

Лекция «Технология создания Web – страниц. Язык HTML».

План

- Основные понятия, связанные с определением языка HTML
 - Назначение языка HTML
 - Теги HTML
 - Атрибуты тегов

Литература:

1. Информатика: учебник / В.П. Омельченко, А.А. Демидова. – М.: ГЭОТАР-Медиа, 2013. – 384с.: ил.
2. Цветкова М.С. Информатика и ИКТ : учебник для нач. и сред проф. образования / М.С. Цветкова, Л.С. Великович. – М.: Издательский центр «Академия», 2011. – 352 с., (8) л. цв. ил.

ДЗ .

- Конспект стр. 358 в.10-17 (В.П. Омельченко «Информатика» стр. 291-306)
- Таблица тегов
- Таблица кодировки цвета

Основные понятия, связанные с определением языка HTML

Все Web-страницы Интернета имеют одну общую черту — они созданы с помощью средств языка HTML.

HTML - не обычный язык программирования, хотя процесс создания Web-страницы очень близок к процессу программирования.

HTML — это язык разметки гипертекста. Он определяет правила, согласно которым обычный текст представляется в виде Web-страниц.

Язык HTML появился одновременно со службой World Wide Web и развивался вместе с ней, постепенно вбирая в себя новые черты, которые позволяли создавать все более и более впечатляющие Web-страницы.

Он является основой World Wide Web и одновременно причиной ее широчайшей популярности. Смысл и назначение языка HTML можно понять, исходя из его названия.

Как было сказано выше *Гипертекст* — это текст, в который встроены специальные коды, управляющие дополнительными элементами, такими как форматирование, иллюстрации, мультимедийные вставки и гиперссылки на другие документы.

Под *разметкой* понимается вставка в текст этих кодов, определяющих то, как итоговый гипертекстовый документ должен отображаться специальной *программой-браузером*.

Разметка, может быть, простой или сложной, но в любом случае исходный текст сохраняется в документе в неизменном виде.

Но самое важное слово в этом описании — **язык**.

HTML представляет собой компьютерный язык, в некотором смысле родственный языкам программирования. Он включает достаточно строгие правила, которые необходимо соблюдать, чтобы получить правильные результаты.

Назначение языка HTML

Несмотря на то, что Web-страницы появляются на экране компьютера в отформатированном виде, язык HTML не предназначен для форматирования документов, поскольку жесткое задание оформления и точное позиционирование элементов текста на странице приводит к ограничениям, недопустимым в Интернете.

Так, например, когда мы форматируем текст с помощью текстового процессора Word, мы *однозначно* определяем, как должен выглядеть этот текст при печати

на совершенно определенном принтере и на бумаге заданного формата. Когда документ размещен в Интернете, невозможно предсказать, какой компьютер будет использован для его просмотра, да и будет ли вообще у этого компьютера монитор.

Может быть, текст появится на экране компьютера, работающего в системе Windows в окне одного из современных браузеров. Может быть, это будет текстовый браузер (неспособный отображать графику), работающий в системе MS-DOS. Возможно, текст документа вообще не будет отображаться на экране, а будет воспроизводиться вслух с помощью синтезатора речи. Для слепого пользователя документ может выводиться на специальное устройство шрифтом Брайля.

В связи с необходимостью подготавливать документы для столь разнообразных устройств язык HTML не предназначен для описания формата документа. Он служит для функциональной разметки документа, то есть позволяет определить назначение фрагментов текста.

Например, если в тексте встречается заголовок, то код HTML просто указывает, что соответствующий фрагмент является заголовком. Получив такой код, программа просмотра (браузер) сама «решает», что ей делать с заголовком. Возможно, она отобразит его более крупным шрифтом, а может быть выровняет по центру экрана. Возможно, что синтезатор речи, воспроизводящий текст документа, использует код заголовка для того, чтобы повысить громкость и сделать необходимую интонационную паузу.

Язык HTML все-таки имеет некоторые команды форматирования, но общий курс на разделение содержания и оформления документа выдерживается четко. В последней версии HTML- 4.0 использовать команды форматирования, как правило, не рекомендуются.

Теги HTML

Коды языка HTML, с помощью которых выполняется разметка исходного текста, называются *тегами*.

Тэг — это набор символов. Все теги начинаются с символа «меньше» (<) и заканчиваются символом «больше» (>). Пару этих символов иногда называют *угловыми скобками*. После открывающей угловой скобки идет *ключевое слово*, определяющее тэг.

Каждый тэг в языке HTML имеет специальное назначение. Регистр букв в названиях тэгов не имеет значения — можно использовать как строчные, так и прописные буквы, хотя общепринято использовать прописные буквы, чтобы теги отличались от обычного текста документа.

Как правило, один тег HTML воздействует только на часть документа, например на абзац. В таких случаях используют парные теги: открывающий и закрывающий. Открывающий тэг создает эффект, а закрывающий — прекращает его действие. Закрывающие теги начинаются с символа косой черты (/).

Некоторые теги дают разовый эффект в месте своего появления. В этом случае необходимости в закрывающем теге нет, и он не употребляется.

Если по ошибке в теге указано ключевое слово, отсутствующее в языке HTML, то тэг игнорируется целиком.

При отображении документа в браузере сами теги не отображаются, но влияют на способ отображения документа.

Теги

Теги описывают браузеру структуру веб-страницы. Большинство из них имеют открывающий и закрывающий тег, но далеко не все.

Например, <title>...</title>, где вместо точек находится содержимое. Первый из них показывает, где начинается тег, а второй закрывает его. Внутри могут находиться другие элементы разметки страницы, они могут вкладываться друг в друга подобно матрёшке. Тут важно своевременно закрывать теги, чтобы страница отображалась корректно.

Встречаются также одиночные теги, которые не нужно закрывать. В них содержимое находится внутри, так же как и атрибуты.

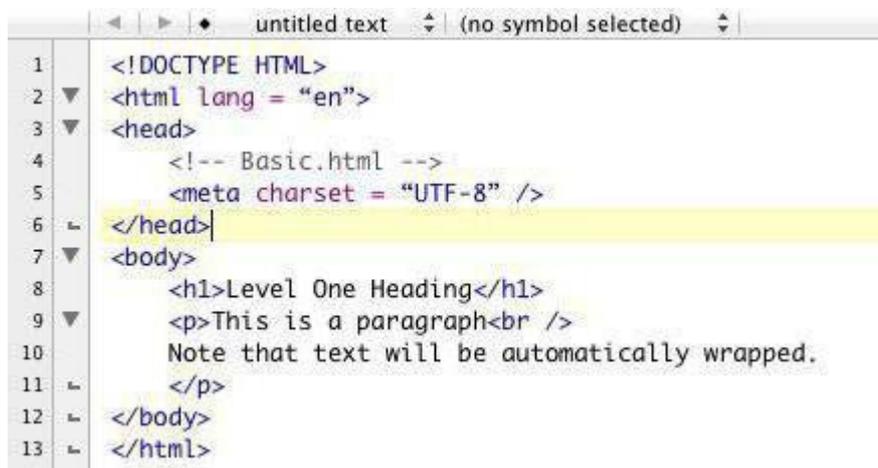
Атрибут может быть прописан для большинства HTML-тегов и задаёт свойства элемента. Обозначается он в открывающем теге и выглядит примерно так: атрибут=«...», где вместо точек находится значение атрибута. Знание тегов является первым и самым важным шагом для освоения HTML. Основы этого искусства также подразумевают понимание структуры веб-страницы.

Структура документа

Каждый документ HTML имеет соответствующее расширение, например, Index.html. Так браузер сможет понять, с чем он имеет дело, и правильно отобразит страницу. Все файлы, используемые для создания сайта, желательно хранить в одной директории, что в дальнейшем существенно облегчит вам жизнь.

Основы языка разметки гипертекста HTML предполагают чёткое понимание структуры документа. Начинается он с тега `<!doctype html>`, сообщаящего браузеру версию HTML, которая используется в этом документе.

На данный момент актуальна пятая версия языка, так что тут не нужно ничего выдумывать, можно смело вставлять приведённый выше тег в начало любой страницы.



```
untitled text (no symbol selected)
1 <!DOCTYPE HTML>
2 <html lang = "en">
3 <head>
4     <!-- Basic.html -->
5     <meta charset = "UTF-8" />
6 </head>
7 <body>
8     <h1>Level One Heading</h1>
9     <p>This is a paragraph<br />
10    Note that text will be automatically wrapped.
11 </p>
12 </body>
13 </html>
```

Потом идут основные парные структуры, составляющие «скелет» сайта. Первый тег, в который вложены все остальные - `<html>...</html>`.

Всё, что находится за его пределами, не распознаётся браузером как веб-страница, так что он открывает документ и закрывает его. Этот тег является обязательным для любого документа. В нём также содержатся ещё несколько обязательных тегов, которые будут рассмотрены ниже.

Head

Внутри тега `<head>...</head>` содержится информация технического характера, которая не будет отображаться на странице, но, тем не менее, является важной частью документа HTML.

Основы сайта закладываются именно в этом месте, здесь выбирается кодировка и вводится название страницы. Оно содержится внутри обязательного тега `<title>...</title>`. Отображается название в верхней части браузера, где также можно разместить небольшую иконку, характеризующую содержимое страницы.

Желательно сразу указать кодировку документа для его корректного отображения. Это можно сделать при помощи тега `<meta charset="UTF-8" />`. Метатеги предоставляют данные о структуре страницы и обычно располагаются внутри head.

Link

Знание основ HTML также предполагает использование каскадных стилей оформления, или CSS. В них задаются свойства элементов, которые будут отображены на странице. Современный подход к этой задаче предполагает вынесение таких

характеристик, как цвет, высота и местоположение элемента, во внешний файл для большего удобства. Для подключения css-файла используется тег `<link/>`.

В готовом виде это выглядит примерно вот так: `<link href="style.css" type="text/css" />`, где **href** указывает на местоположение файла, а **type** - на его тип.

Body

Именно в этой части HTML-документа создаётся видимая часть страницы. Всё, что делается внутри «тела», будет показано браузером. В `<body>` используется огромное количество тегов HTML.

Основы — это форматирование текста, работа со ссылками и простейшие инструменты для структурирования веб-страницы.

Чтобы приступить к работе в HTML, достаточно знать основные теги и уметь ими пользоваться. Ниже приведены самые популярные из них:

`` - применяется для выделения подстроки, на которую будет распространяться особый стиль, описанный в css;

`<a>` - создаёт ссылку на веб-странице; адрес перехода задаётся атрибутом href;

`<div>` - один из популярнейших тегов современности; каждый, кто решил изучить основы языка HTML, должен уделить ему особое внимание, поскольку это блочный элемент, на основе которого сделана львиная доля современных сайтов (параметры для блоков `<div>` задаются в css, и внутри этого тега могут располагаться другие блоки);

`<p>` - выделяет абзац из текста; содержимое абзаца находится между открывающим и закрывающим тегами;

`` - нумерованный список, элементы которого заключаются в парный тег ``;

`` - маркированный список, в котором, так же как и в нумерованном, элементы обозначаются тегом ``

`<h1>`-`<h6>` - заголовки документа (цифра обозначает уровень заголовка, то есть `<h1>` - главный заголовок, а последующие варианты являются его подзаголовками, кстати, заголовки уровней `<h5>`, `<h6>` почти невозможно встретить на просторах интернета), также важно помнить, что на странице может быть лишь один заголовок `<h1>`

`` - жирный текст;

`` - курсив;

`` - вставка картинки на интернет страницу (это единичный тег, ему не требуется закрывающий, но в нём обязательно нужно указать атрибут **alt**, в котором указывается текст для изображения);

`<video>` - вставка видео на веб-страницу;

`<audio>` - тег, вставляющий в документ аудио файл.

Это далеко не все теги, необходимые для создания собственной веб-страницы, но их вполне достаточно, чтобы заложить основы HTML для начинающих.

CSS

Развитие языка HTML привело к тому, что каждый тег обзавёлся массой атрибутов, да и требования к внешнему виду веб-страниц существенно возросли. Код стал громоздким и неудобным, сложно было прочитать его, не то что адаптировать или изменить.

К тому же, если на вашем сайте десять страниц, на которых много заголовков, отмеченных зелёным цветом, а вам вдруг захотелось сделать их красными, то придётся попотеть, изменяя каждый вручную. С появлением каскадных таблиц стилей этот процесс стал простым и логичным, да и HTML-код стал куда более читаемым.

Атрибуты тегов

Открывающие теги часто могут содержать *атрибуты*, влияющие на эффект, создаваемый тэгом. Атрибуты — это дополнительные ключевые слова, отделенные от ключевого слова тега и друг от друга пробелами.

Некоторые атрибуты требуют указания *значения атрибута*. Это значение отделяется от ключевого слова знаком равенства (=). Значение атрибута должно заключаться в кавычки, но во многих случаях эти кавычки могут опускаться без какого-либо вреда. Закрывающие тэги никогда не содержат атрибутов.

Примеры использования тегов HTML:

```
<HEAD> Общий заголовок  
<TITLE> Заголовок окна </TITLE >  
</HEAD>  
<BODY>  
<H1>Крупный текст</H1 >  
<H2>Средний текст</H2>
```

<P> Эти строки изображаются слитно, несмотря на то, что в документе они отделены друг от друга

```
<P> Закрывающий тэг абзаца не обязателен.
```

```
<P>Тэг начала абзаца более важен, чем реальный переход на новую строку. <HR>  
Текст после горизонтальной линейки <BR> разбит на две строки.
```

```
</BODY>  
</HTML>
```

Примеры парных тэгов HTML:

```
<HTML> ... </HTML>  
<HEAD>... </HEAD>  
<H3> ... </H3>
```

Примеры одиночных тэгов HTML:

```
<BR> ... <HR>... <BASEFONT> ... <FRAME> ... <INPUT>...<P>
```

Примеры тэгов HTML с атрибутами:

```
<BODY BGCOLOR=»YELLOW» TEXT=»BLUE» > ...  
<HR COLOR=RED SIZE=16 WIDTH=100%> ...
```

Перечень основных понятий: Язык HTML. Теги HTML.

Применение CSS

Для создания веб-страниц необходимо знать основы HTML и CSS, так как сейчас без знания каскадных таблиц стилей делать в этой сфере нечего. В них задаются атрибуты для любого элемента, распространяющиеся на весь документ. Таким образом, можно задать цвет сразу для всех элементов `<p>` или `<a>`, написав лишь одно правило. Учитывая современные требования к внешнему виду веб-страниц, использование каскадных таблиц стилей является обязательным.

Для того чтобы подключить файл `css` к документу, существует тег `link`. Чуть выше был описан принцип его использования, но он не является единственным вариантом объединения всех стилей в одном месте. Есть ещё тег `<style>`, который располагается в «голове» документа и позволяет прописать стили, не используя файлы CSS. Не обязательно пользоваться либо одним методом, либо другим. Их можно успешно комбинировать для получения наилучшего результата. Для создания файла с таблицами стилей нужно создать файл с расширением `.css`, например, `Styles.css`.

JavaScript

Нередко человек, решивший заняться разработкой веб-страниц, понимает, что для его задач недостаточно тех средств, которые предлагает HTML. Основы позволяют создать красивую страницу, но что делать, если вдруг понадобилось сделать её интерактивной? Для этих целей существует уникальный сценарный язык программирования, который прекрасно взаимодействует с HTML. Называется он JavaScript, так как задумывался как младший брат популярного языка Java. Сегодня эти языки обзавелись существенными различиями, и пропасть между ними только растёт.

JavaScript может расширить возможности HTML, он позволяет создавать и редактировать теги.

Также при помощи этого замечательного инструмента можно работать с Cookie, загружать данные с сервера, не перезагружая страницу, сделать сайт более интерактивным, чем позволяют возможности HTML.

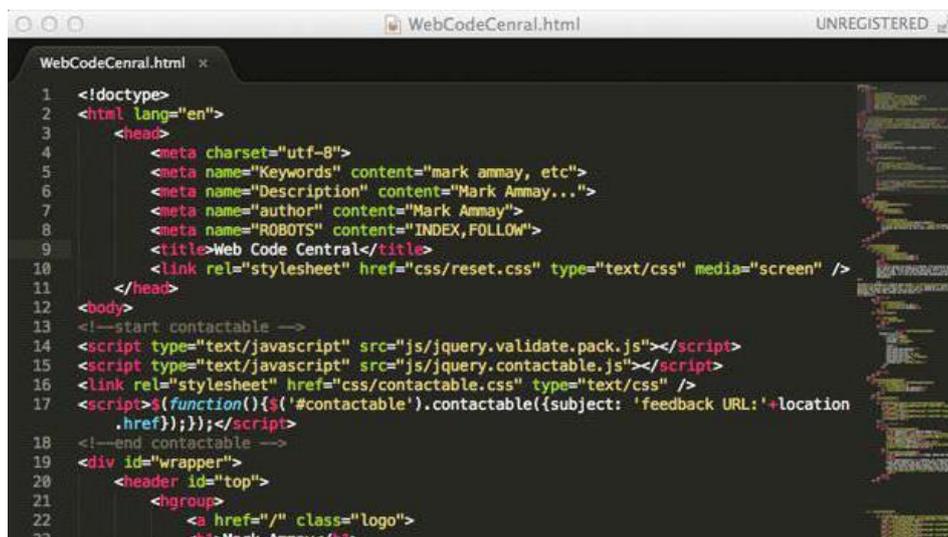
Есть у этого языка и ограничения, которые связаны с безопасностью. Если JavaScript используется не на серверной части, то он будет выполняться в условиях, ограничивающих его возможности, дабы злоумышленники не могли задействовать вредоносный код на любом компьютере.

Редакторы

Основы HTML для новичка предполагают знания о наиболее удобных и практичных программах для создания веб-страниц. Как было написано выше, визуальные редакторы, такие как Dreamweaver и ему подобные, не подходят для этих целей.

Так что же, писать теги в обычном блокноте? Этот вариант также сомнителен, так как стандартный блокнот не располагает никакими специальными инструментами для вёрстки.

С этой задачей может неплохо справиться Notepad++. Большой плюс этого продукта в том, что он имеет открытый код и распространяется абсолютно бесплатно. В нём присутствует удобная подсветка синтаксиса и автоматическое закрывание тегов. Notepad++ также предоставляет широкий выбор языков интерфейса, а его возможности легко расширяются при помощи многочисленных дополнений.

A screenshot of a code editor window titled 'WebCodeCentral.html'. The editor shows HTML code with line numbers from 1 to 23. The code includes a DOCTYPE declaration, a lang attribute, a head section with meta tags for charset, keywords, description, author, and robots, and a title 'Web Code Central'. It also includes a link to a stylesheet and a body section with comments and script tags for jQuery validation and contactable.js. The code ends with a header and a logo link.

```
1 <!doctype>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="Keywords" content="mark ammay, etc">
6     <meta name="Description" content="Mark Ammay...">
7     <meta name="author" content="Mark Ammay">
8     <meta name="ROBOTS" content="INDEX,FOLLOW">
9     <title>Web Code Central</title>
10    <link rel="stylesheet" href="css/reset.css" type="text/css" media="screen" />
11  </head>
12  <body>
13    <!--start contactable -->
14    <script type="text/javascript" src="js/jquery.validate.pack.js"></script>
15    <script type="text/javascript" src="js/jquery.contactable.js"></script>
16    <link rel="stylesheet" href="css/contactable.css" type="text/css" />
17    <script>$(function(){$('#contactable').contactable({subject: 'feedback URL: '+location
18      .href});});</script>
19    <!--end contactable -->
20    <div id="wrapper">
21      <header id="top">
22        <hgroup>
23          <a href="/" class="logo">
24            <img alt="Mark Ammay's logo" />
25          </a>
26        </hgroup>
27      </header>
28    </div>
29  </body>
30 </html>
```

Sublime Text 3 — программа, похожая на Notepad++, но распространяющаяся на платной основе. Именно она покорила сердца большинства разработчиков. Sublime Text 3 идеально подходит для JavaScript, CSS и HTML. Основы работы с ней придётся изучать самостоятельно, но оно того стоит. В ней заложены воистину безграничные возможности для тонкой настройки, что позволяет максимально подогнать программу под свои нужды.

Основы HTML и CSS для начинающих

Как видите, изучить искусство создания веб-страниц не так уж и сложно, как кажется на первый взгляд. Всего несколько месяцев практических занятий превратят вас из робкого пользователя в начинающего разработчика. Научиться вёрстке гораздо проще, чем освоить какой-нибудь язык программирования или Linux. На самом деле тегов HTML не так уж и много, важно понять практический аспект их применения.

Не лишним в этом деле будет навык работы в Adobe Photoshop. Эта программа позволяет работать с фотографиями, картинками и прочими графическими элементами веб-страницы. На данный момент именно "Фотошоп" лучше всего справляется с такими задачами, конкурентов у него немного. Для тех, кому этот продукт компании Adobe пришёлся не по нраву, есть Lightroom, GIMP, Illustrator и другие программы с похожими функциями.

Что даёт знание HTML

Навыки создания веб-страниц очень актуальны сегодня, так как интернет расширяется не по дням, а по часам. Каждое предприятие, любой, даже самый маленький магазинчик, мастерские и спортивные клубы — все хотят иметь собственный сайт. И конечно же, для этого им понадобится разработчик, владеющий CSS и HTML. Основы просто освоить, после чего дело за практикой. Так как технологии вёрстки постоянно развиваются, то фронтенд-разработчики всегда будут востребованы. Тот, кто решит посвятить себя этой интересной отрасли, никогда не останется без работы.

HTML — язык разметки гипертекста, который сделал интернет таким, каким мы его знаем и любим. Именно благодаря этому замечательному инструменту сайты выглядят красиво и современно, а также обеспечивается удобство их использования.

Язык HTML просто компоует элементы веб-страницы в удобный для пользователя вариант. Его работа сравнима с тем, что делают текстовые редакторы типа MS Word или OpenOffice. Они превращают безликую массу букв в документ, в котором есть абзацы, жирный текст, курсив, таблицы и даже изображения. Примерно то же самое делает язык HTML, с той лишь разницей, что его документы отображаются в браузере, да и возможности этого инструмента гораздо шире, чем у текстового редактора.

Для разметки используются **теги** — специальные команды, описывающие структуру веб-страницы. Они заключены в угловые скобки - `<тег>`, чтобы браузер мог отличить их от общей массы текста.

Визуальные редакторы

Новички, только ступившие на путь изучения HTML, часто начинают свою работу с программ, позволяющих создавать сайты без каких-либо знаний. В них можно просто расположить элементы на экране так, как они будут отображаться в браузере. Казалось бы, вот он — источник вечной благодати, позволяющий избавиться от большинства веб-разработчиков.

Но не всё так просто, так как у визуальных редакторов есть масса недостатков, которые делают невозможным их использование в серьёзных проектах. Все эти программы создают много ненужных тегов, которые делают конечный вариант страницы громоздким и неоптимальным.

Конечно, в наш век скоростного интернета это имеет меньшее значение, чем раньше, но есть ряд причин, по которым лаконичный и грамотно написанный сайт получается более практичным, чем его собрат, созданный в визуальном редакторе.

Веб-страница, сделанная в такой программе, будет плохо обрабатываться поисковыми роботами, так как для них важен каждый килобайт кода, а громоздкий и нелогичный код с кучей семантических ошибок едва ли придётся им по вкусу.

Помимо этого, редакторы часто отстают от времени, становясь неактуальными, а тратить ресурсы на их развитие нецелесообразно, так как ни один профессионал не пользуется этими продуктами. Поэтому каждый, кто хочет работать в сфере создания сайтов, должен знать основы HTML.

Вопросы к самопроверке:

- 1) Что такое HTML?
- 2) Что такое гипертекст?
- 3) Что понимается под разметкой?
- 4) Что представляет собой HTML?
- 5) Для чего не предназначен HTML?
- 6) Как называют коды языка HTML, с помощью которого выполняется разметка исходного текста?
- 7) Что такое Тэг?
- 8) В каких случаях используют парные тэги?
- 9) Что такое атрибуты?
- 10) Что требуют некоторые атрибуты?
- 11) Теги структуры документа
- 12) Теги для форматирования текста
- 13) Теги вставки объектов:
 - a. Рисунки
 - b. Таблицы
 - c. ссылки