



Урок 2

ОСНОВЫ HTML

Спецсимволы. Комментарии в HTML. Гиперссылки. Типы изображений для web. Загрузка изображений на страницу. Списки, формы и их элементы.

[Спецсимволы](#)

[Комментарии](#)

[Гиперссылки](#)

[Виды ссылок](#)

[Якоря в гиперссылках](#)

[Изображения](#)

[JPEG](#)

[GIF](#)

[PNG](#)

[SVG](#)

[Загрузка изображений на страницу](#)

[Списки](#)

[Структура вложенного списка](#)

[Формы и их элементы](#)

[Структура форм](#)

[Элементы форм](#)

[Тег <label>](#)

[Практика](#)

[Гиперссылки](#)

[Элементы форм](#)

[Практическое задание](#)

[Дополнительные материалы](#)

[Используемая литература](#)

[http://htmlbook.ru/html/img.](http://htmlbook.ru/html/img)

Спецсимволы

Используя спецсимволы в HTML, можно заменять уже имеющиеся на клавиатуре компьютера символы либо указать символы, которые отсутствуют на клавиатуре, например значок копирайта.

Любые теги никак не показываются в окне браузера, поскольку воспринимаются им как команды для вставки элементов и изменения их свойств. Но иногда требуется на веб-странице вывести теги, например для демонстрации HTML-кода. В этом случае используйте спецсимволы `<` и `>` для замены угловых скобок `<` и `>`.

Вот некоторые варианты спецсимволов:

Код в HTML	Внешний вид	Описание
<code>&nbsp;</code>		неразрывный пробел
<code>&copy;</code>	©	знак copyright
<code>&quot;</code> и <code>&raquo;</code>	« и »	двойные кавычки
<code>&lt;</code> и <code>&gt;</code>	<code><</code> и <code>></code>	символы
<code>&prime;</code>	'	одиночный штрих

Комментарии

В HTML-документах можно оставить какой-либо комментарий, например, название части страницы, который не будет отображен в браузере. Но любой желающий сможет увидеть этот текст, если воспользуется функцией просмотра кода страницы в любом браузере.

Комментарии можно применять для того, чтобы не отображать теги в браузере. Это может пригодиться при редактировании веб-страницы, при этом вы можете комментировать целые блоки кода.

Пример:

```
<!--Текст, который не будет отображаться на сайте-->
<p>Текст основной части сайта</p>
```

Гиперссылки

Гиперссылка – активный элемент (текст, изображение, кнопка и т.п.) одного документа, при нажатии на который происходит переход к другому документу или к его элементу. Документы, содержащие

гиперссылки, называются гипертекстовыми. Гиперссылка может быть добавлена к любому элементу гипертекстового документа и обычно выделяется графически.

В HTML-документах текстовые ссылки по умолчанию выделяются синим цветом, при наведении на них курсора мыши в окне браузера меняют цвет или выделяются подчеркиванием. Посещенная ранее ссылка обычно выделяется цветом, отличным от цвета непосещенной ссылки.

Тег ссылки называется `<a>`, и у него есть обязательный атрибут `href`, в значении которого мы указываем на файл, место в документе или страницу другого сайта, куда хотим перейти. В содержимом этого тега необходимо написать текст, который увидит пользователь.

Виды ссылок

Относительные ссылки используются для перемещения внутри документа или сайта.

```
<a href="file_name.html">
    текст ссылки, который видит пользователь
</a>
```

Абсолютные ссылки используются для перехода на страницы внешнего сайта. Для этого в значении атрибута `href` нужно указать полный путь до той страницы, на которую мы хотим перейти, включая тип протокола.

```
<a href="http://yandex.ru" target="_blank">
    страница yandex.ru откроется в новой вкладке
</a>
```

У тега ссылки существует атрибут `target`. Если указать `target="_blank"`, то страница откроется в новой вкладке браузера. Раньше это значение официально не поддерживалось в HTML, но люди все равно использовали его, поскольку оно работало. Теперь в HTML5 оно полностью поддерживается.

Когда использовать `target="_blank"`:

- Если необходимо открыть сторонний ресурс и не потерять страничку вашего сайта.
- Если пользователь работает над чем-то, что может быть потеряно, если текущая страница изменится.

Когда не использовать `target="_blank"`:

- В меню (иначе пункты меню будут открываться в новых вкладках).
- Из-за личных предпочтений.
- Если клиент думает, что так будет лучше.

Якоря в гиперссылках

При помощи гиперссылок можно перемещаться не только между страницами документа, но и внутри данной страницы. Это бывает очень полезно, когда ваша статья состоит из нескольких разделов и имеет довольно объемное содержание.

В этом случае разумно использовать так называемые якоря, чтобы быстро переходить в интересующий раздел. В том элементе, на который следует перейти, нужно определить уникальный идентификатор.

Для этого существует атрибут `id`, который можно определять практически для любого тега. Название этого атрибута может быть любым и обязательно уникальным. В значении атрибута `href` тега-ссылки ставится символ решетки, после которого указывается имя того идентификатора, к которому необходимо обратиться.

```
<a href="#p10">Прочитать 10 параграф</a>
<p id="p1">Текст параграфа №1</p>
<p id="p2">Текст параграфа №2</p>
...
<p id="p10">Вот и долгожданный параграф №10</p>
```

Изображения

В настоящее время в веб-документах используются четыре формата изображений: `jpeg`, `gif`, `png`, `svg`. Рассмотрим каждый из них подробнее.

JPEG

Плюсы:

- Небольшой размер файла.
- Поддерживает 16 млн цветов.
- Можно управлять качеством изображения при сохранении.

Минусы:

- При сжатии размеров теряется качество.
- Не поддерживает прозрачность.

GIF

Плюсы:

- Поддерживает анимацию.
- Поддерживает прозрачность.
- При сжатии не теряет в качестве.

Минусы:

- Количество цветов до 256.

PNG

Растровый графический формат PNG, набирающий всё большую популярность, появился в 1995 году как замена GIF. Формат PNG, в свою очередь, подразделяется на форматы PNG-8 (аналог формата GIF, за исключением того, что PNG-8 не поддерживает анимацию) и PNG-24.

Еще существует формат PNG-32 – в этом формате растровой графики для одного пикселя возможно использовать четыре байта информации. Три отвечают за формирование полноцветного изображения, а четвертый байт выделяется отдельно для формирования альфа-канала, позволяющего использовать PNG-32 для получения картинок с прозрачным фоном.

Проблема поддержки 32-битной (полной) прозрачности картинки формата PNG в Microsoft Internet Explorer была решена в седьмой версии браузера.

PNG – 8

Плюсы:

- Поддерживает прозрачность.
- При сжатии не теряет в качестве.

Минусы:

- Количество цветов до 256.

PNG – 24

Плюсы:

- При сжатии не теряет в качестве.
- Использует 16 млн цветов.
- Плавный переход от прозрачной области к цветной.

Минусы:

- Большой размер файла.

SVG

SVG – формат векторной графики, файлы SVG можно читать и редактировать (при наличии некоторых навыков) при помощи обычных текстовых редакторов. Существует возможность увеличить любую часть изображения SVG без потери качества.

Широко доступно использование растровой графики в SVG-документах. Имеется возможность вставлять элементы с изображениями в форматах PNG, GIF или JPG. Текст в графике SVG является текстом, а не изображением, поэтому его можно выделять и копировать, он индексируется поисковыми машинами, не нужно создавать дополнительные метафайлы для поисковых роботов.

Плюсы:

- Небольшие объемы файлов, отличное сжатие.
- Возможно увеличить любую часть изображения SVG без потери качества.

Минусы:

- Большой размер изображения.
- Сложность использования.

Из рассмотренных сильных и слабых сторон форматов изображений можно сделать вывод. Для сохранения фотографий с четкими краями лучше всего подходит формат JPEG, так как в нем размер файла получается небольшим. Для анимированных изображений единственный вариант – формат GIF, а для сохранения качественных изображений, в которых используется прозрачность, лучше всего подойдет формат PNG-24. Для векторной графики идеально подойдет SVG.

Загрузка изображений на страницу

Для отображения изображений на странице существует тег ``, который является одиночным. У этого тега есть обязательный атрибут `src`, в значении которого указывают путь к изображению. Этот путь может быть как относительным, так и абсолютным.

```

```

У тега `` можно указать дополнительные атрибуты.

Атрибут `alt` означает альтернативный текст. Используется он для того, чтобы: во-первых, если картинка не загрузится на страницу, пользователь увидит тот текст, который вы указали в этом атрибуте, во-вторых, альтернативный текст необходим при продвижении сайта, в нем можно указать ключевые слова, и тогда это изображение будет участвовать в поиске по картинкам.

Следующий атрибут – `title`, который является универсальным, его можно использовать практически для любого тега. То, что вы напишете в его значении, будет выводиться в виде всплывающей подсказки.

Любому изображению можно задать ширину и высоту, указав эти значения в атрибутах `width` и `height` соответственно. При помощи этих атрибутов изображение лучше не увеличивать, иначе получится очень плохое качество. Если задать ширину и высоту изображению, браузер при загрузке страницы будет сразу выделять заданную область под картинку.

Всегда храните все изображения в отдельной папке, чтобы знать, где искать то или иное изображение.

Списки

Списки в HTML-документах бывают двух видов: маркированные и нумерованные. Отличаются они тем, что у маркированных элементы списка начинаются с маркеров, а у нумерованного – с цифр или букв.

Структура маркированного списка:

```
<ul>
  <li>Первый элемент списка</li>
  <li>Второй элемент списка</li>
  <li>Третий элемент списка</li>
</ul>
```

Структура нумерованного списка:

```
<ol>
  <li>Первый элемент списка</li>
  <li>Второй элемент списка</li>
  <li>Третий элемент списка</li>
</ol>
```

Как видно из примеров, структура обоих типов списков одинаковая, единственное различие в том, что в случае с маркированным списком контейнер будет называться ``, а в нумерованном – ``. В нумерованном списке иногда может потребоваться начать список с какой-нибудь определенной цифры или буквы. Для этого существует атрибут `start`, в значении которого указывается, с какого элемента будет начинаться список. В списках есть возможность менять вид маркеров, например, арабские цифры изменить на римские, либо вообще убрать маркеры.

Структура вложенного списка

```
<ol>
  <li>Первый элемент списка</li>
  <li>Второй элемент списка
    <ul>
      <li>Первый элемент вложенного списка</li>
      <li>Второй элемент вложенного списка</li>
      <li>Третий элемент вложенного списка</li>
    </ul>
  </li>
  <li>Третий элемент списка</li>
</ol>
```

Вложенные списки предназначены для организации сложной иерархической структуры текста, обычно юридических или технических документов. Также вложенные списки используются для создания многоуровневых меню и навигации по сайту.

Формы и их элементы

Формы необходимы для того, чтобы было возможно получать от пользователя информацию, отправлять эти данные на сервер, а там уже их обрабатывать. Документ может содержать любое количество форм, но одновременно на сервер может быть отправлена только одна. По этой причине данные форм должны быть независимы друг от друга.

Допускается внутри контейнера `<form>` помещать другие теги, при этом сама форма никак не отображается на веб-странице, видны только ее элементы и результаты вложенных тегов.

Структура форм

Все элементы форм должны обязательно находиться внутри контейнера `<form>`. В этом теге могут находиться атрибуты, необходимые для правильной работы формы.

Элемент `<fieldset>` предназначен для группирования элементов формы. Такая группировка облегчает работу с формами, содержащими большое число данных. Например, один блок может быть предназначен для ввода текстовой информации, а другой – для флажков. Внутри тега `<legend>` помещают название данной формы. Тогда браузер выделит эту часть рамкой, в которой будет находиться название.

Элемент `<legend>` не обязателен, но если присутствует, должен идти сразу же после тега `<fieldset>`. Другие теги или текст перед `<legend>` недопустимы. Внутри `<legend>` можно использовать теги форматирования вроде ``, `<i>`


```
<form action="URL">
<fieldset>
    <legend>Контактные данные</legend>
    ...
</fieldset>
</form>
```

Элементы форм

В HTML существует три основных тега элементов форм:

1. `<textarea>` используется для того, чтобы сформировать многострочное поле ввода для большого количества информации. С помощью атрибута `cols` можно задать ширину этого поля, а с помощью атрибута `rows` указать число строк. Тег `<textarea>` парный.
2. `<select>` представляет собой выпадающий список, элементы которого указываются в теге `<option>`. У этого тега существует несколько атрибутов. Чтобы сделать возможность выбора нескольких пунктов, необходимо задать атрибут `multiple`, в значении которого также необходимо указать `multiple`. В атрибуте `size` указывается число строк выпадающего списка, которые увидит пользователь в браузере. Если необходимо, чтобы какой-то из элементов выпадающего списка был выбран, нужно этому элементу в теге `<option>` задать атрибут `selected`. Тег `<select>` также парный.
3. Тег `<input />` в зависимости от значения атрибута `type` будет выглядеть по-разному и иметь разные значения. Тег `<input />` одиночный.

Рассмотрим более подробно тег `<input />`

```
<input type="text" size="30" placeholder="Ваше Имя">
<input type="password" size="8">
<input type="checkbox" checked="checked">
<input type="radio" name="radio">
<input type="file">
<input type="submit" value="Сохранить">
<input type="reset" value="Очистить">
<input type="button" value="просто кнопка">
```

У первого тега `<input />` значение атрибута `type="text"`. Это означает, что это обычное однострочное поле ввода. В значении атрибута `size` указывается размер этого поля в символах. Атрибут `placeholder` выводит текст внутри текстового поля, который исчезает при начале ввода символов.

`<input />` со значением `password` выглядит так же, как и предыдущий, но используется для ввода пароля, и все символы будут показываться как точки, чтобы пароль нельзя было подсмотреть.

Следующие два варианта – `<input type="checkbox" />` и `<input type="radio" />` – предоставляют пользователю возможность выбора некоторых пунктов. В первом случае есть возможность множественного выбора нескольких пунктов в виде флажков, а во втором – единственного выбора в виде переключателей. В обоих случаях можно отметить любой пункт по умолчанию, если задать атрибут `checked="checked"`.

`<input type="file" />` предоставляет пользователю возможность выбора файла на своем компьютере, чтобы загрузить его на сервер.

Оставшиеся 3 тега `<input />` – кнопки. `submit` отправляет данные на сервер, `reset` предназначена для того, чтобы очистить все поля в форме. А третья `type="button"` – просто кнопка, при нажатии на которую ничего не произойдет, но ее, например, можно привязать к событию javascript. У всех этих кнопок в значении атрибута `value` указывается текст, который будет отображен на кнопке.

Тег `<label>`

Тег `<label>` устанавливает связь между определенной меткой, в качестве которой обычно выступает текст, и элементом формы (`<input>`, `<select>`, `<textarea>`). Такая связь необходима, чтобы изменять значения элементов формы при нажатии курсором мыши на текст. Кроме того, с помощью `<label>` можно устанавливать горячие клавиши на клавиатуре и переходить на активный элемент подобно ссылкам.

Существует два способа связывания объекта и метки. Первый заключается в использовании идентификатора `id` внутри элемента формы и указании его имени в качестве атрибута `for` тега `<label>`. При втором способе элемент формы помещается внутрь контейнера `<label>`.

```
<!-- Первый способ -->
<input id="идентификатор">
<label for="идентификатор">Текст</label>
<!-- Второй способ -->
<label><input type="...">Текст</label>
```

Практика

Гиперссылки

```
<!-- Создание меню-->
<ul>
  <li>Главная</li>
  <li><a href="catalog.html">Каталог</a></li>
  <li><a href="contacts.html">Контакты</a></li>
</ul>
```

Элементы форм

```
<form action="">
<fieldset>
  <legend>Страница регистрации</legend>
  <input type="text" size="30" placeholder="Ваше Имя">
  <input type="password" size="8">
  <input type="checkbox" checked="checked">
  <input type="radio" name="radio">
  <input type="file">
  <input type="submit" value="Сохранить">
  <input type="reset" value="Очистить">
  <input type="button" value="просто кнопка">
</fieldset>
</form>
```

Практическое задание

Создание основного наполнения нашего сайта, главной страницы, каталога, товара каталога и контактов.

1. Создать файл index.html – это будет главная страница. На этой странице сделайте ссылки в виде маркированного списка на следующие страницы:
 - a. Меню (должно располагаться на каждой странице сайта).
 - i. Главная.
 - ii. Каталог.
 - iii. Контакты.
 - b. Отделить горизонтальной чертой шапку сайта.
 - c. Добавить название вашего магазина.
 - d. Напишите приветственное сообщение.
 - e. Горизонтальная черта.
 - f. Подвал (должен располагаться на каждой странице сайта).
2. Что должно присутствовать в footer:
 - a. Значок копирайта.
 - b. Текст «Все права защищены».
3. На странице «Каталог»:
 - a. Меню сайта.
 - b. Горизонтальная черта.
 - c. Заголовок «Каталог».
 - d. Уменьшенные копии изображений товаров из вашего магазина.
 - e. Ссылки под картинками для перехода в подробное описание товара.
 - f. Footer.
4. На странице «Товар каталога» (страницы из первого ДЗ):
 - a. Добавить картинку товара с возможностью нажатия на нее. Картинка должна открыться в полном размере, в новом окне.
 - b. В подразделе «Характеристики товара» необходимо добавить маркированный или нумерованный список характеристик товара.
5. На странице «Контакты»:
 - a. Меню сайта.
 - b. Горизонтальная черта.
 - c. Заголовок «Напишите нам».
 - d. Поля для заполнения:
 - i. Имя.
 - ii. E-mail.

- iii. Тема.
 - iv. Текстовое поле.
 - v. * Произвольные поля, которые на ваше усмотрение нужны на вашем сайте.
- е. Заголовок «Адрес».
- i. Контактный номер телефона интернет-магазина.
 - ii. Адрес.
 - iii. E-mail.
 - iv. * Любая информация о магазине.
6. * На страницу контактов добавьте реальную карту Yandex или Google.
7. * Если вы сделали несколько товаров, добавьте описание и характеристики для каждого товара.
8. * Количество страниц товаров не ограничено.

Задачи со * предназначены для продвинутых учеников, которым мало сделать обычное ДЗ.

Дополнительные материалы

[Статья про типограф, с чего всё начиналось и для чего же он нужен.](#)

[Чуть подробнее про гипертекст.](#)

[Какие изображения применяются для html.](#)

[Формы html.](#)

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://htmlbook.ru/samhtml/tekst/spetssimvoly>.
2. <http://htmlweb.ru/html/symbols.php>.
3. <http://htmlbook.ru/html/a>.
4. <http://htmlbook.ru/html/img>.
5. <http://htmlbook.ru/html/form>.