

1. HTML

Лекция 1. Основы HTML

1.1. История языка HTML. Создание web-страниц

HTML (*HyperText Markup Language* — «язык гипертекстовой разметки») — стандартный язык разметки документов во Всемирной паутине.

Версии языка HTML

- HTML 0.9
- HTML 2.0, одобренный как стандарт 22 сентября 1995 года;
- HTML 3.2— 14 января 1997 года;
- HTML 4.0 — 18 декабря 1997 года;
- HTML 4.01 — 24 декабря 1999 года;
- HTML 5— 28 октября 2014 года
- HTML 5.1 начал разрабатываться примерно 19 декабря 2012 года.

Для создания web-страниц нам понадобится обычный текстовый редактор, например, **Блокнот** или **NotePad++**. Отличие в том, что создаваемым файлам дается расширение *html*.

Рассмотрим создание страницы, выводящей фразу «Сыктывкарский государственный университет». Для этого в блокноте набираем:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>СыктГУ</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Сыктывкарский государственный университет
```

```
</BODY>
```

```
</HTML>
```

Затем необходимо сохранить этот документ в виде файла с расширением *html*. Далее выполняем двойной щелчок по файлу, после чего запускается браузер для просмотра результата.

1.2. Структура html-документа

HTML — элемент верхнего уровня — ограничивает начало и конец документа. Элемент HTML является элементом верхнего уровня в языке HTML. Использование элемента HTML не сказывается на выводе документа. Если элемент HTML используется, начальный и конечный теги HTML должны идти вокруг *всего* документа, сразу после объявления DOCTYPE.

<HTML>

заголовок и тело документа

</HTML>

HEAD (ЗАГОЛОВОК) — заголовок документа

Основная структура HTML-документа должна состоять из заглавия и тела. Явное вложение заголовка в элемент HEAD не обязательно. Использование элемента HEAD не сказывается на вывод документа. Атрибуты отсутствуют. В документе должен присутствовать только один элемент HEAD, и он должен появиться до элемента BODY .

Синтаксис использования

<HEAD>

элемент TITLE

</HEAD>

Заметим, что начальный и конечный теги могут отсутствовать.

Тег <HEAD> должен содержать один TITLE элемент и возможно следующие элементы

- ISINDEX;
- META;
- LINK;
- STYLE;
- SCRIPT.

BODY (ТЕЛО) — тело документа. Основная структура документа HTML всегда состоит из заголовка и тела. Нет необходимости явно помещать тело в элемент BODY, однако делая так, можно специфицировать атри-

буты, влияющие на представление документа в целом (например, установить фоновое изображение или цвет).

Если элемент BODY не содержит атрибуты, использование его не дает явного эффекта в непосредственном отображении документа. Элемент HTML может быть задан явно или подразумеваться. Только один элемент BODY разрешен в документе, и он должен находиться после элемента HEAD (который тоже может быть как подразумеваемым, так и явным).

Основной синтаксис

`<BODY>тело документа</BODY>`

Возможные атрибуты тега BODY

Имя атрибута	Возможные значения	Смысл
BGCOLOR	<i>Цвет</i>	Фоновый цвет документа
TEXT	<i>Цвет</i>	Цвет для текста документа
LINK	<i>Цвет</i>	Цвет для непосещенной гипертекстовой связи
VLINK	<i>Цвет</i>	Цвет для посещенной гипертекстовой связи
ALINK	<i>Цвет</i>	Цвет для активной гипертекстовой связи; используется для выделения текста, когда пользователь нажал на связь
BACKGROUND	<i>URL</i>	URL фонового образа

1.3. Теги управления форматированием

Абзац формируется при помощи тега `<p>...</p>`. Изначально предполагалось использовать вместе с тегами параметры¹. Параметры тега `<p>` – способы выравнивания (`left`, `right`, `center`, `justify`). Например, если необходимо сделать абзац с выравниванием по ширине, то надо написать:

```
<p align="justify">Содержание абзаца </p>
```

Заголовки предназначены, во-первых, показать важность раздела, к которому относятся, а во-вторых, с помощью различных заголовков легко регулировать размер текста. Чем выше уровень заголовка, тем больше размер шрифта. Самым верхним уровнем является уровень 1 (`<h1>`), а самым нижним — уровень 6 (`<h6>`).

Также следует отметить, что использование заголовков приветствуется поисковыми системами.

Пример использования заголовков.

```
<body>
<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>
</body>
```

Для **полужирного начертания** применяется два тега: `` и ``.

```
<b>Жирное начертание шрифта</b>
<strong>Сильное выделение текста</strong>
```

Для **курсивного начертания** используются два тега: `<i>` и ``.

```
<i>Курсивное начертание шрифта</i>
<em>Выделение текста</em>
```

¹ В настоящее время параметры тегов также поддерживаются, но предпочтение отдается использованию стилей.

Следует отметить, что теги `` и ``, также как `<i>` и `` хотя и похожи по своему действию, являются не совсем эквивалентными и заменяемыми. Первый тег `` — является тегом физической разметки и устанавливает жирное начертание текста, а тег `` — тегом логической разметки и выделяет помеченный текст.

Маркированный список формируется с помощью контейнера ``, а каждый пункт списка начинается с тега ``, как показано ниже.

```
<ul>
<li>Значение №1</li>
<li>Значение №2</li>
<li>Значение №3</li>
</ul>
```

В списке непременно должен присутствовать закрывающий тег ``, иначе возникнет ошибка. Закрывающий тег `` хотя и не обязателен, но советуем всегда его добавлять, чтобы четко разделять элементы списка.

Нумерованные списки формируются так:

```
<ol>
<li>Значение №1</li>
<li>Значение №2</li>
<li>Значение №3</li>
</ol>
```

Бегущая строка задается тегом `<marquee>`.

1.4. Таблицы

Таблицы могут использоваться не только для более наглядного предоставления информации, но и для верстки веб-страниц. Таблица с невидимой границей представляет собой словно модульную сетку, в блоках которой удобно размещать элементы веб-страницы. Однако в настоящее время верстка страниц в основном осуществляется с помощью слоев.

Рассмотрим пример добавления таблицы, показанной ниже.

1	2
3	4

Для этого надо написать следующий код:

```
<table border="1">
  <tr>
    <td> 1</td>
    <td> 2</td>
  </tr>
  <tr>
    <td> 3</td>
    <td> 4</td>
  </tr>
</table>
```

Для объединения двух и более ячеек в одну используются атрибуты colspan и rowspan тега <td>.

Рассмотрим пример добавления таблицы с объединенными ячейками как показано ниже.

1	
3	4

Для этого надо написать следующий код:

```
<table border="1">
  <tr>
    <td colspan=2> 1</td>
  </tr>
  <tr>
    <td> 3</td>
    <td> 4</td>
  </tr>
</table>
```

Лекция 2. Ссылки, рисунки, фреймы

2.1. Ссылки

Ссылки являются основой гипертекстовых документов и позволяют переходить с одной веб-страницы на другую. Особенность их состоит в том, что сама ссылка может вести не только на HTML-файлы, но и на файл любого типа, причем этот файл может размещаться совсем на другом сайте

Общий синтаксис создания ссылок следующий:

```
<a href="URL">ключевая фраза</a>
```

Атрибут **href** определяет URL (Universal Resource Locator, универсальный указатель ресурса). Текст, расположенный между тегами `<a>` и ``, по умолчанию становится синего цвета и подчеркивается.

Адрес ссылки может быть абсолютным или относительным. Абсолютные адреса должны начинаться с указания протокола (обычно `http://`) и содержать имя сайта. Относительные ссылки ведут отсчет от корня сайта или текущего документа.

С тегом `<a>` можно использовать параметры `target` и `title`.

target указывает окно, в котором нужно открыть ссылку. По умолчанию, при переходе по ссылке документ открывается в текущем окне или фрейме. Синтаксис следующий:

```
<a target="имя окна">...</a>
```

В качестве значения используется имя окна или фрейма, заданное атрибутом `name`. Если установлено несуществующее имя, то будет открыто новое окно. В качестве зарезервированных имен применяются следующие.

- `_blank` — загружает страницу в новое окно браузера.
- `_self` — загружает страницу в текущее.
- `_parent` — загружает страницу во фрейм-родитель, если фреймов нет, то это значение работает как `_self`.
- `_top` — отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то это значение работает как `_self`.

Параметр **title** добавляет поясняющий текст к ссылке в виде всплывающей подсказки, которая отображается при наведении мыши на ссылку.

Пример:

```
<a href="http://syktsu.ru" title="СыктГУ">университет</a>
```

Для создания **ссылки на адрес электронной почты** в параметре href надо указать mailto:адрес электронной почты. Например,

```
<a href="mailto:admin@syktsu.ru">администратор</a>
```

Кроме внешних ссылок можно использовать внутренние ссылки. Для этого вначале необходимо сделать метку в соответствующем месте страницы и дать ей имя при помощи атрибута name тега <a> . Для создания метки используется символ решетки (#).

Пример. Создадим в конце web-страницы ссылку, которая введет в начало документа. Для этого в начале документа, задаем метку:

```
<body>
```

```
<a name="ttt"></a>
```

Затем в конце страницы задаем ссылку:

```
<a href="#ttt">В начало</a>
```

```
</body>
```

Также можно сделать ссылку на метку другого документа. Пример:

```
<a href="text.html#ttt">Как написано на сайте...</a>
```

2.2. Рисунки

Для добавления изображения используется тег , атрибут src которого определяет адрес графического файла. Общий синтаксис добавления изображения будет следующий.

```

```

Полезными являются атрибуты alt, width, height. Атрибут **alt** предназначен для создания альтернативного текста. При помощи атрибутов **width** и **height** задается размер рисунка. Следует отметить, что заданный размер не влияет на скорость загрузки.

2.3. Карты изображения

Карты изображения позволяют привязывать ссылки к разным областям одного изображения. По сути это графические изображения с размеченными областями. Для создания карты изображения нужно дать команду

```

```

Далее необходимо определить области на карте следующим образом:

```
<map name="map1">
```

```
<area shape="форма" coords="список координат" href="ссылка">
```

...

```
</map>
```

Возможны следующие варианты форм:

Форма	Название	Задаваемые координаты
rect	прямоугольник	координаты двух вершин
poly	многоугольник	координаты всех вершин
circle	окружность	координаты центра и радиус

Пример. `<area shape="poly" coords="282,2, 222,1, 203,6, 210,27, 212,33, 282,31, 301,33, 301,28" href="1.html">`

2.4. Фреймы

Окно браузера может быть разделено на фреймы, т.е. на самостоятельные области. В каждую из этих областей можно загружать свои html-страницы. Рассмотрим следующий пример (строки для удобства комментирования пронумерованы):

- 1) `<HTML>`
- 2) `<HEAD>`
- 3) `<TITLE>ФРЕЙМЫ</TITLE>`
- 4) `</HEAD>`
- 5) `<FRAMESET ROWS="45%,45%,*">`
- 6) `<FRAME SRC="B611.HTM">`
- 7) `<FRAME SRC="B612.HTM">`
- 8) `<FRAME SRC="B613.HTM">`

9) </FRAMESET>

10) </HTML>

В 5-й строке вместо тега <body> используется тег FRAMESET. Параметр ROWS указывает на то, что страница должна быть разбита на горизонтальные фреймы. Если же надо было разбить на горизонтальные фреймы, то надо указать COLS. В 5-й же строке указано, что фреймов должно быть 3 – на 1-й и 2-й отводится 45% высоты экрана, на третий – все, что осталось.

В 6-й, 7-й, 8-й строках непосредственно подключается содержимое фреймов, которое берется из файлов B611.HTM, B612.HTM, B613.HTM.

В настоящее время фреймы практически не используются, так как более удобным является использование слоев. Тем не менее активно используются плавающие фреймы, для задания которых используется тэг <IFRAME>. Плавающие фреймы поддерживаются не всеми браузерами. Плавающий фрейм можно расположить в любом месте страницы, для этого надо указать код

```
<iframe src="111.html" name="iframe1" width="300" height="300"
align="left"></iframe>
```

Здесь будет подключен файл 111.html.

2. CSS

Лекция 3. Каскадные таблицы стилей

3.1. Синтаксис CSS

Определение. Таблицы стилей (или каскадные таблицы стилей, CSS) – это описание правил, задающих параметры представления отдельных элементов на языке HTML

CSS появились одновременно с HTML 4.0 (Dynamic HTML). Сам термин «каскадные таблицы стилей» был предложен в 1994 году. Все объявления CSS называются *селекторами*, записываются в фигурных скобках.

Примеры использования синтаксиса.