

HTML Crash Course

By Daniel D'Agostino
2nd June 2007

Revised 19th June 2007

HTML Kickstart: Tools Needed

A text editor

- * Notepad is fine
- * Chami's HTML-Kit is better
- * Other systems can use their respective text editors

A web browser, preferably two

- * Firefox
- * Internet Explorer

Some good reference

- * <http://www.w3schools.com>

HTML Kickstart: HTML Demo

- * HTML is just text
- * HTML **tags** give text a special meaning
- * Create a new text file
- * Name it *test.htm* (*htm = html*)
- * *Write the following in the file:*

“The future <u>is</u> <i>now</i>.”

- * Now open the file with a web browser.
- * What happened to the text enclosed in tags?

HTML Kickstart: HTML Syntax

- * HTML tags normally have a **start tag** () and a **closing tag** ()
- * You can have multiple tags around the same text but they must be closed in reverse order

```
<b><i>Hello!</i></b>
```

- * Tags are sometimes called **elements**

HTML Text Formatting: Spacing

White space

- * newlines
- * multiple spaces

Spacing tags

- * `
`
- * `<p>`
- * `
` is called an **empty tag**
- * Block and inline elements

HTML Text Formatting:

- * The tag is **deprecated**
- * Use it until you learn better things
- * HTML tags can have attributes
- * attributes:
 - * face
 - * color
 - * size
- * Order of attributes not important
- * Values in quotes

HTML Text Formatting: Headings

- * <h1> through <h6>
- * Hierarchical structure (use <h1> for page title, etc)
- * Headings may be customised (like all other text)

```
<h2><font color="blue"><u>Heading 1</u></font></h2>
```

```
<h2><font color="blue"><u>Heading 2</u></font></h2>
```

```
<h2><font color="blue"><u>Heading 3</u></font></h2>
```

Why Layout Tags are Bad

- * Previous example has lots of extra layout code
- * A lot of layout code is repeated several times
- * It adds nothing to the meaning of the text (the heading is still a heading)
- * Ideally content and presentation should be separate
- * This is achieved with CSS (another language)... for now live with HTML

Logical Tags vs Layout Tags

- * `` is very bad
- * ``, `<i>` and `<u>` are bad as well
- * `` and `` should be used instead of `` and `<i>`
- * `<u>` should not be used because text can be confused with links
- * `` means text should stand out
- * `` means text should be emphasised
- * `` and `<i>` just mean the text should look different (no meaning)
- * example of JAWS browser reading text
- * `not` or `not`?

HTML Text Formatting: Quotations

The `<blockquote>` tag

- * needs inner tags (e.g. `<p>`)
- * block element

The `<q>` tag

- * inline element
- * browser incompatibilities

HTML Comments

`<!-- comment -->`

Useless in HTML

- * demonstrate using `<blockquote>`
- * HTML tags are clear and legible
- * HTML comments increase the size of the page, unlike with programming

HTML Text Formatting: Code

Tags used to format source code in a page

- * `<code>` (inline, monospace)
- * `<var>` (inline, not monospace)
- * `<pre>` (block, monospace, kills whitespace)

HTML Text Formatting: Other

`<p>` can be aligned left, right, center or justify (left is usually best)

Serif fonts are best for printed media; sans-serif are best for reading off a screen

Horizontal rules (`<hr>`): another empty tag

More (less useful) text formatting tags:

http://www.w3schools.com/html/html_formatting.asp

HTML Entities

- * You cannot use < or > characters because they are used for HTML tags
- * Entities are used where special characters are needed
- * Examples: < > © & "
- * Entity formats: name and number
- * Reference:
http://www.w3schools.com/html/html_entitiesref.asp

HTML Links

- * Link is the most important thing
- * Format: `link`
- * Links are relative unless they start with a protocol (e.g. `http://` or `file://`)
- * `google`
 - wrong! (relative)
- * For internal links, relative links are better
 - * shorter
 - * transparent when relocating website

Relative Links: Folder Navigation

- * / - root
- * .. - parent directory
- * . - current directory
- * dir/ - child directory named 'dir'
- * ../dir1/ - 'dir1' directory in parent directory

Append slash to domains/folders to prevent multiple requests

HTML Links: Local Links

- * Used to link to a specific point in a page
- * `local link`
- * `local anchor` (old)
- * `<h2 id="label">local anchor</h2>` (best)
- * Example of simple table of contents

Links to other media

mailto: links

- * try mailto: in browser
- * mailto: link
- * additional parameters
- * make it VERY clear that mail client will pop up

Other file types (zip, pdf, doc, etc)

- * Linking to any file type is possible, not just htm
- * make it clear that link does not go to a webpage
 - * it is very annoying to have a pdf load when a webpage is expected
 - * put a little icon or something

Notes about links

When you link to a folder, it will fetch index.*, so index.htm is generally the first file you should create

Link text should be descriptive (e.g. “Photos of my garden”) to show where they go

Link text should *not* be an action (e.g. “Click here!”)

- * link destination is not clear

- * people don't like being told what to do

Use *title* attribute when a link cannot be clear

HTML Lists

- * Ordered Lists (,)
- * Unordered Lists (,)
- * When to use each
- * Definition Lists (<dl>, <dt>, <dd>)
- * Nested Lists

HTML Images

- * Images are separate files (compare doc with htm)
- * ``
- * ALT is not there to popup text, even though Internet Explorer does it
- * ALT = alternate text, used if the image fails to load
- * Use *title* attribute to popup text

Other attributes: width, height, border, title, align, hspace, vspace

HTML Images: Advice

Use only jpg, gif and png formats

- * bitmaps are too large
- * other formats are not always readable

Specify a width and a height for the image to make sure page doesn't keep resizing every time an image loads

Thumbnailing

- * To make a thumbnail, enclose an image in a link

```
<a href="largepic.jpg">  
    
</a>
```

- * It is possible to use the same picture and resize it using width and height attributes
- * Very bad – wastes bandwidth!

Image Maps

This section is here for completeness' sake and can be skipped

- * image map => image with clickable areas
- * `<map>` tag, id attribute
- * `<area>` tag, attributes: alt, coords, href, shape, title
- * shape can be rect, circle or polygon

HTML Tables

- * A table is made up of rows and columns
- * A table can be considered a list of rows
- * A row can be considered a list of cells

- * `<table>`: table
- * `<tr>`: table row
- * `<td>`: table data (cell)
- * `<th>`: table heading

HTML Tables: Example

```
<table>  
  <tr>  
    <td>Row 1 Cell 1</td>  
    <td>Row 1 Cell 2</td>  
  </tr>  
  <tr>  
    <td>Row 2 Cell 1</td>  
    <td>Row 2 Cell 2</td>  
  </tr>  
</table>
```

HTML Tables: Attributes

- * cellpadding and cellspacing (difference between padding and margin)
- * colspan and rowspan (merging cells)
- * summary
- * usual width, height, border, background, bgcolor

Basic Page Layout with Tables

Page Title	
Navigation	Content

Why Tables are Not So Good

- * They still bloat the page with presentation information
- * They defeat the purpose of having a logical relationship between rows and columns

Item	Price
A	\$4.99
B	\$3.50
C	\$8.99
D	\$1.00

- * They are still better than frames (next)

HTML Frames

- * Why frames are/were used
 - * Navigation in one file
- * Why frames are bad
 - * No logical page structure
 - * Printing problems (old)
 - * Browser compatibility (old)
 - * Search engine difficulties
 - * If a search engine does find a page, that page is isolated
 - * Address bar does not tell you where you are

Frames (continued)

- * Why frames are bad (continued)
 - * Cannot view source
 - * Bookmarking/deep-linking
- * Why there is no excuse to use frames
 - * Now there are languages (e.g. PHP) for server-side page inclusion
- * Targeting links
 - * target attribute: `_top`, `_blank`
 - * not recommended!

HTML Forms

- * Forms are a way of interacting with a website (e.g. an application form)
- * Forms are useless until you know a language capable of processing them (e.g. PHP)
- * The `<form>` tag:
 - * method (difference between get/post)
 - * name/id
 - * action

HTML Forms: <input>

<input> tag used for most inputs

- * type: button, checkbox, file, hidden, image, password, radio, reset, submit, text
- * importance of submit button
- * id attribute (to distinguish input fields)
- * value attribute (to set a default value)

HTML Forms: Other input tags

```
<textarea rows="40" cols="10" id="message">  
hi  
</textarea>
```

```
<select name="day">  
  <option value="1">Monday</option>  
  <option value="2">Tuesday</option>  
  <option value="3">Wednesday</option>  
  <option value="4">Thursday</option>  
  <option value="5">Friday</option>  
  <option value="6">Saturday</option>  
  <option value="7">Sunday</option>  
</select>
```

<fieldset>

Creates a border around some elements, good for distinguishing a form or parts of a form

```
<fieldset>  
  <legend>Caption!</legend>  
  <p>Other stuff...</p>  
</fieldset>
```

Meaning of HTML

Now that you know HTML, you can understand what it stands for.

HTML = Hypertext Markup Language

Hypertext is about links.

A markup language differs from a programming language in that it is text-based and uses tags to 'mark up' text.

History of HTML:

- * Once upon a time there was HTML 3.2
- * HTML 3.2 introduced the tag
- * This allowed webmasters to handle both presentation and content using HTML
- * It made a mess
- * HTML 4 was later introduced, along with CSS. HTML handles page structure, while CSS handles layout.
- * The latest standard in pure HTML is HTML 4.01
- * The latest HTML standard is XHTML 1.1
- * HTML 5 and XHTML 2 are being developed

History of HTML: The Browser Wars

- * Once upon a time there were no standards for HTML
- * Microsoft Internet Explorer and Netscape Navigator were the top browsers at the time, so they decided which HTML tags to create
- * They made a mess
- * Each one started inventing its own tags, e.g. Microsoft invented `<marquee>` and Netscape invented `<blink>`, both of which are very annoying

History of HTML: W3C

- * A body was needed to set standards
- * The World Wide Web Consortium (W3C) was created
- * The W3C is directed by the inventor of the World Wide Web, Tim Berners-Lee
- * Tim Berners-Lee was knighted in July 2004
- * The W3C sets standards of HTML and many other web technologies
- * The W3C also maintains an HTML validator:
<http://validator.w3.org/>
- * Your HTML will not validate as it is!

HTML Document Structure

- * You can't just throw your tags around the page blindly
- * There is a document structure that must be adhered to
- * Each HTML document consists of a head and a body
- * All your content goes in the body
- * The head section contains information about the page

HTML Document Structure Example

```
<html>  
<head>  
  <title>My page</title>  
  <meta http-equiv="Content-Type"  
    content="text/html; charset=iso-8859-1">  
</head>  
<body>  
  
<!-- page content -->  
  
</body>  
</html>
```

Head Section Explained

- * `<title>` is the browser title
- * `<meta>` tags are metadata, i.e. data about data, i.e. information about the page
- * `<meta>` tags can include keywords, author, description, etc.
- * They are mostly useless but the one in the example (which specifies the character encoding of the page) is necessary for validation
- * Don't try to remember the character encoding meta tag by heart... just copy and paste!

Document Type Definition

Before the `<html>` tag, we must place a Document Type Definition showing the HTML version and one of three document types:

- * Transitional – the validator is very lenient
- * Strict – recommended for good code
- * Frameset – used for frames (keep away)

The DTD is another of those things you copy and paste and don't remember by heart

DTDs for HTML 4.01

Taken from:

http://www.w3schools.com/html/html_whyusehtml4.asp

```
<!DOCTYPE HTML PUBLIC  
"-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<!DOCTYPE HTML PUBLIC  
"-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE HTML PUBLIC  
"-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

Validate your code

- * Put one of those DTDs (preferably the strict one) at the beginning of your HTML page, before the `<html>` tag
- * Try validating your page against the W3C Validator now
- * The Validator will complain if any HTML is not well-formed
- * Don't worry if you see loads of errors... errors tend to cascade so chances are that fixing one line will solve about 20 errors

XHTML

XHTML is Extensible HTML

- * it is based on XML so it is stricter
- * it is among the latest HTML standards
- * some handheld devices (e.g. mobile phones) can read XHTML webpages
- * it is a good way to get used to writing good code

HTML to XHTML

- * All tags must be closed, even if they are empty tags
 - * `
` becomes `
`
- * All tags and attributes must be lowercase
 - * `` not ``
- * All tags must be closed in the right order
 - * `<i>wow</i>` is wrong
- * Attribute values must be in quotes
 - * `<p align="center">text</p>`
- * Use the *id* attribute instead of *name*
- * Use an XHTML DTD

XHTML 1.0 DTDs

Taken from:

http://www.w3schools.com/xhtml/xhtml_dtd.asp

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

XHTML 1.1

- * XHTML 1.1 is a bit trickier
- * It is not just about adding a DTD
- * An XML version declaration is added at the top before the DTD
- * This declaration also includes the character encoding, so we no longer need the meta tag
- * The `<html>` tag now also has some extra attributes
- * Don't remember! Copy and paste!

XHTML 1.1 Strict Example

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">  
<head>  
...
```

Remember to omit the `<meta>` tag for character encoding!

Moving On

- * HTML on its own is very basic
- * The next language to learn is definitely CSS
- * After CSS, you can optionally learn JavaScript
- * Or you can go straight to a server-side language such as PHP