

Алексей Владимирович Усков

Web-технологии

www.uskov.info (Курсы по Web)

uskov@iee.unn.ru

<http://vk.com/alexeyuskov>

HTML5

(Hyper Text Markup Language)

HTML5 это не новая версия старого языка разметки и даже не усовершенствованный вариант этой уже перешедшей в разряд «древних» технологии. HTML5 - совершенно новая концепция построения веб-сайтов и приложений

HTML5 считается результатом объединения HTML, CSS и JavaScript.

В HTML5 реализовано множество новых синтаксических особенностей (<video>, <audio> и <canvas> ...)

Элементы <section>, <article>, <header> и <nav>, разработаны для того, чтобы обогащать семантическое содержимое документа.

CSS - набор правил форматирования, позволяющий менять внешний вид документа, например размер и цвет текста.

CSS3 (Cascading Style Sheets 3 — каскадные таблицы стилей третьего поколения). Главной особенностью CSS3 является возможность создавать анимированные элементы без использования JS, поддержка линейных и радиальных градиентов, теней, сглаживания и многое другое.

СТРУКТУРА HTML ДОКУМЕНТА. ТЕГИ

<body>

С тегом <body>* вы уже встречались в первом примере. Все, помещаемое внутрь этого элемента, отображается в основном окне браузера.

<head>

глава-01/body-head-title.html

HT

```
<html>
  <head>
    <title>Это заголовок страницы</title>
  </head>
  <body>
    <h1>Это "тело" страницы</h1>
    <p>Содержимое, помещенное в "тело" страницы,
      будет отображено в основном окне браузера.</p>
  </body>
</html>
```

ЗАГОЛОВКИ

`<h1>`

`<h2>`

`<h3>`

`<h4>`

`<h5>`

`<h6>`

В языке HTML существует
шесть уровней заголовков:

глава-02/headings.html

```
<h1>Основной заголовок</h1>
```

```
<h2>Заголовок 2-го уровня</h2>
```

```
<h3>Заголовок 3-го уровня</h3>
```

```
<h4>Заголовок 4-го уровня</h4>
```

```
<h5>Заголовок 5-го уровня</h5>
```

```
<h6>Заголовок 6-го уровня</h6>
```


Упорядоченные списки создаются при помощи элемента .

Каждый элемент списка помещается между открывающим тегом и закрывающим тегом .

По умолчанию браузеры автоматически расставляют нужные отступы для элементов списков.

```
<ol>
  <li>Порежьте картофель кубиками.</li>
  <li>Варите картофель в соленой воде на медленном
      огне в течение 15-20 минут до размягчения.</li>
  <li>Подогрейте и смешайте молоко, сливочное масло
      и мускатный орех.</li>
  <li>Слейте воду из кастрюли и потолките
      картофель.</li>
  <li>Перемешайте картофель со смесью из молока,
      масла и мускатного ореха.</li>
</ol>
```

РЕЗУЛЬТАТ

1. Порежьте картофель кубиками.
2. Варите картофель в соленой воде на медленном огне в течение 15-20 минут до размягчения.
3. Подогрейте и смешайте молоко, сливочное масло и мускатный орех.
4. Слейте воду из кастрюли и потолките картофель.
5. Перемешайте картофель со смесью из молока, масла и мускатного ореха.

```
<ul>
  <li>1 кг картофеля</li>
  <li>100 мл молока</li>
  <li>50 г соленого масла</li>
  <li>Тертый мускатный орех</li>
  <li>Соль и перец по вкусу</li>
</ul>
```

Неупорядоченные списки создаются при помощи элемента ``.

Каждый элемент списка помещается между открывающим тегом `` и закрывающим тегом ``.

По умолчанию браузеры автоматически расставляют нужные отступы для элементов списков.

Иногда при просмотре существующих страниц вы будете встречать аргумент `type` в элементе

РЕЗУЛЬТАТ

- 1 кг картофеля
- 100 мл молока
- 50 г соленого масла
- Тертый мускатный орех
- Соль и перец по вкусу

ВЛОЖЕННЫЕ СПИСКИ

HTML

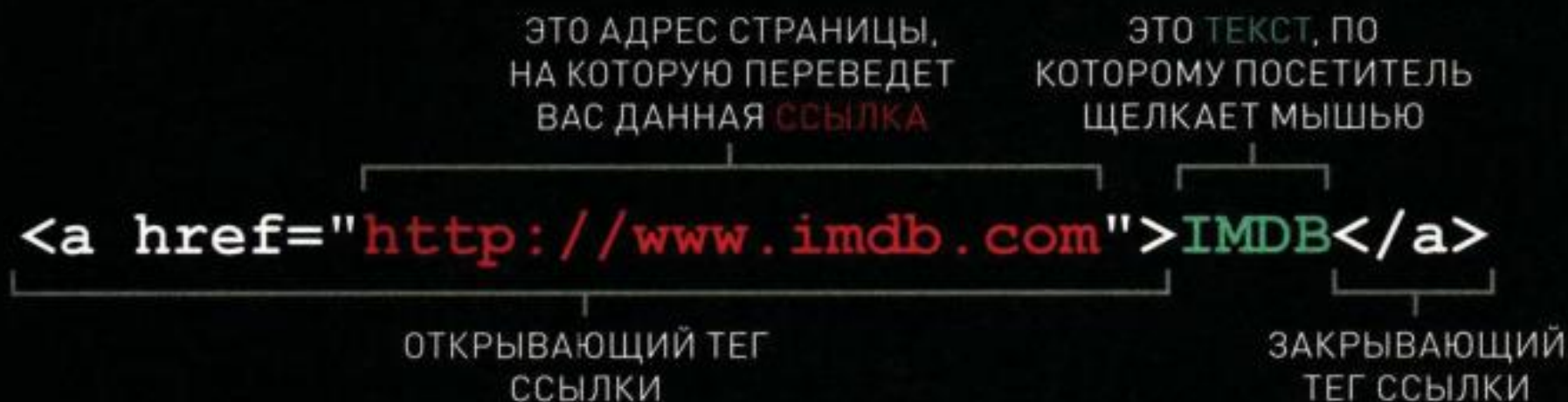
глава-03/nested-lists.html

```
<ul>
  <li>Муссы</li>
  <li>Пирожные
    <ul>
      <li>Круассаны</li>
      <li>Наполеон</li>
      <li>Чизкейки</li>
      <li>Профитроли</li>
    </ul>
  </li>
  <li>Торты</li>
</ul>
```

Внутри элемента `` допустимо создание вложенного списка, или списка второго уровня.

Браузеры отображают вложенные списки с большим отступом по сравнению с основным, или родительским, списком. Кроме того, браузер, как правило, также изменяет вид маркеров вложенного неупорядоченного списка.

Ссылки создаются с помощью элемента `<a>`. Пользователи могут щелкнуть по любому слову текста, помещенного между открывающим `<a>` и закрывающим `` тегами элемента. Используя атрибут `href`, вы можете указать, на какую страницу будет перенаправлен пользователь при щелчке мышью по данной ссылке.




```
<p>
  <ul>
    <li><a href="index.html">На главную</a></li>
    <li><a href="about-us.html">О нас</a></li>
    <li><a href="movies.html">Фильмы</a></li>
    <li><a href="contact.html">Обратная связь</a></li>
  </ul>
</p>
```

РЕЗУЛЬТАТ

- [На главную](#)
- [О нас](#)
- [Фильмы](#)
- [Обратная связь](#)

При создании ссылок на другие страницы вашего сайта нет необходимости указывать в URL-адресе его доменное имя. Вместо этого вы можете воспользоваться сокращенным вариантом, называемым также **относительным URL-адресом**.

Если все страницы сайта находятся в одной и той же папке, то значением атрибута href должно быть лишь имя файла страницы.

Если страницы вашего сайта распределены по разным папкам, то вам придется воспользоваться несколько более сложным синтаксисом для указания местоположения нужной страницы

ОТКРЫТИЕ ССЫЛОК В НОВОМ ОКНЕ

HTML

глава-04/opening-links-in-a-new-window.html

```
<a href="http://www.imdb.com" target="_blank">  
База данных фильмов во Всемирной паутине</a>  
(открывается в новом окне)
```

РЕЗУЛЬТАТ

[База данных фильмов во Всемирной паутине](http://www.imdb.com) (открывается в новом окне)

target

Чтобы создать ссылку, открывающую страницу в новом окне браузера, вы можете воспользоваться атрибутом `target` открывающего тега элемента `<a>`. Значение этого атрибута должно быть установлено в `_blank`.

Чаще всего ссылки данного вида ставятся, когда щелчок по ним должен вести к переходу на сторонний сайт. В этом случае создатели сайтов надеются, что посетитель вернется на их сайт по завершении

ССЫЛКА НА ОПРЕДЕЛЕННУЮ ЧАСТЬ СТРАНИЦЫ

В верхней части довольно длинной страницы можно поместить что-то наподобие оглавления: перечень ссылок, указывающих на соответствующие разделы, расположенные ниже. Кроме того, вы также можете поместить ссылку на верхнюю часть страницы где-нибудь в середине или в нижней части страницы, избавив посетителя от необходимости вращать колесико мышки, чтобы добраться до верха. Прежде чем создавать ссылку на определенную часть страницы, сначала нужно отметить те участки страницы, на которые будут указывать ссылки. Это

глава-04/linking-to-a-specific-part.html

HTML

```
<h1 id="top">Термины, используемые при съемке
    фильмов</h1>
<a href="#zoom">Трансфокация</a><br />
<a href="#interlude">Интерлюдия</a><br />
<a href="#prologue">Пролог</a><br />
<h2 id="zoom">Трансфокация</h2>
<p>Плавный наезд на объект съемки или отъезд от него
    камерой.</p>
<h2 id="interlude">Интерлюдия</h2>
<p>Короткая вводная сцена или последовательность
    сцен, появляющаяся в фильме, но не обязательно
    связанная с основным сюжетом.</p>
<h2 id="prologue">Пролог</h2>
<p>Речь, введение или короткая сцена, предшествующая
    основному действию сюжета фильма; антоним
    эпилогу.</p>
<p><a href="#top">Наверх</a></p>
```


РАЗМЕЩЕНИЕ ИЗОБРАЖЕНИЙ НА САЙТЕ

Если вы создаете сайт с нуля, то полезно создать отдельную папку для хранения всех используемых на нем изображений.

ДОБАВЛЕНИЕ ИЗОБРАЖЕНИЙ

Для добавления изображения на страницу воспользуйтесь элементом ``. Он является пустым (то есть без закрывающего тега). Для него должны быть указаны следующие два атрибута.

src

Данный атрибут сообщает браузеру расположение нужного изображения. Как правило, его значением будет относительный URL-адрес, указывающий на изображение, хранящееся в директориях вашего собственного сайта. (В данном примере все изображения расположены в подпапке *images*. Об относительных URL-адресах речь шла в предыдущей главе.)

глава-05/adding-images.html

HTML

```

```

РЕЗУЛЬТАТ



ВЫСОТА И ШИРИНА ИЗОБРАЖЕНИЙ

HTML

глава-05/height-and-width-of-images.html

```

```

РЕЗУЛЬТАТ



Зачастую вы будете видеть элемент `` с еще двумя атрибутами, определяющими его размер:

height

Данный атрибут устанавливает высоту изображения в пикселах.

width

Данный атрибут устанавливает ширину изображения в пикселах. Как правило, время загрузки изображений значительно дольше, чем HTML-кода, составляющего всю остальную

1

СОХРАНЯЙТЕ ИЗОБРАЖЕНИЯ В ПРАВИЛЬНОМ ФОРМАТЕ.

В основном на сайтах используется графика трех файловых форматов: JPEG, GIF и PNG. При использовании других форматов изображение может потерять резкость, замедлить скорость загрузки страницы или вовсе не отобразиться на компьютере посетителя.

2

СОХРАНЯЙТЕ ИЗОБРАЖЕНИЯ С ПРАВИЛЬНЫМИ ЗНАЧЕНИЯМИ ШИРИНЫ И ВЫСОТЫ.

Сохраняйте изображения с теми же значениями ширины и высоты, с какими они должны быть отображены на сайте. Если значения ширины и/или высоты изображения меньше указанных вами в коде, это может привести к растягиванию и искажению рисунка. Если ширина и/или высота изображения больше, чем вы указали в коде, то для его отображения браузеру потребуется больше времени.

3

ИСПОЛЬЗУЙТЕ ПРАВИЛЬНОЕ РАЗРЕШЕНИЕ.

Экраны компьютеров состоят из точек, называемых пикселями. Изображения, опубликованные во Всемирной паутине, также состоят из очень маленьких точек. Разрешением называется количество точек, помещающихся на одном дюйме. Большинство мониторов отображают сайты с разрешением 72 пиксела на дюйм. Сохранение рисунка с большим разрешением приведет к увеличению его размеров, результатом чего, в свою очередь, будет более низкая скорость загрузки.

```
<figure>
  
  <img />
  <br />
  <figcaption>Во время сна выдры держат друг
    друга за лапки, чтобы их не разнесло
    течением.</figcaption>
</figure>
```

РЕЗУЛЬТАТ



Зачастую изображения сопровождаются подписями.

В спецификации HTML5 появился новый элемент `<figure>`, объединяющий изображение и подпись к нему, таким образом текст и рисунок оказываются связанными друг с другом.

В один элемент `<figure>` можно включить несколько изображений, в случае если для них используется одна подпись.

<figcaption>

Элемент `<figcaption>` был включен в язык HTML5, чтобы позволить веб-дизайнерам добавлять

БАЗОВАЯ СТРУКТУРА ТАБЛИЦЫ

<table>

Для создания таблиц используется элемент <table>. Контент таблицы описывается построчно.

<tr>

Открывающий тег <tr>* обозначает начало новой строки таблицы.

После него помещаются элементы <td>, каждый из которых соответствует отдельной ячейке в этой строке.

Конец строки обозначается закрывающим тегом </tr>.

глава-06/basic-table-structure.html

HTML

```
<table>
  <tr>
    <td>15</td>
    <td>15</td>
    <td>30</td>
  </tr>
  <tr>
    <td>45</td>
    <td>60</td>
    <td>45</td>
  </tr>
  <tr>
    <td>60</td>
    <td>90</td>
    <td>90</td>
  </tr>
</table>
```


ЗАГОЛОВКИ ТАБЛИЦ

HTML

глава-06/table-headings.html

<th>

```
<table>
  <tr>
    <th></th>
    <th scope="col">Суббота</th>
    <th scope="col">Воскресенье</th>
  </tr>
  <tr>
    <th scope="row">Продано билетов</th>
    <td>120</td>
    <td>135</td>
  </tr>
  <tr>
    <th scope="row">Выручка</th>
    <td>18 000 руб.</td>
    <td>20 250 руб.</td>
  </tr>
</table>
```

Элемент `<th>*` используется точно так же, как и элемент `<td>`, однако его назначение — создание заголовка строки или столбца.

Помните, что даже если ячейка не содержит данных, то вы все равно должны представить ее пустым элементом `<td>` или `<th>`, в противном случае таблица будет некорректно отображена в браузере. (В примере первая ячейка первой строки пустая.) Употребление элемента `<th>` помогает людям, использующим программы экранного доступа, улучшает результативность индексирования таблиц поисковыми системами, а еще этот элемент помогает управлять оформлением таблиц при использовании CSS.

ОБЪЕДИНЕНИЕ СТОЛБЦОВ

Иногда вам может понадобиться, чтобы ячейка таблицы занимала не один, а несколько столбцов.

Атрибут `colspan` может быть использован с элементами `<th>` и `<td>` для обозначения, сколько столбцов должна занимать ячейка.

В примере справа вы можете видеть расписание, состоящее из пяти столбцов: первый содержит заголовок строки (день недели), оставшиеся четыре представляют собой часы.

Если вы посмотрите на код ячейки с текстом «География», то увидите, что значение атрибута `colspan` равно 2, это значит, что

глава-06/spanning-columns.html

HTML

```
<table>
  <tr>
    <th></th>
    <th>9:00</th>
    <th>10:00</th>
    <th>11:00</th>
    <th>12:00</th>
  </tr>
  <tr>
    <th>Понедельник</th>
    <td colspan="2">География</td>
    <td>Музыка</td>
    <td>ИЗО</td>
  </tr>
  <tr>
    <th>Вторник</th>
    <td colspan="3">Физкультура</td>
    <td>География</td>
```


ОБЪЕДИНЕНИЕ СТРОК

HTML

глава-06/spanning-rows.html

```
<table>
  <tr>
    <th></th>
    <th>TV1000</th>
    <th>THT</th>
    <th>CTC</th>
  </tr>
  <tr>
    <th>20:00-21:00</th>
    <td rowspan="2">Фильм</td>
    <td>Ток-шоу</td>
    <td>Юмор</td>
  </tr>
  <tr>
    <th>21:00-22:00</th>
    <td>Ток-шоу</td>
    <td>Мультфильм</td>
  </tr>
</table>
```

Кроме того, вам может понадобиться сделать так, чтобы одна ячейка занимала несколько строк.

Атрибут `rowspan` может быть использован с элементами `<th>` и `<td>` для обозначения, на сколько строк должна простирается ячейка. В примере слева вы можете видеть, что телеканал TV1000 показывает фильм с 18:00 до 20:00, а в это время на каналах ТНТ и СТС транслируются по две развлекательные телепередачи (каждая из которых длится один час). Если вы посмотрите на последний элемент `<tr>`, то увидите, что

ШИРИНА И ПРОМЕЖУТОК

Существует несколько устаревших атрибутов, вытесненных каскадными таблицами стилей (CSS). Использование этих атрибутов не рекомендуется, однако вы можете встретить их при просмотре исходного кода старых сайтов, потому мы решили все же коротко рассказать о них.

Атрибут `width*` используется в открывающем теге `<table>` для указания ширины таблицы. Кроме того, его можно ставить в открывающих тегах элементов `<th>` и `<td>` для задания ширины отдельных ячеек таблицы. Значение этого атрибута — ширина таблицы или ячейки, указанная в пикселах.

Так как столбцы таблицы должны быть равными, то чаще всего вы встретите данный атрибут только в первой строке таблицы (все последующие строки наследуют это значение).

В открывающем теге элемента `<table>` также может быть указан атрибут `cellpadding`,

глава-06/width-and-spacing.html

HTML

```
<table width="400" cellpadding="10"
cellspacing="5">
```

```
<tr>
```

```
<th width="150"></th>
```

```
<th>Снято с карты</th>
```

```
<th>Кредит</th>
```

```
<th width="150">Баланс</th>
```

```
</tr>
```

```
<tr>
```

```
<th>Январь</th>
```

```
<td>2500,00</td>
```

```
<td>6600,50</td>
```

```
<td>4100,50</td>
```

```
</tr>
```

```
<tr>
```

```
<th>Февраль</th>
```

```
<td>1350,50</td>
```

```
<td>8950,20</td>
```

```
<td>11700,10</td>
```

```
</tr>
```

```
</table>
```

АТРИБУТ ID

Каждый HTML-элемент может иметь атрибут `id`. Он используется для идентификации, поэтому его значение должно быть уникальным для каждого элемента на странице. Значение атрибута может начинаться с латинской буквы или со знака подчеркивания (но не с цифры или другого символа). Очень важно, чтобы значения атрибута `id` не повторялись на странице (в противном случае они не будут уникальными). Как вы увидите при изучении каскадных таблиц стилей,

глава-08/id-attribute.html

HTML

```
<p>Вода и воздух. Как обыденны и банальны эти две  
субстанции: они едва ли привлекают к себе чье-то  
внимание, но, тем не менее, они обеспечивают наше  
существование.</p>
```

```
<p id="pullquote">Каждый раз, смотря на море, во  
мне рождается успокаивающее чувство защищенности,  
подобно тому, что появляется во время посещения  
родительского дома; я словно отправляюсь в морское  
путешествие, а это путешествие есть  
наблюдение.</p>
```

```
<p>Самая тайная тайна: вода и воздух здесь, рядом  
с нами: в море.</p>
```


АТТРИБУТ CLASS

HTML

глава-08/class-attribute.html

```
<p class="vazhno">В течение года начиная с 15  
января в Музее современного искусства будет  
проводиться цикл из четырех выставок Хироши  
Сугимото.</p>
```

```
<p>Каждая из четырех выставок будет представлять  
одну из четырех тем в работах художника: "Наука",  
"Архитектура", "История" и "Религия". Таким  
образом посетители смогут увидеть полную панораму  
творчества Хироши Сугимото.</p>
```

```
<p class="vazhno chasy_raboty">Часы работы:  
10:00-18:00 (после 17:30 посетители в музей не  
допускаются).</p>
```

В языке HTML элементу также может быть присвоен атрибут `class`. Иногда вместо идентификации каждого элемента разметки на странице гораздо удобнее оказывается выделить группу элементов. Например введенный вами текст может содержать несколько абзацев, более важных, чем другие, или вы вдруг захотите выделить особым стилем ссылки, ведущие на страницы других сайтов.

ГРУППИРОВКА ТЕКСТА И ЭЛЕМЕНТОВ В БЛОК

Элемент `<div>` позволяет сгруппировать несколько элементов в единый блок. Например, вы можете создать `<div>`, объединяющий в себе все элементы заголовка сайта (логотип и панель навигации). Вы также можете создать `<div>` для объединения комментариев посетителей сайта. В браузере содержимое элемента `<div>` будет отображено с новой строки, никаких других изменений внешнего вида не произойдет.

глава-08/grouping-block-elements.html

HTML

```
<div id="header">
  
  <ul>
    <li><a href="index.html">Главная</a></li>
    <li><a href="biography.html">Биография</a></li>
    <li><a href="works.html">Дискография</a></li>
    <li><a href="contact.html">Контакты</a></li>
  </ul>
</div> <!-- конец заголовка -->
```

РЕЗУЛЬТАТ



HTML5: ДОБАВЛЕНИЕ ВИДЕОКОНТЕНТА НА СТРАНИЦУ

HTML

глава-09/adding-html5-video.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML5: добавление видео на страницу</title>
  </head>
  <body>
    <video src="video/puppy.mp4"
      poster="images/puppy.jpg"
      width="400" height="300"
      preload
      controls
      loop>
      <p>Щенок, играющий в снегу</p>
    </video>
  </body>
</html>
```

<video>

Элемент <video> обладает целым набором атрибутов, позволяющих управлять воспроизведением видеоролика.

src

Данный атрибут позволяет указать расположение нужного видеофайла. (Видеоролик для этого примера был сохранен в формате H.264 и потому может быть воспроизведен только в браузерах Internet Explorer и Safari.)

poster

Данный атрибут позволяет установить изображение, показываемое в процессе загрузки видеоконтента и до момента,

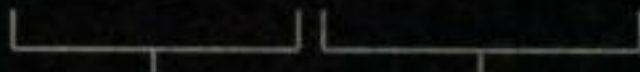
ВВЕДЕНИЕ В CSS

- ▶ Назначение каскадных таблиц стилей
- ▶ Принцип работы каскадных таблиц стилей
- ▶ Правила, свойства и значения

СВОЙСТВА CSS ОПРЕДЕЛЯЮТ ВНЕШНИЙ ВИД ЭЛЕМЕНТОВ

Объявления CSS заключаются в фигурные скобки и состоят из двух частей: **свойства** и **значения**, разделенных двоеточием. В одном объявлении вы можете указать несколько свойств, отделив их друг от друга точкой с запятой.

```
h1, h2, h3 {  
    font-family: Arial;  
    color: yellow;  
}
```



ВНУТРЕННИЕ КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ

HTML

+

CSS

глава-10/using-internal-css.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Использование внутренней каскадной
      таблицы стилей</title>
    <style type="text/css">
      body {
        font-family: arial;
        background-color: rgb(185,179,175);}
      h1 {
        color: rgb(255,255,255);}
    </style>
  </head>
  <body>
    <h1>Картофель</h1>
    <p> Существуют десятки сортов картофеля.
      Обычно они подразделяются на ранние,
```

<style>

Вы можете включить правила CSS в HTML-страницу, поместив их внутри элемента `<style>`, который обычно находится в элементе `<head>`.

Элемент `<style>` должен использовать атрибут `type` для указания, что стили описаны в CSS. Нужно указать значение `text/css`.

При создании сайта, состоящего более чем из одной страницы, вам следует использовать внешнюю таблицу стилей, поскольку это:

Это пустой элемент (он не нуждается в наличии закрывающего тега), который располагается внутри элемента `<head>`. Элемент `<link>` должен использовать три следующих атрибута.

href

Указывает путь к файлу CSS, который часто расположен в папке с именем `css` или `styles`.

type

Данный атрибут определяет тип документа, на который указывает ссылка. Должно быть указано значение `text/css`.

rel

Определяет отношение между HTML-страницей и связанным файлом. При создании ссылки на файл CSS должно быть указано значение `stylesheet`.

```
<link href="css/styles.css" type="text/css"
      rel="stylesheet" />
</head>
<body>
  <h1>Картофель</h1>
  <p> Существуют десятки сортов картофеля.
      Обычно они подразделяются на ранние,
      среднеранние и поздние.</p>
</body>
</html>
```

глава-10/styles.css

CSS

```
body {
  font-family: arial;
  background-color: rgb(185,179,175); }
h1 {
  color: rgb(255,255,255); }
```

РЕЗУЛЬТАТ

Картофель

Существуют десятки сортов картофеля. Обычно они подразделяются на ранние, среднеранние и поздние.

SELECTOR	MEANING	EXAMPLE
УНИВЕРСАЛЬНЫЙ СЕЛЕКТОР	Применяется ко всем элементам в документе	<code>* {}</code> Все элементы страницы
СЕЛЕКТОР ТИПА	Соответствует именам элементов	<code>h1, h2, h3 {}</code> Элементы <code><h1></code> , <code><h2></code> и <code><h3></code>
СЕЛЕКТОР КЛАССА	Соответствует элементу, чей атрибут <code>class</code> имеет значение, совпадающее с указанным после точки	<code>.note {}</code> Элемент, чей атрибут <code>class</code> имеет значение <code>note</code> <code>p.note {}</code> Только элементы <code><p></code> , чьи атрибуты <code>class</code> имеют значение <code>note</code>
СЕЛЕКТОР ИДЕНТИФИКАТОРА	Соответствует элементу, чей атрибут <code>id</code> имеет значение, совпадающее с указанным после символа <code>#</code>	<code>#introduction {}</code> Элемент, чей атрибут <code>id</code> имеет значение <code>introduction</code>
ДОЧЕРНИЙ СЕЛЕКТОР	Соответствует элементу, являющемуся прямым потомком другого элемента	<code>li>a {}</code> Элементы <code><a></code> , являющиеся прямыми потомками элемента <code></code> , но не остальные элементы <code><a></code> на странице
СЕЛЕКТОР ПОТОМКА	Соответствует элементу, являющемуся потомком другого элемента (не обязательно прямым)	<code>p a {}</code> Любые элементы <code><a></code> , располагающиеся внутри <code><p></code> , даже в том случае, когда между ними находятся другие элементы

ПРИНЦИПЫ КАСКАДИРОВАНИЯ ПРАВИЛ CSS

Если к одному и тому же элементу применяется два или более правил, важно понимать, которое из них получит приоритет.

ПОСЛЕДНЕЕ ПРАВИЛО

При наличии двух одинаковых селекторов приоритет получает последний. В примере вы можете видеть, как второй селектор `i` получает приоритет над первым.

СПЕЦИФИЧНОСТЬ

Если один селектор является более специфичным, чем остальные, то он получает приоритет. В данном примере: `h1` является более специфичным, чем `*`;

`p b` является более специфичным, чем `p`;

`p#intro` является более специфичным, чем `p`.

ВАЖНОСТЬ

Вы можете добавить текст `!important` после значения любого свойства, чтобы указать, что его следует считать самым важным из правил, относящихся к одному и тому же элементу.

глава-10/cascade.html

HTML

```
<h1>Картофель</h1>
<p id="intro">Существуют <i>десятки</i> сортов
  <b>картофеля</b>.</p>
<p>Обычно они подразделяются на ранние, среднеранние
  и поздние.</p>
```

CSS

```
* {
  font-family: Arial, Verdana, sans-serif;}
h1 {
  font-family: "Courier New", monospace;}
i {
  color: green;}
i {
  color: red;}
b {
  color: pink;}
p b {
  color: blue !important;}
p b {
  color: violet;}
p#intro {
  font-size: 100%;}
p {
  font-size: 75%;}
```


Цвет

```
/* имя цвета */  
h1 {  
    color: DarkCyan;}  
/* шестнадцатеричный код */  
h2 {  
    color: #ee3e80;}  
/* значение rgb */  
p {  
    color: rgb(100,100,90);}
```

Цвет фона

```
body {  
    background-color: rgb(200,200,200);  
h1 {  
    background-color: DarkCyan;  
h2 {  
    background-color: #ee3e80;  
p {  
    background-color: white;
```


Прозрачность

```
p.one {  
  background-color: rgb(0,0,0);  
  opacity: 0.5;}  
p.two {  
  background-color: rgb(0,0,0);  
  background-color: rgba(0,0,0,0.5);}
```

РЕЗУЛЬТАТ



ТЕКСТ

- ▶ Размер и гарнитура шрифта
- ▶ Полужирный, курсив, капитель, подчеркнутый
- ▶ Интервалы между строками, словами и буквами

ШРИФТ С ЗАСЕЧКАМИ*

Шрифты с засечками имеют дополнительные элементы по краям основных штрихов.

im

Шрифты с засечками традиционно используются для оформления больших объемов текста, поскольку считается, что они облегчают чтение.

РУБЛЕННЫЙ ШРИФТ**

Рубленые шрифты (также называемые **гротесками**) характеризуются прямыми краями букв и, следовательно, имеют гораздо более «чистый» вид.

im

Разрешение экранов ниже, чем разрешение печати, поэтому при небольшом размере шрифта текста рубленые шрифты воспринимаются легче.

МОНОШИРИННЫЙ ШРИФТ***

Все буквы моноширинного шрифта имеют одинаковую ширину (у букв в других шрифтах она разная).

i m

Моноширинные шрифты, как правило, используются при написании кода, поскольку они легко выравниваются, что делает текст более удобным для восприятия.

При выборе
гарнитуры шрифта
важно понимать
то, что браузер
отобразит его
только в том
случае, если
данный шрифт
установлен на
компьютере
пользователя.

ШРИФТ С ЗАСЕЧКАМИ

Шрифты с засечками имеют
дополнительные элементы
по краям основных штрихов.

ПРИМЕРЫ:

Georgia

Times

Times New Roman

РУБЛЕННЫЙ ШРИФТ

Рубленые шрифты (без
засечек) характеризуются
прямыми краями букв и, сле-
довательно, имеют гораздо
более чистый вид.

ПРИМЕРЫ:

Arial

Verdana

Helvetica


```
body {
    font-family: Arial, Verdana, sans-serif;
    font-size: 12px;}
h1 {
    font-size: 200%;}
h2 {
    font-size: 1.3em;}

@font-face {
    font-family: 'ChunkFiveRegular';
    src: url('fonts/chunkfive.eot');}
h1, h2 {
    font-family: ChunkFiveRegular, Georgia, serif;}
```

ПОЛУЖИРНОЕ НАЧЕРТАНИЕ

font-weight

Свойство `font-weight` позволяет форматировать шрифт текста полужирным начертанием. Обычно данное свойство имеет два значения.

normal

Использование данного значения приводит к тому, что текст имеет обычный вес.

bold

Это значение соответствует полужирному начертанию.

глава-12/font-weight.html

CSS

```
credits {  
  font-weight: bold;  
}
```

РЕЗУЛЬТАТ

Бриары

Автор: Аня Давыт

Бриар, или *berger de brie*, — это порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, вывозилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары служили на границах выскранки. В настоящее время популяция бриаров понемногу восстанавливается. Владетелями бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

text-transform

Свойство text-transform используется для изменения регистра текста и принимает одно из следующих значений.

uppercase

При данном значении для всех символов в тексте используется верхний регистр.

lowercase

При данном значении для всех символов в тексте используется нижний регистр.

capitalize

При данном значении каждое слово начинается с прописной буквы.

глава-12/text-transform.html

CSS

```
h1 {
  text-transform: uppercase;}
h2 {
  text-transform: lowercase;}
.credits {
  text-transform: capitalize;}
```

БРИАРЫ

Автор: Айва Даветт

Бриар, или *berger de brie*, — это крупная охотничья собака, традиционно используемая в качестве пастухов и охранников для стада овец.

история породы

Порода Бриар, возникшая предположительно во Франции, выводилась веками для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря острому слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров повсеместно восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

РЕЗУЛЬТАТ

text-decoration

CSS

глава-12/text-decoration.html

```
.credits {  
  text-decoration: underline;  
}  
a {  
  text-decoration: none;  
}
```

Свойство text-decoration принимает следующие значения.

none

Данное значение сбрасывает любое примененное к тексту оформление.

underline

Данное значение добавляет линию под текстом.

overline

Данное значение добавляет линию над текстом.

line-through

Данное значение добавляет линию, проходящую через слова.

blink

Данное значение заставляет текст мигать. Обычно эта

РЕЗУЛЬТАТ

Бриары

Автор: Айви Дакетт

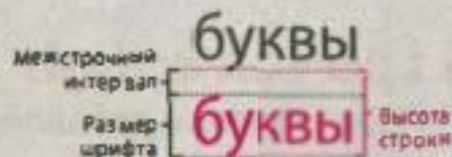
Бриар, или berger de brie, — это крупная порода собак, традиционно использовавшаяся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, выводится скрещиванием для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря отличному слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров повсеместно восстанавливается. Владельцами бриаров были Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

line-height

Межстрочный интервал — это типографский термин, обозначающий расстояние между строками текста. Часть символа, которая располагается ниже базовой линии, называется **нижним выносным элементом**, в то время как самая высокая точка символа называется **верхним выносным элементом**. Межстрочный интервал измеряется от нижней части нижнего выносного элемента на одной строке до верхней части верхнего выносного элемента на следующей строке.



глава-12/line-height.html

CSS

```
p {  
  line-height: 1.4em;}
```

РЕЗУЛЬТАТ

Бриары

Автор: Айван Даветт

Бриар, или Berger de brie, — это крупная порода собак, традиционно использовавшаяся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, выводится скрещиванием для выпаса и охраны овец. Во французский армия собаки данной породы использовались в качестве чистых в посылках. Они также помогали в поисках раненых солдат благодаря острому слуху. Во время Первой мировой войны бриары оказались на грани вымирания. В настоящее время популяция бриаров понемногу восстанавливается. Владельцами бриаров были: Карл Великий, Наполеон, Томас Джефферсон и Лафайет.

ВЫРАВНИВАНИЕ

text-align

Свойство `text-align` позволяет контролировать выравнивание текста. Оно может принимать одно из четырех следующих значений.

left

Определяет, что текст должен быть выровнен по левому краю.

right

Определяет, что текст должен быть выровнен по правому краю.

center

Данное значение позволяет выравнивать текст по центру.

глава-12/text-align.html

CSS

```
h1 {
  text-align: left;}
p {
  text-align: justify;}
.credits {
  text-align: right;}
```

Бриары

Автор: Айви Дакетт

Бриар, или *berger de brie*, — это крупная порода собак, традиционно использовавшихся в качестве пастухов и охранников для стада овец.

История породы

Порода Бриар, возникшая предположительно во Франции, выводилась специально для выпаса и охраны овец. Во французской армии собаки данной породы использовались в качестве часовых и посыльных. Они также помогали в поисках раненых солдат благодаря острому

РЕЗУЛЬТАТ

CSS3: ТЕКСТ С ТЕНЬЮ

text-shadow

CSS

rnana-12/text-shadow.html

```
p.one {
  background-color: #eeeeee;
  color: #666666;
  text-shadow: 1px 1px 0px #000000;}
p.two {
  background-color: #dddddd;
  color: #666666;
  text-shadow: 1px 1px 3px #666666;}
p.three {
  background-color: #cccccc;
  color: #ffffff;
  text-shadow: 2px 2px 7px #111111;}
p.four {
  background-color: #bbbbbb;
  color: #cccccc;
  text-shadow: -1px -2px #666666;}
```

Свойство text-shadow широко используется, несмотря на то, что поддерживается не всеми браузерами.

Оно используется для создания тени, представляющей собой более темную и слегка смещенную версию слова, расположенную позади него. С помощью этого свойства также можно создать эффект тиснения, добавив тень, которая несколько светлее самого текста.

Для создания тени свойству нужно сообщить довольно сложное значение, состоящее из трех длин и цвета тени.

ПРИМЕНЕНИЕ СТИЛЕЙ К ССЫЛКАМ

:link, :visited

CSS

глава-12/link-visited.html

```
a:link {
  color: deeppink;
  text-decoration: none;}
a:visited {
  color: black;}
a:hover {
  color: deeppink;
  text-decoration: underline;}
a:active {
  color: darkcyan;}
```

По умолчанию браузеры, как правило, выделяют ссылки синим цветом и подчеркиванием, а также меняют цвет посещенной ссылки, чтобы помочь пользователю отследить страницы, на которых он уже побывал.

В CSS существуют два псевдокласса, позволяющих применять различные стили к посещенным и не посещенным ссылкам.

РЕЗУЛЬТАТ

Породы собак: Б

- Бассет-хаунд
- Бедлингтон-терьер
- Бельгийская овчарка

:link

Этот псевдокласс позволяет применить стиль к еще не посещенной ссылке.

:visited

БЛОКИ

- ▶ Управление размером блоков
- ▶ Блочная модель: границы, поля и отступы
- ▶ Отображение и скрытие блоков

1

ГРАНИЦА

Каждый блок имеет границы, даже если они не видны или их толщина равна 0 пикселям. Границы отделяют края одного блока от краев других.

2

ПОЛЯ

Поля находятся за пределами границ. Вы можете настроить ширину полей для создания интервала между границами двух соседних блоков.

3

ОТСТУПЫ

Отступ — это интервал между границами блока и его содержимым. Добавление отступов может улучшить удобочитаемость контента.

Если вы задали ширину блока, то границы, поля и отступы будут добавлены к его ширине и высоте.



Блок


```
<title>Блоки</title>
<style type="text/css">
  body {
    font-size: 80%;
    font-family: "Courier New", Courier, monospace;
    letter-spacing: 0.15em;
    background-color: #efefef;}
  #page {
    max-width: 940px;
    min-width: 720px;
    margin: 10px auto 10px auto;
    padding: 20px;
    border: 4px double #000;
    background-color: #ffffff;}
  #logo {
    width: 150px;
    margin: 10px auto 25px auto;}
  ul {
    width: 570px;
    padding: 15px;
    margin: 0px auto 0px auto;
    border-top: 2px solid #000;
    border-bottom: 1px solid #000;
    text-align: center;}
```

CSS3: ТЕНИ БЛОКОВ

box-shadow

CSS

глава-13/box-shadow.html

```
p.one {
  -moz-box-shadow: -5px -5px #777777;
  -webkit-box-shadow: -5px -5px #777777;
  box-shadow: -5px -5px #777777;}

p.two {
  -moz-box-shadow: 5px 5px 5px #777777;
  -webkit-box-shadow: 5px 5px 5px #777777;
  box-shadow: 5px 5px 5px #777777;}

p.three {
  -moz-box-shadow: 5px 5px 5px 5px #777777;
  -webkit-box-shadow: 5px 5px 5px 5px #777777;
  box-shadow: 5px 5px 5px 5px #777777;}

p.four {
  -moz-box-shadow: 0 0 10px #777777;
  -webkit-box-shadow: 0 0 10px #777777;
  box-shadow: 0 0 10px #777777;}

p.five {
  -moz-box-shadow: inset 0 0 10px #777777;
  -webkit-box-shadow: inset 0 0 10px #777777;
```

Свойство `box-shadow` позволяет добавить тень вокруг блока. Оно работает так же, как свойство `text-shadow`, с которым вы познакомились в главе 12. Данное свойство должно использовать, по крайней мере, первое из следующих двух значений, а также цвет.

ГОРИЗОНТАЛЬНОЕ СМЕЩЕНИЕ

Отрицательные значения позиционируют тень слева от блока.

ВЕРТИКАЛЬНОЕ СМЕЩЕНИЕ

Отрицательные значения позиционируют тень над блоком.

РАЗМЫТИЕ

Если данное значение отсутствует, то тень изображается в виде сплошной линии, как граница.

CSS3: ЗАКРУГЛЕННЫЕ УГЛЫ

border-radius

В спецификации CSS3 появилась возможность создания закругленных углов блока с помощью свойства `border-radius`. Значение указывает радиус в пикселах.

Старые версии браузеров, которые не поддерживают это свойство, будут отображать блок с прямыми углами.

Свойства `-moz-border-radius` и `-webkit-border-radius` не входят в спецификацию CSS, однако их использование в некоторых версиях браузеров Chrome, Firefox и Safari обеспечивает работоспособность данного стиля.

Вы можете указать различ-

глава-13/border-radius.html

HTML

`<p>` При записи альбома Pet Sounds был использован целый ряд нетрадиционных инструментов, таких как звонок велосипеда, орган, клавесин, флейты, свисток для собак, поезд, струнные инструменты с характерным гавайским звучанием, банки из-под кока-колы и лающие собаки.`</p>`

CSS

```
p {
  border: 5px solid #cccccc;
  padding: 20px;
  width: 275px;
  text-align: center;
  border-radius: 10px;
  -moz-border-radius: 10px;
  -webkit-border-radius: 10px;}
```

МАКЕТ В HTML5

- ▶ Элементы макета HTML5
- ▶ Как старые версии браузеров воспринимают новые элементы
- ▶ Применение стилей к элементам макета HTML5 с помощью CSS

