

# Урок №2. Основы HTML.

---

## Содержание урока

- Структура HTML документа
- Основные теги оформления текста
- Спецсимволы
- Гиперссылки
- Изображения
- Списки
- Формы и их элементы

## Структура HTML документа

Каждая страница в HTML-документе состоит из 3-х обязательных элементов:

1. Объявляется тип документа (подробнее о типах документов в последнем уроке).
2. Раздел `<head>`, в котором размещается служебная информация, например, объявляются meta-теги, которые могут иметь разное назначение (кодировка, описание страницы, ключевые слова), подключаются файлы каскадных листов стилей (css), подключаются файлы скриптов. Все что находится в данном разделе не видно пользователю, за исключением содержимого тега `<title>`, в котором указывается название веб-страницы. Обязательно указывайте название каждой странице в теге `<title>`, причем давайте ей адекватное название, например: «Мегабыт – интернет-магазин бытовой техники | Каталог товаров | Моющий пылесос Sharp».
3. Раздел `<body>`, в котором уже располагаются все элементы, который видит пользователь. Именно с этим разделом мы, в основном, и будем работать.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Мой первый документ</title>
<meta http-equiv="Content-Type" content="text/html; charset=win-1251">
</head>
<body>
...
</body>
</html>
```

# Основные теги оформления текста

## Заголовки

Так же как и в газетах и журналах, в HTML-документе любая статья, новость и т.д. должна начинаться с заголовка. Предусмотрено 6 уровней заголовков, 1-й из которых будет отображен самым крупным шрифтом, а далее, чем выше уровень, тем размер шрифта будет уменьшаться.

Для отображения заголовков существует тег `<h>`, и указывается цифра от 1 до 6, которая соответствует уровню заголовка. Как видно из примера, тег заголовка, является парным тегом, т.е. обязательно должен быть закрыт.

```
<h1>Заголовок первого уровня</h1>
```

```
<h2>Заголовок второго уровня</h2>
```

```
<h3>Заголовок третьего уровня</h3>
```

```
<h4>Заголовок четвертого уровня</h4>
```

```
<h5>Заголовок пятого уровня</h5>
```

```
<h6>Заголовок шестого уровня</h6>
```

## Параграфы

Далее, после заголовка обычно следует какой-нибудь текст, который необходимо заключать в параграфы, или иначе сказать, абзацы. При составлении документа старайтесь выделять блоки текста в отдельные параграфы, как это сделано в книгах, в противном случае, может получиться сплошной текст, который очень трудно будет читать посетителю вашего сайта.

В HTML для параграфов используется тег `<p>`, который также является парным, и внутри него помещается тот текст, который нужно отобразить.

```
<p>Здесь мы напишем первый параграф.</p>
```

```
<p>А здесь мы напишем второй.</p>
```

```
<p>И для третьего место тоже найдется.</p>
```

## Теги выделения текста

Иногда при написании, например, статьи необходимо выделить слово, словосочетание, предложение или целый участок текста, чтобы привлечь внимание читателя (или поискового робота). А для того, чтобы это сделать, можно выделить фрагмент жирным, курсивом, или подчеркнуть. Кстати, с подчеркиванием текста будьте осторожны, т.к. принято по стандартам, что подчеркнуты гиперссылки, и многие пользователи уже настолько к этому привыкли, что когда видят подчеркнутый текст, то им хочется «кликнуть» по этому участку текста, ожидая перейти на другую страницу.

Для того, чтобы выделять текст существуют специальные теги:

<code>&lt;b&gt;...&lt;/b&gt;</code>		текст будет
<code>&lt;strong&gt;...&lt;/strong&gt;</code>		<b>жирным</b>
<code>&lt;i&gt;...&lt;/i&gt;</code>		в этом случае
<code>&lt;em&gt;...&lt;/em&gt;</code>		<i>курсивное начертание</i>
<code>&lt;u&gt;...&lt;/u&gt;</code>		<u>подчеркнутый</u>

Как видно из примера, можно выделить текст жирным используя теги `<strong>` или `<b>`, а курсивным - `<em>` или `<i>`. В чем же тогда разница и какие из этих вариантов. Разница в том, что:

- `<strong>` и `<em>` - теги для логического выделения текста.
- `<b>` и `<i>` - для визуального выделения.

По поводу, какие из вариантов нужно использовать, однозначного ответа нет, просто *не рекомендуется* использовать теги `<b>` и `<i>`. Возможно, что в будущем эти теги исключат из спецификации HTML.

Еще примеры основных HTML тегов:

H`<sub>2</sub>`O - H<sub>2</sub>O

M`<sup>2</sup>` - м<sup>2</sup>

`<pre>`

Я Московский озорной гуляка  
По всему Тверскому околотку

`</pre>`

Теги <sub> и <sup> нужны для того, чтобы сделать нижние и верхние индексы соответственно. Тег <pre> необходим тогда, когда необходимо вывести в браузер текст в том виде, в котором он был набран в текстовом редакторе, т.е. со всеми пробелами и отступами.

## Вложенные теги

Очень часто при верстке веб-страниц необходимо вкладывать одни теги в другие. И это вложенность может достигать несколько уровней. Поэтому следует запомнить одно простое правило вложенных тегов: закрывать теги необходимо именно в той последовательности в которой вы их открывали.

```
<p>Использовать вложенность тегов
  <b>легко
    <i>и просто</i>
  </b>
</p>
```

Обратите внимание, так как тег <i> был открыт последним, то закрываем его первым, а далее уже все остальные теги по очереди. Еще один момент, следите за тем, чтобы открывающий и закрывающий теги находились на одном уровне.

## Спецсимволы

Используя спецсимволы в HTML можно либо заменять уже имеющиеся на клавиатуре компьютера символы, либо при помощи них можно указать символы, которые отсутствуют на клавиатуре, например значок копирайта. Синтаксис спецсимволов довольно простой, сначала ставим значок амперсанта (&), далее указывается код спецсимвола, обычно это сокращенное название символа, а в конце ставим точку с запятой.

Вот некоторые варианты спецсимволов (Полный список можно найти в справочниках).

&copy;	- значок копирайта
&lt; и &gt;	- символы < и >
&laquo; и &raquo;	- символы « и »
&larr; и &rarr;	- символы ← и →
&nbsp;	- неразрывной пробел
&mdash;	- длинное тире

## Комментарии

Комментарии в HTML-документах используются для того, чтобы можно было оставить какой-либо комментарий, например, название части страницы, который не будет отображен в браузере. Но учтите, что любой желающий сможет увидеть этот текст, если воспользуется функцией просмотр кода страницы в любом браузере. Комментарии еще 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.

```
<!-- Здесь можно писать комментарий, который не будет отображаться в документе -->
```

## Гиперссылки.

Гиперссылки могут связывать страницы как в пределах одного сайта, так и указывать на любую страницу в Интернете. Рассмотрим синтаксис гиперссылки:

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

Тег ссылки называется `<a>` и у нее есть обязательный атрибут `href`, в значении которого мы указываем на тот файл, на то место в документе, или на ту страницу другого сайта, куда хотим перейти. В содержимом этого тэга необходимо написать тот текст, который увидит пользователь. По умолчанию, это текст синего цвета и подчеркнут.

Ссылки могут быть двух видов:

1. Относительные ссылки, которые используются для перемещения внутри нашего док-та.

```
<a href="orders/order.html">  
  посмотреть заказ, который находится в папке orders  
</a>
```

2. Абсолютные ссылки. Они 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="orders/order.html" target="_blank">  
    страница yandex.ru откроется в новой вкладке  
</a>
```

У тега ссылки существует атрибут - target. Если указать target="\_blank" , то это означает, что данная страница откроется в новой вкладке браузера.

## 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 будет использовать так называемые якоря, чтобы быстро переходить в интересующий нас раздел. В том элементе, на который нам следует перейти, нужно определить уникальный идентификатор. Для этого существует атрибут id, который можно определять практически для любого HTML тэга. Название этого атрибута может быть любым и обязательно должно быть 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>  
  
<h1 id="p10">  
    Вот и долгожданный параграф  
</h1>
```

## Изображения

В настоящее время в веб-документах используются 3 формата изображений, это jpeg, gif, png. Рассмотрим каждый из этих форматов подробнее.

### **Формат JPEG**

*Плюсы:*

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

*Минусы:*

- При сжатии размеров теряется качество
- Не поддерживает прозрачность

## **Формат GIF**

*Плюсы:*

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

*Минусы:*

- Кол-во цветов до 256

## **Формат PNG**

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, но он пока используется достаточно редко и имеет большой размер файла.

### **PNG – 8**

*Плюсы:*

- Поддерживает прозрачность
- При сжатии не теряет в качестве

*Минусы:*

- Кол-во 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

Из рассмотренных сильных и слабых сторон рассмотренных 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

– это использовать формат gif, а для сохранения качественных изображений, в которых используется прозрачность лучше всего подойдет формат png-24.

## Загрузка изображений на страницу

Рассмотрим, каким образом изображения появляются на нашей странице. А для этого существует тег `<img>`, который является одиночным тегом. У этого тега есть обязательный атрибут `src`, в значении которого мы указываем путь к изображению. Этот путь может быть как относительным так и абсолютным.

```

```

У тега `<img />` можно указать дополнительные атрибуты.

Атрибут `alt` означает альтернативный текст. Используется он для того, чтобы: во-первых, если картинка не загрузится на страницу, то пользователь увидит тот текст, который вы указали в этом атрибуте, и 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 - это `title`, который является универсальным, и его можно использовать практически для любого тега. То, что вы напишите 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, а то получится очень плохое качество. Если задать ширину и высоту изображению, то браузер при загрузке страницы будет сразу выделять заданную область под картинку.

Всегда храните все изображения в отдельной папке, тогда вы будете знать, где искать то или иное изображение

## Списки

Списки в html-документах бывают двух видов: маркированные и нумерованные. Отличаются они тем, что у маркированных элемент списка начинается с маркера, а у нумерованного с цифр или букв.

## Структура маркированного списка

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

## Структура нумерованного списка

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

Как видно из примеров структура обоих типов списков одинаковая, и единственное различие в том, что в случае с маркированным списком, контейнер будет называться `<ul>`, а в нумерованном - `<ol>`.

В нумерованном списке иногда может потребоваться начинать 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, или арабские цифры изменить, например на римские, либо вообще убрать маркеры. Для этого лучше использовать `css`, что и будем делать в следующем уроке.

## Формы и их элементы

Формы необходимы для того, чтобы было возможно получать от пользователя какую-либо информацию, отправлять эти данные на сервер, а там уже их обрабатывать. В данном курсе не рассматривается то, как надо обрабатывать данные полученные из форм, так как для этого необходимо уже знать язык программирования. В рамках этого курса вы научитесь просто выводить различные элементы форм на веб-страницы.

### Структура форм

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

В значении атрибута `action` указывается тот файл, который будет обрабатывать данную форму, атрибут `method` может двух вариантов: `post` или `get`. Атрибут `enctype="multipart/form-data"` необходим для того, чтобы загружать файлы.

Для того, чтобы отделить форму или ее часть, для этого можно воспользоваться тегом `<fieldset>`, а внутрь тэга `<legend>` поместить название данной формы или

ее части. Тогда браузер выделит эту часть рамкой, в которой будет находиться название.

```
<form action="file.php"
      method="post"
      enctype="multipart/form-data">
  <fieldset>
    <legend>Контактные данные</legend>
    ...
  </fieldset>
</form>
```

## Элементы форм

В html существует 3 основных тега элементов форм

```
<textarea cols="30" rows="5" name="title" >
</textarea>

<select name="year" multiple="multiple" size="2">
  <option value="1">1981</option>
  <option value="2" selected="selected">1982</option>
</select>
```

1. Тег `<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 данного поля. Тег `<textarea>` является парным тэгом.

2. Элемент `<select>` представляет собой выпадающий список, элементы которого указываются в тэге `<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 существует необходимость, чтобы какой-то из элементов выпадающего списка был выбран, то нужно этому элементу в тэге `<option>` задать атрибут `selected`. Тег `<select>` также является парным тэгом.

3. Тег `<input />`. Он в зависимости от значения атрибута `type` будет выглядеть по-разному и иметь разные значения. Тег `<input />` является уже одиночным тэгом.

```
<input type="text" size="50" />
<input type="password" size="30" />
<input type="checkbox" checked="checked" />
<input type="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.

`<input />` со значением `password` атрибута `type` выглядит также, как и предыдущий, но он используется для ввода пароля и все символы будут в виде точек, т.е. зашифрованы.

Следующие два варианта `<input type="checkbox" />` и `<input type="radio" />` предоставляют пользователю возможность право 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 есть возможность отметить любой пункт по умолчанию, если задать атрибут `checked="checked"`.

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

Оставшиеся 3 тега `<input />` - это кнопки. При нажатии на кнопку `type="submit"` все данные, введенные в данной форме, будут отправлены на сервер. Вторая кнопка `type="reset"` предназначена для того, чтобы очистить все поля в форме. А третья `type="submit"` это просто кнопка, при нажатии на которой ничего не произойдет, но ее, например, можно привязать к событию javascript. У всех этих кнопок в значении атрибута `value` указывается тот текст, который будет отображен на кнопке.

## Тег `<label>`

При помощи тега `<label>` устанавливается связь между определенным элементом формы и той надписи, которая соответствует данному элементу. Особенно важно использовать этот тег, когда оформляются флажки или переключатели, так как, если пользователь щелкнет по названию пункта, этот пункт отметится.

```
<label>
  <input type="radio" />
  Выбери меня<br />
</label>
<label>
  <input type="radio" />
  А лучше меня<br />
</label>
```

## Домашнее задание

1. Создать файл index.html – это будет главная страница. На этой странице сделайте ссылки в виде маркированного списка на следующие страницы:
  - Статьи
  - Каталог
  - Галерея изображений
  - Регистрация
  - Контакты
2. Создайте соответствующие страницы, причем на этих страницах обязательно должны быть ссылки на все остальные, и каким-либо образом выделяйте соответствующую страницу, чтобы пользователю было понятно, где он находится. Внизу каждой страницы выделите «footer», где поставьте значок копирайта.
3. На странице «Статьи» перечислите 2-3 статьи в следующем виде:

### [Заголовок первой статьи.](#)

Краткое содержание первой статьи... >>> [читать далее](#)

[Перейти к комментариям](#)

---

### [Заголовок второй статьи.](#)

Краткое содержание второй статьи... >>> [читать далее](#)

[Перейти к комментариям](#)

---

Тексты статей произвольные, найдите в интернете. Заголовок статьи должен быть в виде ссылки. При клике по заголовку и по ссылке «читать далее» открывается конкретная статья, где у статьи есть заголовок, текст статьи (попытайтесь найти статьи объемней, чтобы появлялась вертикальная полоса прокрутки). А внизу статьи должны быть комментарии, причем при нажатии на ссылку «перейти к комментариям» в общем списке статей, можно было сразу прочитать эти комментарии. Статьи должны находиться в отдельной папке.

4. На странице «Каталог» перечислите в виде списка некоторые произвольные категории товаров:
- Цветы
    - Живые цветы
    - Искусственные цветы
  - Игрушки
    - Мягкие игрушки
    - Конструкторы
  - Книги
    - Аудиокниги
    - Книги в мягком переплете
5. В «галерее изображений» выведите на экран несколько изображений разных форматов, чтобы можно было увидеть, какого формата данное изображение. Причем в галерее должны быть уменьшенные копии изображений, а в полном размере картинку можно посмотреть в отдельном окне.
6. Страница «Регистрация» представляет из себя форму, в которой должны быть следующие поля:
- Фамилия
  - Имя
  - Отчество
  - Добавьте фотографию
  - Адрес электронной почты
  - Пароль
  - Адрес сайта (причем в поле ввода уже д.б. указан протокол http://)
  - Выберите пол
  - Выберите дату рождения
  - Выбрать пункты из checkbox и radio (Например: Блюда какой кухни предпочитаете?)
  - О себе в произвольной форме
  - Кнопки «зарегистрироваться» и «очистить форму»
7. На странице «Контакты» - адрес организации, схема «Как добраться» (Например: Фотография из «Яндекс.Карты»), форма обратной связи (Имя, телефон, email, задайте вопрос).