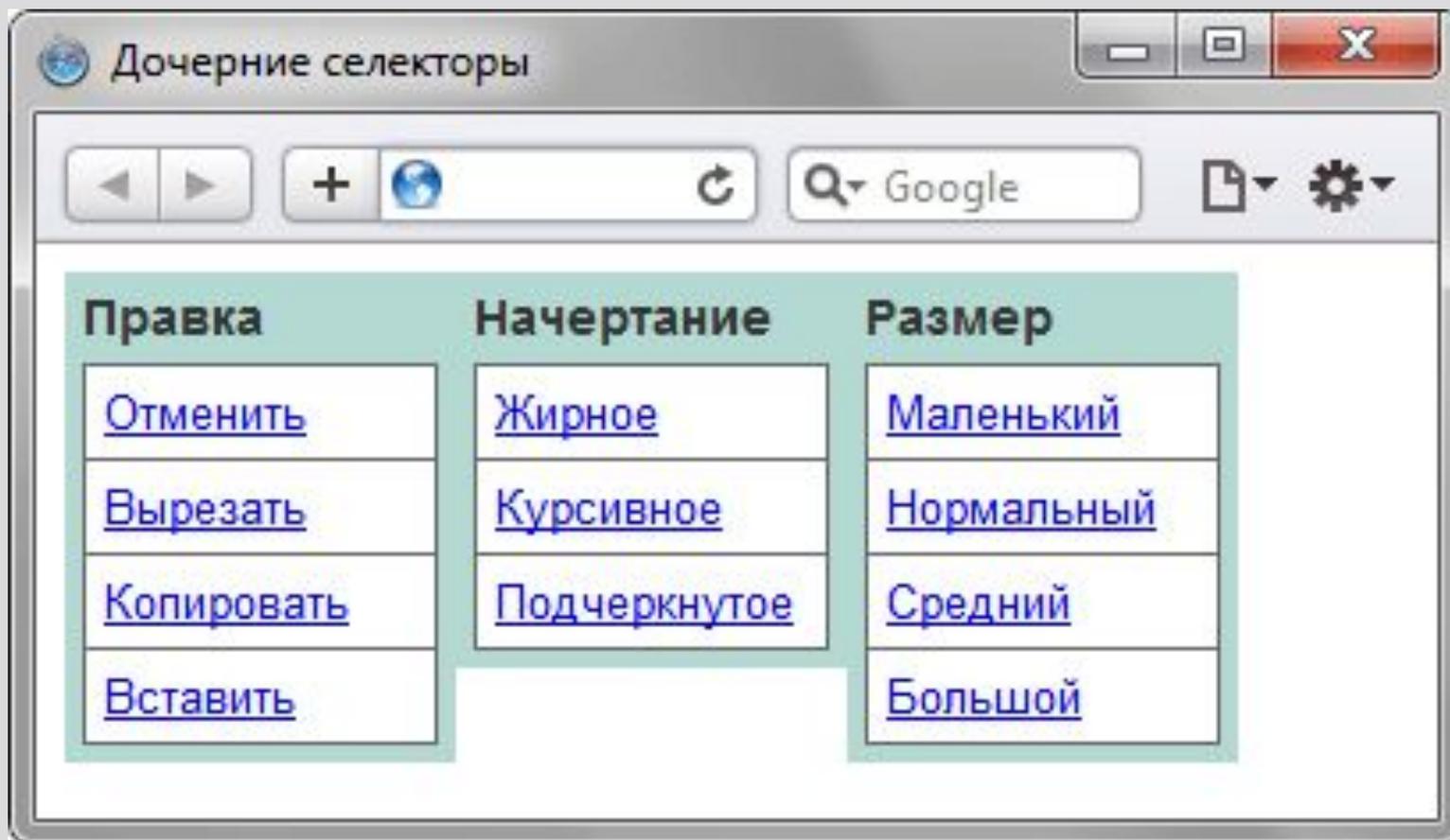
Использование дочерних селекторов

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Дочерние селекторы</title>
<style>
UL#menu {
margin: 0; padding: 0; /* Убираем отступы */
}
UL#menu > LI {
list-style: none; /* Убираем маркеры списка */
width: 100px; /* Ширина элемента в пикселах */
background: #b3d9d2; /* Цвет фона */
color: #333; /* Цвет текста */
padding: 5px; /* Поля вокруг текста */
font-family: Arial, sans-serif; /* Рубленый шрифт */
font-size: 90%; /* Размер шрифта */
font-weight: bold; /* Жирное начертание */
float: left; /* Располагаем элементы по горизонтали */
}
LI > UL {
list-style: none; /* Убираем маркеры списка */
margin: 0; padding: 0; /* Убираем отступы вокруг элементов
списка */
border-bottom: 1px solid #666; /* Граница внизу */
padding-top: 5px; /* Добавляем отступ сверху */
}
LI > A {
display: block; /* Ссылки отображаются в виде блока */
font-weight: normal; /* Нормальное начертание текста */
font-size: 90%; /* Размер шрифта */
background: #fff; /* Цвет фона */
border: 1px solid #666; /* Параметры рамки */
border-bottom: none; /* Убираем границу снизу */
padding: 5px; /* Поля вокруг текста */
}
```

```

</style>
</head>
<body>
<ul id="menu">
<li>Правка
<ul>
<li><a href="#">Отменить</a></li>
<li><a href="#">Вырезать</a></li>
<li><a href="#">Копировать</a></li>
<li><a href="#">Вставить</a></li>
</ul>
</li>
<li>Начертание
<ul>
<li><a href="#">Жирное</a></li>
<li><a href="#">Курсивное</a></li>
<li><a href="#">Подчеркнутое</a></li>
</ul>
</li>
<li>Размер
<ul>
<li><a href="#">Маленький</a></li>
<li><a href="#">Нормальный</a></li>
<li><a href="#">Средний</a></li>
<li><a href="#">Большой</a></li>
</ul>
</li>
</ul>
</body>
</html>
```





Спасибо за внимание!

