

ФГБОУ ВПО
«Воронежский государственный технический университет»

Кафедра автоматизированных и вычислительных систем

142-2013

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА WEB-САЙТОВ
С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ HTML 5 И CSS 3

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению курсовой работы
по дисциплине «Web-дизайн»
для бакалавров направления 230100 «Информатика
и вычислительная техника», профиля
«Вычислительные машины, комплексы, системы и сети»
очной формы обучения



Воронеж 2013

Составители: канд. техн. наук М.Ю. Сергеев,
канд. техн. наук Т.И. Сергеева

УДК 681.3

Проектирование и разработка Web-сайтов с использованием технологий HTML 5 и CSS 3: методические указания к выполнению курсовой работы по дисциплине «Web-дизайн» для бакалавров направления 230100 «Информатика и вычислительная техника», профиля «Вычислительные машины, комплексы, системы и сети» очной формы обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост. М.Ю. Сергеев, Т.И. Сергеева. Воронеж, 2013. 44 с.

Методические указания содержат перечень тем курсовых работ и практических заданий к ним, а также правила оформления элементов курсовой работы, которая оформляется согласно требованиям СТП ВГТУ 62-2007.

Предназначены для студентов второго курса.

Ил.11. Библиогр.: 4 назв.

Рецензент канд. техн. наук, доц. О.Г. Яскевич

Ответственный за выпуск зав. кафедрой д-р техн. наук, проф. С.Л. Подвальный

Печатается по решению редакционно-издательского совета Воронежского государственного технического университета

© ФГБОУ ВПО «Воронежский государственный
технический университет», 2013

Введение

Данные методические указания содержат задания для выполнения курсовых работ по дисциплине «Web-дизайн» и рекомендации по оформлению курсовой работы в соответствии с требованиями СТП ВГТУ 62-2007.

1 Методические рекомендации по выполнению курсовой работы

1.1 Требования к содержанию пояснительной записки

Каждая курсовая работа имеет одинаковую структуру. Расчетно-пояснительная записка должна содержать: титульный лист (приложение А), задание на курсовую работу (приложение Б), лист для замечаний руководителя (приложение В), содержание (приложение Г), введение, теоретическая часть, практическая часть, заключение, список литературы (приложение Д).

Титульный лист оформляется в соответствии с требованиями стандартов ГОСТ и требованиями ВГТУ по образцу приложения А.

Задание на курсовую работу представлено на втором листе пояснительной записки и оформлено по образцу приложения Б. Задание помимо единых для всех заголовочных строк содержит:

- перечень вопросов, подлежащих разработке;
- объем работ;
- сроки выполнения этапов;
- срок защиты курсовой работы;
- подпись руководителя и дату выдачи задания;
- подпись студента и дату получения задания.

В перечне вопросов, подлежащих разработке, требуется указать поставленные в задании цели курсовой работы, например, освоить новые технологии разработки веб-страниц и веб-сайтов, реализовать практическое задание по созданию веб-сайта, оформить расчетно-пояснительную записку.

Объем работ описывает необходимые действия по реализации поставленной цели. Сюда можно включить поиск и обзор литературных источников, изучение необходимых технологий разработки элементов веб-страниц и веб-сайтов, реализацию практических заданий на создание сайта, оформление пояснительной записки.

В соответствии с указанным выше объемом работ все перечисленные работы распределяются по этапам и для каждого из них устанавливаются сроки их выполнения.

Лист замечаний руководителя остается пустым. При проверке пояснительной записки руководитель вписывает свои замечания по выполненной работе.

Содержание пояснительной записки имеет обычный смысл и включает наименование ее разделов и подразделов и соответствующие номера страниц.

Введение должно содержать цель курсовой работы, перечень решаемых задач для достижения поставленной цели. Во введении должно быть кратко отражено содержание основных разделов курсовой работы. Объем введения 1-2 страницы.

Теоретическая часть содержит обзор основных технологий веб-дизайна, используемых при создании сайта. Объем теоретической части 5-10 страниц.

Практическая часть описывает выполнение практического задания. Содержит описание созданного сайта и важные фрагменты HTML- и CSS-кода. Объем практической части занимает 10-15 страниц.

В заключении кратко описываются решенные в ходе курсовой работы задачи. Оцениваются изученные технологии с точки зрения удобства и эффективности использования для создания веб-сайта. Объем заключения – 1-2 страницы.

Список использованной литературы включает литературные источники, ссылки на которые должны быть сделаны по ходу выполнения работы. При использовании в пояснительной записке по курсовой работе цитат из литературы, периодических изданий или официальных нормативных документов обязательно следует делать ссылки на источники, спи-

сок которых составляется в порядке их упоминания в тексте. Ссылки записываются в квадратных скобках в тексте записки внутри предложения.

Список используемых литературных источников должен содержать не менее пяти наименований, существующих на бумажных носителях, и не менее двух адресов сайтов сети Интернет. В список литературы включаются только источники, использованные в работе и содержащие существенную информацию по разрабатываемой теме. Объем списка литературы – 1 страница.

1.2 Требования к оформлению пояснительной записки

Общий объем расчетно-пояснительной записки по курсовой работе составляет 20-25 листов формата А4, выполненных в редакторе Microsoft Word одной из последних версий на момент формирования расчетно-пояснительной записки.

К оформлению записки предъявляются следующие требования:

- шрифт Times New Roman Cyr 14 (или 12 пт.);
- полуторный (или одинарный) интервал;
- страницы пронумерованы, номера располагаются внизу страницы по центру;
- абзацы выровнены по ширине, первая строка абзаца имеет отступ 1.25 см;
- названия разделов, введение, заключение, список литературы выводятся с первой прописной буквы, остальные строчные и выровнены по центру; названия подразделов выводятся с первой прописной буквы, остальные строчные и оформляются как абзацы;
- подписи к рисункам выводятся по центру под рисунком в виде «Рисунок 1.1 – Структура главной страницы сайта»;
- название таблицы выводится над таблицей без абзацного отступа в виде «Таблица 1.2 – Основные стили оформления страниц сайта».

2 Теоретические сведения

2.1 Общие сведения о HTML 5

Считается, что разработка стандарта началась в 2004 году Рабочей группой по разработке Гипертекстовых Прикладных Технологий в Веб (Web Hypertext Application Technology Working Group), когда W3C сосредоточилась на разработке XHTML 2.0. В 2009 году от разработки последнего решено было отказаться и обе группы объединились в работе над будущим стандартом HTML.

Следует заметить, что стандарт HTML 5 до сих пор находится в разработке и, соответственно, существует в виде черновика. Но большинство современных браузеров поддерживают те нововведения, которые внесены этим стандартом в разметку web-документов. Принятие полной спецификации HTML ожидается в 2014 году.

К новым особенностям HTML, любопытных в плане Web-дизайна можно отнести:

- добавление тегов, облегчающих отображение мультимедийной и графической информации;
- добавление тегов, отображающих семантическое содержание документа;
- добавление новых видов полей для формы.

HTML 5 потенциально дает много возможностей web-дизайнеру. Некоторые даже утверждают, что он заменит JavaScript и CSS. На самом деле это не так – за HTML остается структурирование содержимого веб-страницы, CSS управляет отображением элементов на странице, а JavaScript «оживляет» элементы web-страницы, добавляя им анимацию и делая их интерактивными.

Основные различия между HTML 4.01 и HTML 5

Каким образом можно увидеть, что мы имеем дело со страницей, созданной с помощью HTML 5? Самый верный способ – обратиться к элементу `<!DOCTYPE>`.

Элемент `<!DOCTYPE>` предназначен для указания типа текущего документа – DTD (document type definition, описание типа документа). Чтобы браузер «не путался» и понимал, согласно какому стандарту отображать веб-страницу и необходимо в первой строке кода задавать `<!DOCTYPE>`.

В старых версиях `<!DOCTYPE>` представлял собой весьма громоздкую конструкцию, состоящую из множества частей. В HTML 5 он пишется достаточно просто:

```
<!DOCTYPE html>
```

Если у документа сверху стоит такой `<!DOCTYPE>`, то он интерпретируется браузером как документ HTML 5.

Прежде, чем перейти к описанию новых тегов, стоит огласить список старых тегов, использование которых осуждается спецификацией HTML 5.

Теги, не вошедшие в спецификацию HTML 5 (возможно, в окончательной спецификации будет другой список):

<code><acronym></code>	<code><isindex></code>
<code><applet></code>	<code><marquee></code>
<code><basefont></code>	<code><nobr></code>
<code><bgsound></code>	<code><noembed></code>
<code><big></code>	<code><noframes></code>
<code><blink></code>	<code><plaintext></code>
<code><center></code>	<code><s></code>
<code><comment></code>	<code><strike></code>
<code><dir></code>	<code><tt></code>
<code></code>	<code><u></code>
<code><frame></code>	<code><xmp></code>
<code><frameset></code>	

2.2 Теги для отображения мультимедийных объектов

Тег `<audio>`

Тег `<audio>` Добавляет, воспроизводит и управляет настройками аудиозаписи на веб-странице. Путь к файлу задается через атрибут `src` или вложенный тег `<source>`. Внутри кон-

тейнера `<audio>` можно написать текст, который будет выводиться в браузерах, не работающих с этим тегом. Данный тег будет работать в браузерах IE (с 9-й версии), Chrome 10.0 и выше, Opera 10.50 и выше, Firefox 4.0 и выше, Safari 3.1 и выше.

Имеет синтаксис:

```
<audio src="URL"></audio>
```

или

```
<audio>  
  <source src="URL">  
</audio>
```

Список кодеков и поддерживаемых форматов несколько ограничен и зависит от вида и версии браузера.

Тег `<audio>` имеет ряд важных атрибутов:

- `autoplay` – при добавлении этого атрибута к тегу звук начинает проигрываться сразу при загрузке страницы. Можно добавить этот атрибут просто без значения, а можно написать `autoplay="autoplay"`;

- `controls` – добавляет панель управления к аудиотреку. Вид панели и ее содержимое зависит от браузера и может в себя включать: кнопку воспроизведения, паузы, перемотки, ползунок для изменения уровня громкости и др. Способ добавления – аналогичен `autoplay`;

- `loop` – закликивает воспроизведение аудио, чтобы оно повторялось с начала после завершения. Способ добавления – аналогичен `autoplay`;

- `preload` – используется для загрузки аудиофайла вместе с загрузкой веб-страницы. Этот атрибут игнорируется, если установлен `autoplay`. Имеет 3 значения: `none` – не загружать аудиофайл (значение по умолчанию); `metadata` – загружать только служебную информацию (продолжительность звучания и др.); `auto` – загружать файл при загрузке страницы.

Пример использования:

```
<audio autoplay preloadp="metadata">  
  <source src="audio/music.ogg" type='audio/ogg;  
    codecs=vorbis'>  
  <source src="audio/music.mp3" type="audio/mpeg">  
  Тер audio не поддерживается вашим браузером.  
  <a href="audio/music.mp3">  
    Скачайте музыку  
  </a>  
</audio>
```

Результат использования тега <audio> показан на рисунке 1.

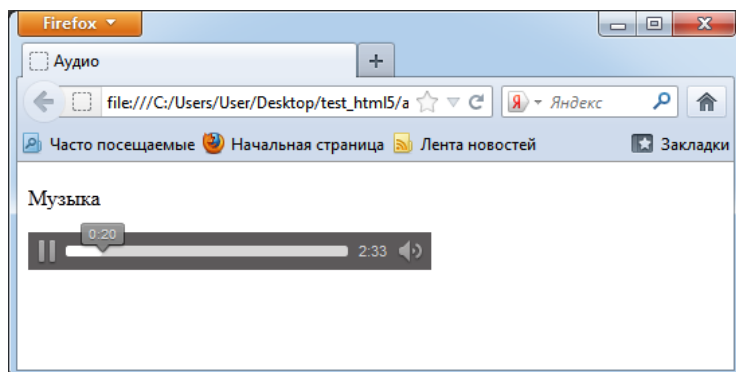


Рисунок 1 - Результат использования тега <audio> в браузере Firefox

Для универсального воспроизведения во всех браузерах аудио кодируют с помощью разных кодеков и добавляют файлы одновременно через тег <source>. Браузер просмотрит все теги <source> выберет из них файл, который он сможет воспроизводить.

Прежде, чем перейти к тегу для воспроизведения видео обратим внимание, на упомянутый выше тег <source>.

Тег <source>

Данный тег предназначен для добавления ссылки на файл. Обычно используется в связке с тегами для отображения аудио и видео.

Синтаксис:

<audio>

<source src="URL">

</audio>

<video>

<source src="URL">

</video>

Тег <source> имеет ряд следующих атрибутов.

media – тип устройства, на котором будет воспроизводиться данный файл. Имеет следующие значения:

all – все устройства;

braille – устройства, основанные на системе Брайля и предназначенные для слепых людей;

handheld – наладонники, смартфоны, устройства с малой шириной экрана;

print – печатающее устройство вроде принтера;

screen – экран монитора;

speech – речевые синтезаторы, а также программы для воспроизведения текста вслух. Сюда же входят речевые браузеры;

projection – проектор;

tty – телетайпы, терминалы, портативные устройства с ограниченными возможностями экрана;

tv – телевизор;

src – адрес файла.

type – MIME-тип медийного источника. Существует большое количество MIME-типов, определяющих вид добавленного на страницу файла.

Например:

type="audio/mpeg" – файл mp3;
type="video/mpeg" – файл mpeg;
type="text/css" – файл CSS и т.д.

Тег <video>

Добавляет, воспроизводит и управляет настройками видеоролика на веб-странице. Путь к файлу задается через атрибут src или вложенный тег <source>. Список поддерживаемых браузерами видеокодеков ограничен.

Синтаксис:

```
<video>  
  <source src="URL">  
</video>
```

Тег имеет следующие атрибуты:

- autoplay, controls, loop, preload, src – аналогичны тегу <audio>;
- height и width – задают высоту и ширину области для просмотра видеоролика;
- poster – указывает адрес картинки, которая будет отображаться пока видео недоступно или не воспроизводится.

Пример добавления видео на страницу:

```
<video width="400" height="300" controls="controls"  
poster="video/myvid.jpg">  
  <source src="video/myvid.ogv" type='video/ogg;  
    codecs="theora, vorbis"'>  
  <source src="video/myvid.mp4" type='video/mp4;  
    codecs="avc1.42E01E, mp4a.40.2"'>  
  <source src="video/myvid.webm" type='video/webm;  
    codecs="vp8, vorbis"'>  
  Тег video не поддерживается вашим браузером.  
  <a href="video/myvid.mp4">  
    Скачайте видео  
  </a>  
</video>
```

Пример использования тега `<video>` показан на рисунке 2.

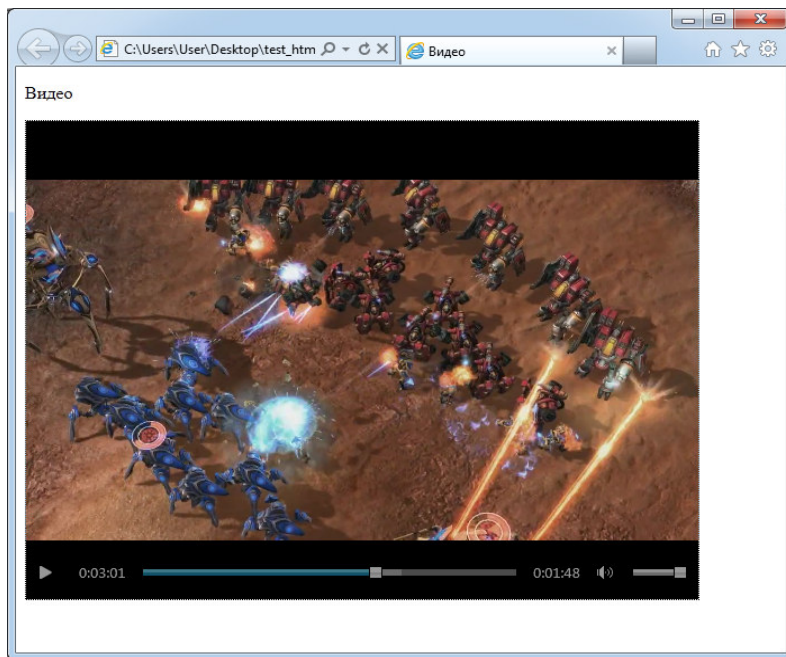


Рисунок 2 - Результат использования тега `<video>` в браузере Internet Explorer

Тег `<canvas>`

Данный тег предназначен для создания области, в которой при помощи JavaScript можно рисовать разные объекты, выводить изображения, трансформировать их и менять свойства.

Синтаксис:

```
<canvas id="идентификатор">  
</canvas>
```

Обратившись к ID элемента можно применить к нему сценарий JavaScript и добавить в него рисунки и анимацию.

Имеет два атрибута – width и height – устанавливающие высоту и ширину области для рисования.

2.3 Теги для обозначения блоков страницы

При блочной верстке страниц, содержимое страницы делилось на отдельные части (блоки) с помощью тегов <div>. Каждый тег <div> помечался определенным идентификатором, который вместе со стилем CSS определял, что это за блок и где он будет размещаться. Например, блок заголовка страницы помечался тегом <div> с ID “header”, блок подножия – тегом <div> с ID “footer”.

В HTML 5 разработчики решили пойти навстречу веб-дизайнеру и добавили несколько аналогов тегу <div>, имена которых призваны отразить предназначение (смысл или семантику) блока. Действительно, более удобным и наглядным будет вот такой код:

```
<header>  
  Заголовок сайта  
</header>
```

чем

```
<div id="header">  
  Заголовок сайта  
</div>
```

Далее веб-дизайнер может создать стиль для тега <header> и отформатировать заголовок сайта, как он считает нужным.

Для удобства верстки в HTML 5 добавлены следующие теги.

<article> – задает содержание сайта вроде новости, статьи, записи блога, форума или др. Атрибутов нет.

`<aside>` – определяет блок сбоку от контента для размещения рубрик, ссылок на архив, меток и другой информации. Такой блок, как правило, называется «сайдбар» или «боковая панель». Применимы универсальные атрибуты.

`<details>` – используется для хранения информации, которую можно скрыть или показать по требованию пользователя. По умолчанию содержимое тега не отображается, для изменения статуса применяется атрибут `open` (например, `<details open=«open»>Текст</details>`).

`<footer>` – задаёт «подвал» сайта или раздела, в нём может располагаться имя автора, дата документа, контактная и правовая информация. Атрибутов нет.

`<header>` – задает «шапку» сайта или раздела, в которой обычно располагается заголовок. Атрибутов нет.

`<hgroup>` – используется для группирования заголовков веб-страницы или раздела. Внутри располагаются теги заголовков от `<h1>` до `<h6>`. Атрибутов нет.

`<menu>` – предназначен для отображения списка пунктов меню. Аналогично тегам `<ol>` и `<ul>` внутри контейнера `<menu>` список формируется с помощью тегов `<li>`. Атрибут `label` устанавливает видимую метку для меню, атрибут `type` задает тип меню (значения: `context` – контекстное меню, `toolbar` – меню в виде панели инструментов, `list` – меню в виде списка).

`<nav>` – задает навигацию по сайту. Если на странице несколько блоков ссылок, то в `<nav>` обычно помещают приоритетные ссылки. Также допустимо использовать несколько тегов `<nav>` в документе. Запрещается вкладывать `<nav>` внутрь `<address>`. Атрибутов нет.

`<section>` – задает раздел документа, который может включать в себя заголовки, шапку, подвал и текст. Допускается вкладывать один тег `<section>` внутрь другого.

`<summary>` – указывает заголовок для тега `<details>`, по которому можно щелкать для разворачивания/сворачивания информации. Тег `<summary>` должен идти первым внутри `<details>`.

Синтаксис:

```
<details>
  <summary>Текст</summary>
</details>
```

В дополнение можно упомянуть следующие теги полезные для использования на странице.

`<mark>` – помечает текст как выделенный. Такой текст ничем не отличается от обычного, но его вид может быть изменен с помощью стилей. В браузере Chrome и Firefox фоновый цвет текста внутри `<mark>` выделяется желтым цветом.

`<figure>` – используется для группирования любых элементов, например, изображений и подписей к ним.

`<figcaption>` – содержит описание для тега `<figure>`. Тег `<figcaption>` должен быть первым или последним элементом в группе.

Пример – создание изображения с подписью:

```
<figure>
  <div></div>
  <figcaption>Led Zeppelin II <br> 1969 </figcaption>
</figure>
```

Теперь создадим стили для тегов `<figure>` и `<figcaption>`:

```
figure {
  background: #5f6a72; /* Цвет фона */
  padding: 5px; /* Поля вокруг */
  display: block; /* Блочный элемент */
  width: 350px; /* Ширина */
  float: left; /* Блоки выстраиваются по горизонтали */
  margin: 0 10px 10px 0; /* Отступы */
  text-align: center; /* Выравнивание по центру */
}
figcaption {
  color: #fff; /* Цвет текста */
}
```

Получившийся результат показан на рисунке 3.



Рисунок 3 - Результат использования тегов `<figure>` и `<figcaption>`

2.4 Общие сведения о CSS 3

С 1996 года CSS значительно «повзрослел» и стал основным инструментом веб-дизайнеров. Благодаря их стремлению делать все более красивые веб-сайты последующие версии CSS приобретали больше свойств и, как следствие, большую сложность. Версию CSS 3 W3C разделил на независимые модули, каждый из которых описывает отдельные наборы

свойств. Назначение некоторых модулей известно, например CSS Transitions (анимация или трансформация переходов между двумя состояниями – подсветка текста при наведении на него указателя мыши) или CSS Web Fonts (инструкции для изменения различных свойств шрифта). Функции других пока точно не определены, например CSS Math или CSS Namespaces.

Основная цель подобного модульного деления состоит в том, чтобы дать возможность CSS развиваться постепенно, не заставляя всех ждать, когда же наконец будет готов один гигантский набор инструкций. Каждый модуль развивается отдельно, а производители браузеров могут использовать отдельные инструкции модулей сразу, как только те будут готовы.

Хотя CSS 3 еще не стал общепринятым стандартом, многие наиболее привлекательные свойства CSS 3 нашли свое применение в браузерах.

Некоторые новые возможности CSS 3:

- введение новых видов селекторов для более гибкой сборки элементов на странице;
- новые виды оформления текста (тень, разбиение текста на колонки и т.д.);
- настройка прозрачности элементов страницы;
- создание скругленных границ элементов;
- новые свойства, специально предназначенные для элементов форм.

2.5 Селекторы в CSS 3

Селекторы дочерних элементов

CSS 3 предоставляет новые возможности по форматированию тегов, которые являются дочерними по отношению к остальным.

Рассмотрим основные свойства, связанные с дочерними элементами.

`:first-child` – данный псевдокласс применяет стилевое оформление к первому дочернему элементу своего родителя. На самом деле, `:first-child` появился в стандарте CSS 2, но остальные подобные селекторы добавились в CSS 3, поэтому уместно разместить его описание здесь.

`:last-child` – псевдокласс задает стилевое оформление последнего элемента своего родителя.

В качестве примера, можно использовать данные селекторы для оформления первого и последнего элемента маркированного списка:

```
ul :first-child {text-decoration: underline}
```

```
ul :last-child { text-decoration: line-through}
```

Результат использования данных стилей показан на рисунке 4.

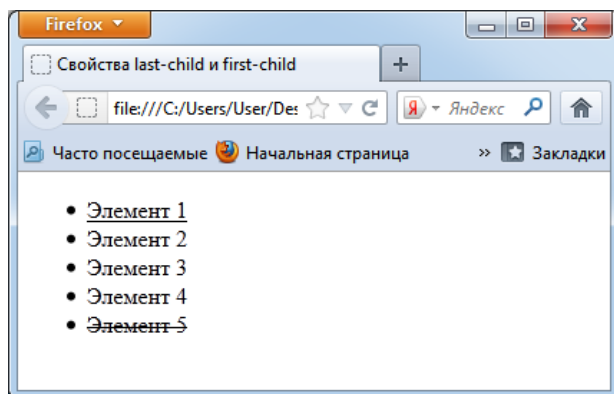


Рисунок 4 - Результат применения селекторов `:first-child` и `:last-child` к маркированному списку

Следует обратить внимание на пробел, расположенный между `ul` и `:first-child`. Эта конструкция означает «найти первый дочерний элемент тега `<ul>`». Конструкция вида `ul:first-child` без пробела имеет совершенно другой смысл: «найти все теги `<ul>`, которые являются первыми дочерними для какого-то другого тега».

:nth-child() – используется для добавления стиля к элементам на основе нумерации в дереве элементов. Синтаксис данного свойства выглядит следующим образом:

элемент :nth-child(odd | even | <число> | <выражение>) {...}

Как видно в скобках после название свойства должно быть указано число или выражение для определения стилизуемых элементов. Возможны следующие значения:

odd – все нечетные номера элементов;

even – все четные номера элементов;

число – порядковый номер дочернего элемента относительно своего родителя. Нумерация начинается с 1, это будет первый элемент в списке;

выражение – задается в виде $an+b$, где a и b целые числа, а n – счетчик, который автоматически принимает значение 0, 1, 2...

Если a равно нулю, то оно не пишется и запись сокращается до b . Если b равно нулю, то оно также не указывается и выражение записывается в форме an . a и b могут быть отрицательными числами, в этом случае знак плюс меняется на минус, например: $5n-1$.

За счет использования отрицательных значений a и b некоторые результаты могут также получиться отрицательными или равными нулю. Однако на элементы оказывают влияние только положительные значения из-за того, что нумерация элементов начинается с 1.

Примеры выражений:

$2n - 2, 4, 6, 8, 10...$ – все четные элементы, аналог значения even.

$2n+1 - 1, 3, 5, 7, 9...$ – все нечетные элементы, аналог значения odd.

$3n+2 - 5, 8, 11, 14$ и т.д.

$-n+3 - 3, 2, 1$.

Применим данный селектор к элементам маркированного списка, воспользовавшись значениями odd и even:

```
ul :nth-child(odd) {text-decoration: underline}  
ul :nth-child(even) { text-decoration: line-through}
```

Результат использования данных стилей показан на рисунке 5.

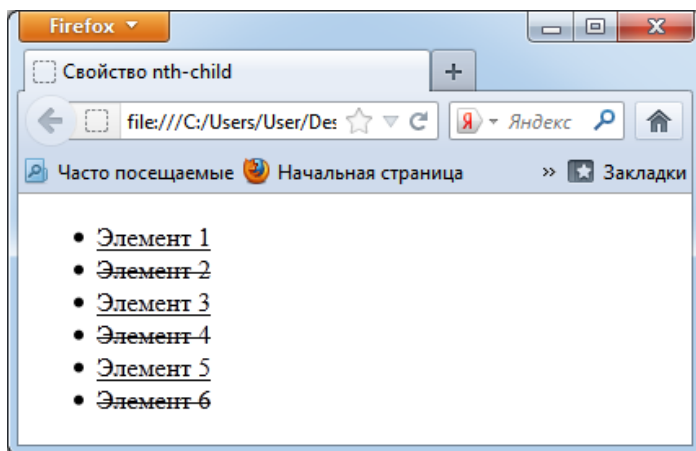


Рисунок 5 - Результат применения селекторов :nth-child

:nth-last-child – используется для добавления стиля к элементам на основе нумерации в дереве элементов. В отличие от псевдокласса :nth-child отсчет ведется не от первого элемента, а от последнего. В остальном принцип использования данного свойства тот же самый.

:only-child – применяется к дочернему элементу, только если он единственный у родителя. Аналогичен использованию :first-child:last-child или :nth-child(1):nth-last-child(1).

Селекторы типов

В CSS 3 введены новые селекторы, которые работают так же, как и селекторы дочерних элементов за тем лишь исклю-

чением, что позволяют указать на конкретный тип выбираемого тега. Предположим, что необходимо оформить первый абзац внутри боковой панели особым образом, однако на некоторых страницах первым абзацем является заголовок, начинающийся с тега `<h2>`, в то время как на других страницах это тег `<p>`. В данном случае не получится использовать `:first-child` для выбора абзаца, поскольку в некоторых случаях он будет являться вторым после тега `<h2>`. Тем не менее тег `<p>` всегда является первым абзацем, даже если перед ним находятся другие теги. Поэтому можно будет воспользоваться селектором типа `:first-of-type`.

Рассмотрим данные селектор и подобные ему селекторы типов.

`:first-of-type` – работает точно так же, как и `:first-child`, но позволяет указать на конкретный тип дочернего тега. Например, есть боковая панель, в которой `id="sidebar"`. Чтобы оформить первый абзац в этой панели, можно воспользоваться следующим стилем:

```
#sidebar p:first-of-type
```

Следует обратить внимание на то, что именно тег `<p>` является первым нужным типом.

`:last-of-type` – работает точно так же, как и `:last-child`, но указывает на последний тег конкретного типа.

`:nth-of-type` – работает точно так же, как и `:nth-child`, но указывает на последний тег конкретного типа. Данный селектор удобно использовать, например, для изображений находящихся в объемном тексте. Чтобы разместить изображения в тексте в шахматном порядке можно добавить следующие стили:

```
img:nth-of-type(odd) {float: left}  
img:nth-of-type(even) {float: right}
```

`:nth-last-of-type` – используется для добавления стиля к элементам указанного типа на основе нумерации в дереве элементов. В отличие от псевдокласса `:nth-of-type` отсчет ведется не от первого элемента, а от последнего.

`:only-of-type` – применяется к дочернему элементу указанного типа, только если он единственный у родителя. Аналогичен использованию `:first-of-type:last-of-type` или `:nth-of-type(1):nth-last-of-type(1)`.

2.6 Форматирование текста в CSS 3

Создание тени для текста

В CSS 3 включено свойство, которое позволяет добавить тень тексту, чтобы придать ему глубины (например, заголовкам, спискам или отдельным абзацам). Это свойство называется `text-shadow`.

Синтаксис:

`text-shadow: none | тень [ ,тень ]`

где тень:

`<цвет> <сдвиг по x> <сдвиг по y> <радиус размытия>`

Данное свойство может принимать следующие значения.

`none` – отменяет добавление тени.

`<цвет>` – цвет тени в любом доступном CSS формате. По умолчанию цвет тени совпадает с цветом текста. Необязательный параметр.

`<сдвиг по x>` – смещение тени по горизонтали относительно текста. Положительное значение этого параметра задает сдвиг тени вправо, отрицательное – влево. Обязательный параметр.

`<сдвиг по y>` – смещение тени по вертикали относительно текста. Также допустимо использовать отрицательное значение, которое поднимает тень выше текста. Обязательный параметр.

`<радиус>` – задает радиус размытия тени. Чем больше это значение, тем сильнее тень сглаживается, становится шире и

светлее. Если этот параметр не задан, по умолчанию устанавливается равным 0. Учтите, что алгоритм сглаживания в браузерах обычно разный, поэтому вид тени может несколько различаться в зависимости от заданных параметров сглаживания.

Допускается указывать несколько параметров тени, разделяя их между собой запятой. В CSS3 учитывается следующий порядок: первая тень в списке размещается на самом вер-ху, последняя в списке – в самом низу.

В качестве примера зададим следующий стиль:

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

Первое значение задает цвет тени. Значение «-4px» можно прочесть как «поместить тень на 4 пикселя левее текста». Следующий параметр – «4px» – размещает тень на 4 пикселя ниже текста. Значение «3px» определяет смазанность тени. Нулевое значение (полное отсутствие смазанности) даст очень четкуб и резкую тень, в то время как большие значения сделают ее более размытой.

Можно добавить две тени сразу, например:

```
.sh1 {  
    text-shadow: -4px 4px 3px #666, 1px -1px 2px  
#000;  
}
```

Результат использования данных стилей показан на рисунке 6. К верхней надписи применен первый стиль, к нижней - второй, с двойной тенью.

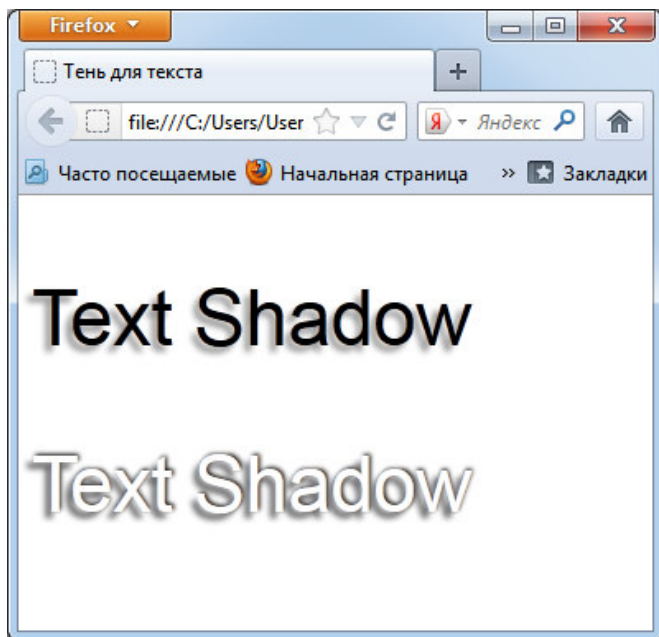


Рисунок 6 - Пример использования свойства text-shadow

Разбиение текста на колонки

В CSS 3 добавлен ряд свойств позволяющий разбить многострочный текст на две или большее количество колонок. Рассмотрим данные свойства.

`column-count` – Определяет количество колонок в тексте. Следует заметить, что данное свойство поддерживается только в браузере Internet Explorer версии 10 и выше, а также в браузере Opera, начиная с версии 11.1. Браузер Firefox использует нестандартное свойство `-moz-column-count`, Safari, Chrome и iOS поддерживают нестандартное свойство `-webkit-column-count`.

Чтобы разбиение текста на колонки работало во всех браузерах можно использовать вот такой стиль:

```
.book {  
  -moz-column-count: 2; /* Для Firefox */
```

```
-webkit-column-count: 2; /* Для Safari и Chrome */
column-count: 2;
}
```

В качестве значения используется либо целое число, либо значение `auto` (число колонок вычисляется автоматически на основе других свойств: `column-width`, `column-gap` и т.д.)

`column-width` – задаёт оптимальную ширину колонки в многоколоночном тексте. Реальная ширина колонки может быть больше введённого значения, сокращение указанной ширины приводит к отмене нескольких колонок, текст при этом выстраивается в одну колонку. Браузер Firefox использует нестандартное свойство `-moz-column-width`, Safari, Chrome и iOS поддерживают нестандартное свойство `-webkit-column-width`. Значение задается в абсолютных величинах, например, в пикселях. Также можно использовать значение `auto` (ширина колонок вычисляется автоматически на основании других свойств).

`column-gap` – задаёт расстояние между колонками в многоколоночном тексте. Браузер Firefox использует нестандартное свойство `-moz-column-gap`, Safari, Chrome и iOS поддерживают нестандартное свойство `-webkit-column-gap`. Значение задается в абсолютных величинах, например, в пикселях. Также можно использовать значение `normal` (межколоночный интервал задается браузером).

`column-rule` – в многоколоночном тексте отрисовывает линию между колонками. Синтаксис:

```
column-rule: <border-width> || <border-style> || <цвет>
```

Значения:

`border-width` определяет толщину линии. Для управления видом линии предоставляется несколько значений `border-style` (значения аналогичны видам границ элемента: `solid` – сплошная линия, `double` – сплошная двойная линия, `dashed` – штриховая линия, `dotted` – линия из точек). `<цвет>` устанавливает

цвет линии, значение может быть в любом допустимом для CSS формате.

`columns` – универсальное свойство, которое позволяет одновременно задать ширину и количество колонок многоколоночного текста. Браузер Firefox использует нестандартное свойство `-moz-columns`, Safari, Chrome и iOS поддерживают нестандартное свойство `-webkit-columns`. Синтаксис:

```
columns: [column-width] || [column-count]
```

В качестве значений выступает комбинация свойств `column-width` и `column-count`. Порядок значения не имеет.

Пример использования свойства `column`:

```
columns: 200px auto;
```

Ширина колонок 200 пикселей, количество задаётся браузером.

```
columns: 3 12em;
```

Три колонки минимальной ширины 12em каждая.

Результат разбиения текста на три колонки шириной 12em показан на рисунке 7.

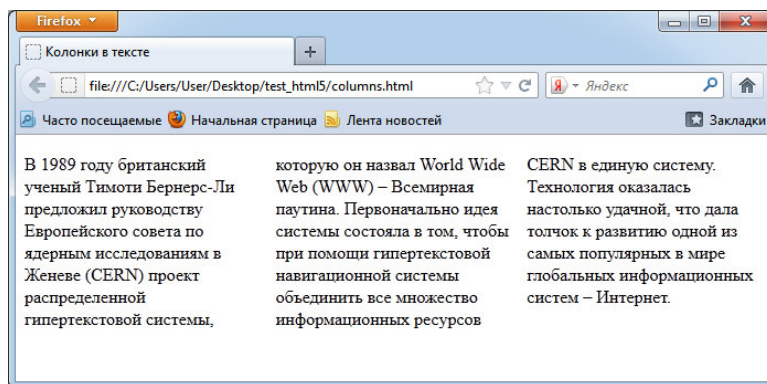


Рисунок 7 - Текст, разбитый на колонки с помощью CSS

Подключение собственного шрифта

Благодаря директиве `@font-face` перед веб-дизайнерами открываются новые возможности. Используя, данное нововведение, можно разместить на сервере файл шрифта в формате

Open Type (.otf) или True Type (.ttf), а затем воспользоваться директивой `@font-face`, чтобы браузер посетителя скачал указанный шрифт на время просмотра сайта.

Пример типичного использования директивы `@font-face`:

```
@font-face {  
    font-family: Lavoisier;  
    src: url('lavoisier.otf');  
}
```

Свойство `font-family` определяет имя, которое будет использоваться в стилях. В данном случае это `Lavoisier`. Оно не обязательно должно совпадать с настоящим именем шрифта, который будет использоваться. Свойство `src` используется для указания URL-адреса файла шрифта.

Лишь после определения имени шрифта его возможно использовать в своем файле стилей. Например:

```
h1 {  
    font-family: Lavoisier, Arial, Helvetica, sans-serif;  
    color: #FF993E;  
    font-size: 48px;  
}
```

В данном случае рекомендуется в свойстве `font-family` указать дополнительные шрифты, которые браузер пользователя применит к элементу, если не поймет директиву `@font-face`.

2.7 Работа с прозрачностью и фоном элемента в CSS 3

Прозрачность

В CSS 3 добавлено специальное свойство `opacity`, которое позволяет настроить степень «прозрачности» видимого элемента на странице.

Данное свойство может принимать значение в диапазоне от 0 (полностью прозрачный) до 1 (непрозрачный). Например, если необходимо сделать какое-либо изображение полупрозрачным, то можно использовать следующий стиль:

```
.ghost {  
    opacity: 0.5;  
}
```

Firefox до версии 3.5 поддерживает нестандартное свойство `–moz-opacity`.

Пример использования свойства `opacity` показано на рисунке 8. Слева приведено непрозрачное изображение, справа - изображение с прозрачностью 0.5.

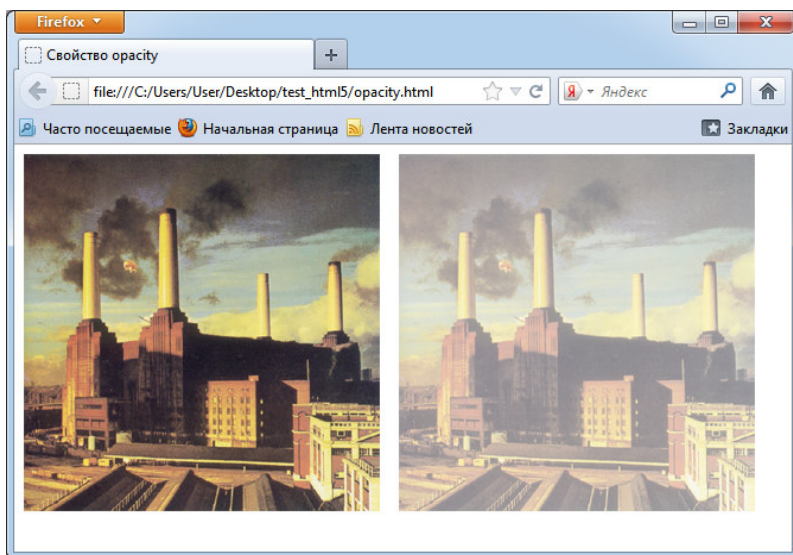


Рисунок 8 - Пример использования свойства `opacity`

Можно применять данное свойство вместе с псевдоклассом `:hover`, чтобы реализовать эффект выделения ссылок или изображений при наведении на них курсора мыши.

Добавление тени элемента

Свойство `box-shadow` добавляет тень к элементу. Допускается использовать несколько теней, указывая их параметры через запятую, при наложении теней первая тень в списке будет выше, последняя ниже. Добавление тени увеличивает ширину элемента, поэтому возможно появление горизонтальной полосы прокрутки в браузере.

Синтаксис:

`box-shadow: none | <тень> [,<тень>]`

где <тень>:

`inset <сдвиг по x> <сдвиг по y> <радиус размытия> <растяжение> <цвет>`

Для настройки вида тени используются следующие значения.

`none` – отменяет добавление тени.

`inset` – тень выводится внутри элемента. Необязательный параметр.

`<сдвиг по x>` – смещение тени по горизонтали относительно элемента. Положительное значение этого параметра задает сдвиг тени вправо, отрицательное – влево. Обязательный параметр.

`<сдвиг по y>` – смещение тени по вертикали относительно элемента. Положительное значение задает сдвиг тени вниз, отрицательное – вверх. Обязательный параметр.

`<размытие>` – задает радиус размытия тени. Чем больше это значение, тем сильнее тень сглаживается, становится шире и светлее. Если этот параметр не задан, по умолчанию устанавливается равным 0, тень при этом будет четкой, а не размытой.

`<растяжение>` – положительное значение растягивает тень, отрицательное, наоборот, ее сжимает. Если этот параметр не задан, по умолчанию устанавливается 0, при этом тень будет того же размера, что и элемент.

<цвет> – цвет тени в любом доступном CSS формате, по умолчанию тень черная. Необязательный параметр.

Допускается указывать несколько теней, разделяя их параметры между собой запятой. Учитывается следующий порядок: первая тень в списке размещается на самом верху, последняя в списке – в самом низу.

Firefox до версии 4.0 поддерживает нестандартное свойство `-moz-box-shadow`. Safari до версии 5.1, Chrome до версии 10.0 и iOS поддерживают свойство `-webkit-box-shadow`.

Пример использования:

```
.shadow {  
background: #fc0; /* Цвет фона */  
-moz-box-shadow: 0 0 10px #999; /* Для Firefox */  
-webkit-box-shadow: 0 0 10px #999; /* Для Safari и Chrome */  
box-shadow: 0 0 10px #999; /* Параметры тени */  
padding: 10px;  
}
```

Результат показан на рисунке 9.

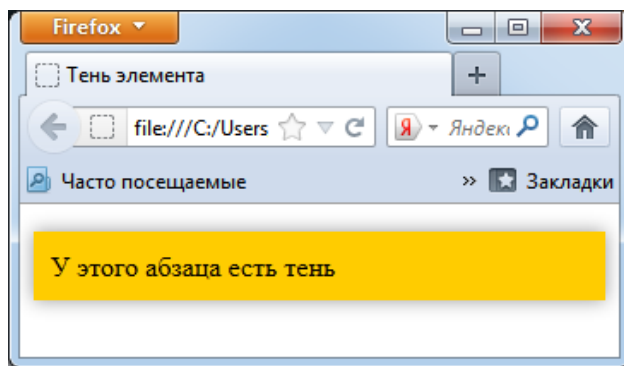


Рисунок 9 - Пример создания тени элемента средствами CSS

Работа с фоновым изображением

Рассмотрим несколько новых свойств для работы с фоновыми изображениями.

`background-size` – масштабирует фоновое изображение согласно заданным размерам.

Синтаксис:

`background-size: [ <значение> | <проценты> | auto ]{1,2} | cover | contain`

Может принимать следующие значения.

`<значение>` – задает размер в любых доступных для CSS единицах – пиксели (px), сантиметры (cm), em и др.

`<проценты>` – задает размер фоновой картинки в процентах от ширины или высоты элемента.

`auto` – если задано одновременно для ширины и высоты (`auto auto`), размеры фона остаются исходными; если только для одной стороны картинки (`100px auto`), то размер вычисляется автоматически исходя из пропорций картинки.

`cover` – масштабирует изображение с сохранением пропорций так, чтобы его ширина или высота равнялась ширине или высоте блока.

`contain` – масштабирует изображение с сохранением пропорций таким образом, чтобы картинка целиком поместилась внутрь блока.

Если установлено одно значение, оно задает ширину фона, второе значение принимается за `auto`. Пропорции картинки при этом сохраняются. Использование двух значений через пробел задает ширину и высоту фоновой картинки.

Safari до версии 4.1 и Chrome до версии 3.0 используют нестандартное свойство `-webkit-background-size`. Opera до версии 10.53 использует нестандартное свойство `-o-background-size`. Firefox до версии 4.0 использует нестандартное свойство `-moz-background-size`.

`background-origin` – определяет область позиционирования фонового рисунка. Это свойство не применяется, когда значение `background-attachment` задано как `fixed`.

Синтаксис:

`background-origin: [padding-box | border-box | content-box] [, [padding-box | border-box | content-box]]`

Может принимать следующие значения.

`padding-box` – фон позиционируется относительно края элемента с учетом толщины границы.

`border-box` – фон позиционируется относительно границы, при этом линия границы может перекрывать изображение.

`content-box` – фон позиционируется относительно содержимого элемента.

Значений может быть несколько (для каждого из множественных фоновых рисунков), при этом значения разделяются между собой запятой.

Если фон задан один, а значений `background-origin` несколько, то браузеры покажут разное поведение. Firefox и Opera используют первое значение, Chrome и Safari создадут несколько фоновых рисунков. Chrome до версии 3.0 и Safari до версии 5.0 поддерживают нестандартное свойство `-webkit-background-origin`. Firefox до версии 4.0 поддерживает нестандартное свойство `-moz-background-origin`.

`background-clip` – определяет, как цвет фона или фоновая картинка должна выводиться под границами. Эффект заметен при прозрачных или пунктирных границах.

Синтаксис:

`background-clip: [padding-box | border-box | content-box] [, [padding-box | border-box | content-box]]`

Может принимать следующие значения.

`padding-box` – фон отображается внутри границ.

`border-box` – фон выводится под границами.

`content-box` – фон отображается только внутри контента.

Значений может быть несколько (для каждого из множества фоновых рисунков), при этом значения разделяются между собой запятой.

Safari понимает только нестандартное свойство `-webkit-background-clip`, также с ним работает и Chrome. Opera не поддерживает значение `content-box`. Firefox до версии 4.0 поддерживает нестандартное свойство `-moz-background-clip`.

2.8 Стилизация границ элемента в CSS 3

Одной из интересных особенностей CSS 3 стала возможность создавать элементы с закругленными уголками, делая их более визуально красивыми, с помощью специального свойства `border-radius`.

Свойство `border-radius` устанавливает радиус скругления уголков рамки. Если рамка не задана, то скругление также происходит и с фоном.

Синтаксис:

```
border-radius: <радиус>{1,4} [ / <радиус>{1,4} ]
```

Разрешается использовать одно, два, три или четыре значения, перечисляя их через пробел. Также допустимо писать два значения через слэш (/). В качестве значений указываются числа в любом допустимом для CSS формате. В случае применения процентов, отсчет ведется относительно ширины блока.

Можно ввести следующие количества значений для свойства `border-radius`:

- 1 – радиус указывается для всех четырех уголков;
- 2 – первое значение задает радиус верхнего левого и нижнего правого угла, второе значение – верхнего правого и нижнего левого угла;
- 3 – первое значение задает радиус для верхнего левого угла, второе – одновременно для верхнего правого и нижнего левого, а третье – для нижнего правого угла;

4 – по очереди устанавливает радиус для верхнего левого, верхнего правого, нижнего правого и нижнего левого уголка.

В случае задания двух параметров через слэш, то первый задает радиус по горизонтали, а второй по вертикали (эллиптические уголки). На рисунке 10 показана разница между обычным скругленным уголком и эллиптическим уголком.

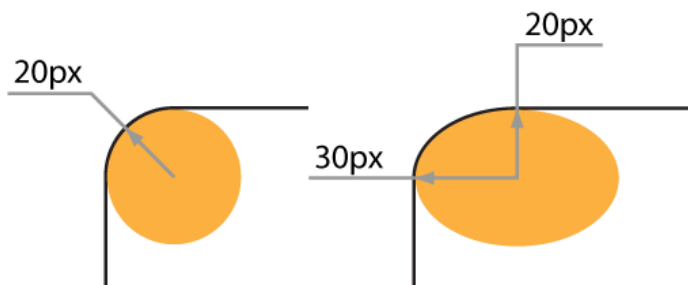


Рисунок 10 - Радиус скругления для создания разных типов уголков

Пример использования различных вариантов свойства `border-radius` приведен на рисунке 11. Использовались следующие значения (сверху вниз):

```
border-radius: 50px 0 0 50px;  
border-radius: 40px 10px;  
border-radius: 13em/3em;  
border-radius: 13em 0.5em/1em 0.5em;  
border-radius: 8px;
```

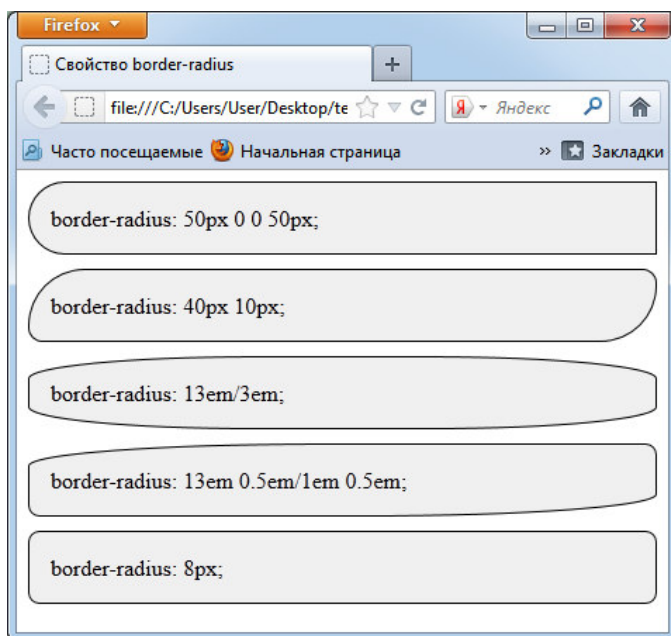


Рисунок 11 - Пример использования свойства border-radius

Chrome до версии 4.0, Safari до версии 5.0, iOS используют нестандартное свойство -webkit-border-radius.

Firefox до версии 4.0 использует нестандартное свойство -moz-border-radius.

Для управления отдельными границами используется следующая группа свойств:

- border-bottom-left-radius
- border-bottom-right-radius
- border-top-left-radius
- border-top-right-radius

Каждое свойство может иметь одно или два значения.

В качестве радиуса указывается любое допустимое в CSS значение (px, em и др.), а также проценты, в этом случае радиус скругления считается от ширины блока.

Необязательное второе значение предназначено для создания эллиптического уголка, первое значение при этом устанавливает радиус по горизонтали, а второе – радиус по вертикали.

Chrome до версии 4.0, Safari до версии 5.0, iOS используют нестандартные свойства вида `-webkit-border-bottom-left-radius` и т.д. Firefox до версии 4.0 использует свойства вида `-moz-border-radius-bottomleft` и т.д.

3 Практическое задание на курсовую работу

Цель работы – создание сайта с использованием технологий HTML 5 и CSS.

Исходные материалы:

- файл `shablon.html` – рекомендуемый шаблон страницы сайта;

- файл `styles.css` – таблица со стилями сайта;

- каталог с содержимым сайта, имеющий следующую структуру:

- файл «Информация.doc», содержащий текстовую информацию для занесения на сайт;

- каталог `pics`, содержащий изображения для сайта;

- каталог `disc`, содержащий изображения для страницы «Дискография»;

- каталог `media`, содержащий аудиофайлы для страницы «Музыка».

- файл «HTML 5 и CSS 3» – глава из пособия по Web-дизайну, описывающая новые технологии HTML и CSS;

- файл «Ссылки и панели навигации.doc» – глава из пособия по Web-дизайну, описывающая создание панелей навигации.

Структура страницы сайта:

Неофициальный сайт группы [название группы]	
Панель навигации	Основное содержимое
Сведения об авторе	

Состав страниц сайта:

- index.html – главная страница;
- bio.html – биография группы;
- group.html – состав группы;
- disc.html – дискография группы;
- media.html – примеры композиций группы;
- gal.html – галерея группы.

У каждой страницы свое основное содержимое, остальные элементы везде одинаковы.

Состав блока навигации:

Блок навигации должен иметь следующие ссылки:

- «Главная»;
- «Биография»;
- «Состав»;
- «Дискография»;
- «Музыка»;
- «Галерея».

Каждая ссылка должна быть оформлена в виде кнопки – создание ссылки-кнопки и вертикальной панели навигации см. в файле «Ссылки и панели навигации.doc».

Состав блока «Основное содержимое»

Страница index.html

Логотип группы
(изображение из каталога pics)

Текст из файла «Информация.doc»
(раздел «Общие сведения»)

Страница bio.html

История группы

Содержание

Раздел 1

Раздел 2

...

Раздел n

[Изображение]

Раздел 1

[Изображение]

Текст Раздела 1 (Берется из файла «Информация.doc», блок «История группы»)

...

Раздел n

[Изображение]

Текст Раздела n (Берется из файла «Информация.doc», блок «История группы»)

Замечание: Блок «Содержание» должен состоят из ссылок на якоря, которыми помечены названия разделов биографии группы.

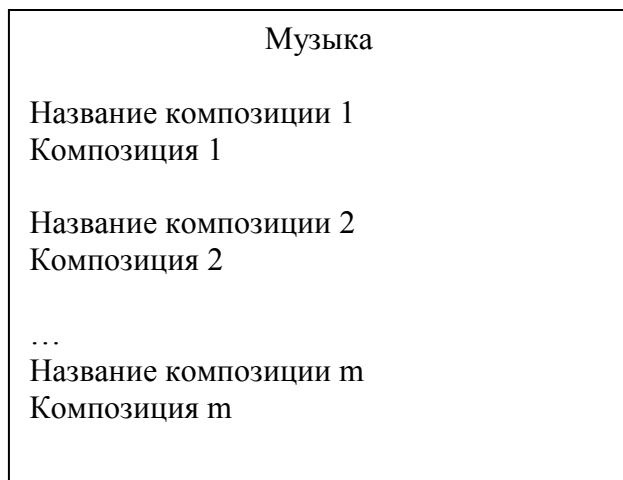
Состав группы Текст из файла «Информация.doc» (раздел «Состав группы») [Изображение]

Дискография	
[Обложка альбома] (берется из папки disc) Название альбома Год выпуска	[Обложка альбома] (берется из папки disc) Название альбома Год выпуска
[Обложка альбома] (берется из папки disc) Название альбома Год выпуска	[Обложка альбома] (берется из папки disc) Название альбома Год выпуска

Замечание: Блоки с отдельными альбомами рекомендует-ся оформить с помощью тегов <figure> и <figcaption> (см. файл «HTML 5 и CSS 3.doc», стр. 11-12).

Названия альбомов берутся из файла «Информация.doc», раздел «Дискография».

Страница media.html

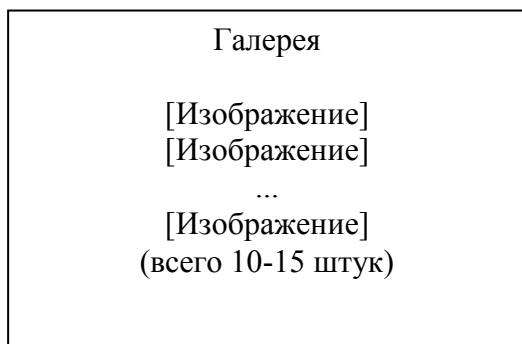


Замечание: Композиции добавляются на страницу с помощью тега <audio>. Для воспроизведения композиции как можно в большем количестве браузеров, следует добавить на страницу треки как в формате mp3, так и ogg.

Список композиций можно взять из файла «Информация.doc», раздел «Музыка».

Аудиофайлы следует взять из папки media.

Страница gal.html



Изображения на странице можете размещать и форматировать по своему усмотрению. Можно даже подключить скрипт для показа галереи или слайд-шоу – подобных вещей есть в свободном доступе в сети Интернет.

Требования к содержанию практической части пояснительной записки.

Пояснительная записка в практической части должна содержать следующее:

- общее описание сайта – тематика, перечень страниц, схема структуры сайта;
- наиболее важные фрагменты HTML- и CSS-кода и примеры созданных стилей для различных элементов веб-страниц;
- обзор сайта со скриншотами его страниц.

Список литературы

1 Макфарланд Д. Большая книга CSS / Д. Макфарланд. - СПб.: Питер, 2010. - 512 с.

2 Сергеева Т.И. Web-дизайн: учеб. пособие / Т.И. Сергеева, М.Ю. Сергеев. - Воронеж: ВГТУ, 2006. - 185 с.

3 Сергеев М.Ю. Разработка Web-сайтов: практикум: учеб. пособие / М.Ю. Сергеев. - Воронеж: ГОУВПО «ВГТУ», 2010, - 239 с.

4 Сергеев М.Ю. Web-дизайн: создание Web-сайтов с помощью HTML и CSS: учеб. пособие / М.Ю. Сергеев, Т.И. Сергеева. - Воронеж: ВГТУ, 2012. - 219 с.

ПРИЛОЖЕНИЕ А

Министерство образования и науки РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»
(ФГБОУ ВПО «ВГТУ»)

Факультет информационных технологий и компьютерной
безопасности
Кафедра «Автоматизированные и вычислительные системы»
Профиль «Вычислительные машины, комплексы,
системы и сети»

КУРСОВАЯ РАБОТА

по дисциплине «Web-дизайн»

Тема: «Проектирование и разработка веб-сайта с использованием технологий HTML 5 и CSS 3»

Расчетно-пояснительная записка

Разработал(а) студент(ка)

Подпись, дата Инициалы, фамилия

Руководитель

Подпись, дата Инициалы, фамилия

Нормоконтролер

Подпись, дата Инициалы, фамилия

Защищена _____ Оценка _____
Дата

Воронеж 2013

ПРИЛОЖЕНИЕ Б

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВПО «ВГТУ»)

Кафедра «Автоматизированные и вычислительные системы»

ЗАДАНИЕ
на курсовую работу

по дисциплине «Web-дизайн»

Тема: «Проектирование и разработка веб-сайта с использованием технологий HTML 5 и CSS 3»

Студент группы ВМ-111 Сорокин Александр Николаевич
Фамилия, имя, отчество

Номер варианта _____ Разработка сайта группы...

Технические условия: осуществить разработку веб-сайта средствами HTML и CSS.

Содержание и объем работы (графические работы, расчеты и прочее): изучение новых технологий Web-дизайна; проектирование структуры веб-сайта; реализация проекта веб-сайта; составление расчетно-пояснительной записки.

Сроки выполнения этапов: изучение технологий Web-дизайна; проектирование сайта – до 25.10.2013 г., реализация практического задания – до 25.11.2013 г., оформление расчетно-пояснительной записки – до 25.12.2013 г.

Срок защиты курсовой работы _____ январь 2013

Руководитель

Подпись, дата	Инициалы, фамилия
---------------	-------------------

Задание принял студент

Подпись, дата	Инициалы, фамилия
---------------	-------------------

ПРИЛОЖЕНИЕ В
Замечания руководителя

ПРИЛОЖЕНИЕ Г

Содержание

Задание на курсовую работу	2
Замечания руководителя	3
Введение	4
1 Теоретическая часть	5
1.1 Обзор стандарта HTML 5	5
1.2 Особенности селекторов CSS 3	8
2 Практическая часть	12
2.1. Структура веб-сайта	12
2.2. Реализация веб-сайта	15
2.3. Описание веб-сайта.....	18
Заключение	24
Список литературы	25

ПРИЛОЖЕНИЕ Д
Пример оформления списка литературы

Список литературы

- 1 Дронов В.А. Adobe Dreamweaver CS4 / В.А. Дронов. - СПб.: БХВ-Петербург, 2009. - 832 с.
- 2 Макфарланд Д. Большая книга CSS / Д. Макфарланд. - СПб.: Питер, 2011. - 560 с.
- 3 Дронов В.А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / В.А. Дронов. - СПб.: БХВ-Петербург, 2011. - 416 с.
- 4 Сергеева Т.И. Web-дизайн: учеб. пособие / Т.И. Сергеева, М.Ю. Сергеев. - Воронеж: ВГТУ, 2006. - 185 с.
- 5 Сергеева Т.И. Введение в Web-дизайн / Т.И. Сергеева, М.Ю. Сергеев. - Воронеж: ВГТУ, 2007. - 175 с.

Содержание

Введение	1
1 Методические рекомендации по выполнению курсовой работы	1
1.1 Требования к содержанию пояснительной записки	1
1.2 Требования к оформлению пояснительной записки ...	3
2 Теоретические сведения	4
2.1 Общие сведения о HTML 5	4
2.2 Теги для отображения мультимедийных объектов	5
2.3 Теги для обозначения блоков страницы	11
2.4 Общие сведения о CSS 3	14
2.5 Селекторы в CSS 3	15
2.6 Форматирование текста в CSS 3	20
2.7 Работа с прозрачностью и фоном элемента в CSS 3 .	25
2.8 Стилизация границ элемента в CSS 3	31
3 Практическое задание на курсовую работу.....	34
Список литературы	39
Приложения	41

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА WEB-САЙТОВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ HTML 5 И CSS 3

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к выполнению курсовой работы
по дисциплине «Web-дизайн»
для бакалавров направления 230100 «Информатика
и вычислительная техника», профиля
«Вычислительные машины, комплексы, системы и сети»
очной формы обучения

Составители: Сергеев Михаил Юрьевич
Сергеева Татьяна Ивановна

В авторской редакции

Подписано в печать 26.06.2013.
Формат 60х84/16. Бумага для множительных аппаратов.
Усл. печ. л. 2,9. Уч.-изд. л. 2,7. Тираж 25 экз. «С» .
Заказ № .

ФГБОУ ВПО «Воронежский государственный
технический университет»
394026 Воронеж, Московский просп., 14.