

# MCM 545 Web Design

with your host Joshua Logsdon

## HTML Structure Primer

### Syntax

The rules governing the structure of statements. Below is HTML syntax.

### Elements

This term can be used in different contexts in order to group objects together. In a technical context, several elements, such as the heading and body elements, comprise a web page. Inside these elements, other elements can exist and so on. In design, elements can refer to portions of the visual layout of your web page.

### Tags

Technical elements are comprised of HTML "tags." Tags are the code that the browser interprets in order to display formatting and processing information. They are obvious when looking at code because they start with a less-than sign, <, and end with a greater-than sign, >. The tag name comes directly after the starting less-than sign, i.e. <tag\_name>. Tags that manipulate text or other code come in pairs, an opening tag and a closing tag, with the text or other code in between, i.e. <tag\_name>filler text</tag\_name>. The closing tag is important so that it is known when manipulation should stop. For tags that do not require a closing tag, it is considered good form to put a space and closing slash before the last greater-than sign, i.e. <tag\_name />.

### Attributes

Tags can have additional information, attributes, that allows for more control over their functionality. Attributes and their values inside quotes are placed immediately inside of the opening tag, i.e. <tag\_name attribute="value" />.

# MCM 545

# Web Design

with your host Joshua Logsdon

## HTML Tags — <a>

### Don't Can't Leave Home Without It

This lovely tag is the basis for web navigation; it's what turns text into links (aka hyperlinks). There are two main attributes of concern. The first is **href** and is used to define what the link will be referred to. This could be another web page, an E-mail address, a file, an anchor within the page, or a trigger for a JavaScript action. The second attribute of concern is **target**. There are four possible values: **\_blank**, which opens the link in a new browser window; **\_self**, which opens the link within the current page and is the default value; and **\_parent** and **\_top**, which are used in coordination with frame-based navigation. This tag should be used in pairs, i.e. `<a href="new_page.htm" target="_blank">Click here.</a>`.

### E-mail Link

Special text (mailto:) is required in order to have a link open up the viewer's default E-mail client and place the e-mail address in the To: field.

```
<a href="mailto:me@mymail.com">E-mail Me</a>
```

### Anchors

Need to navigate to a different area on the same page? First you need to place an "anchor" at the point you want to navigate to using the name attribute, i.e.

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

With a valid anchor name in place, we can link to it using the # sign:

```
<a href="#anchor_name">View Current Page Anchor</a>
```

```
<a href="new_page.htm#anchor_name">View External Page Anchor</a>
```

# MCM 545

# Web Design

with your host Joshua Logsdon

## Absolute and Relative Linking

Let's pretend we are viewing <http://www.mysite.com/news/local.htm>. This page links to an image, <http://www.mysite.com/images/news/josh.jpg>, and an archived file to <http://www.mysite.com/news/archive/local.htm>.

### Absolute Linking

An absolute reference is the complete URL (Uniform Resource Locator) of a file or directory as if you were typing the location into the address field of a web browser. URLs should be in their full format:

`scheme://host.domain[:port]/path/filename`, i.e.

`<a href="http://www.mysite.com/news/archive/local.htm">Past</a>`

### Relative Linking

There are four ways to reference files relative to the current directory, in subdirectories, and in previous directories. The differentiation between them is how the reference starts, with either a file or directory, `./`, `../`, or just `/`.

1) Starting with a file or directory implies that the reference starts from the current directory, i.e. `<a href="archive/local.htm">Past</a>`

2) Using `./` before a reference is the formal declaration to start from the current directory, i.e. `<a href="./archive/local.htm">Past</a>`

3) Using `../` before a reference will start the reference from the previous directory. You can have multiple `../` instances to move through multiple previous directories, i.e. `<a href="../images/news/josh.jpg">See Josh</a>`

4) Using `/` before a reference will always use the site's domain as the starting point of reference, i.e. `<a href="/images/news/josh.jpg">See Josh</a>`

# MCM 545 Web Design

with your host Joshua Logsdon

## Filename Conventions Primer

### Old School — 8.3

There is a old DOS standard of using no more than eight letters for the filename and three for the file extensions, i.e. **pol\_man.htm**. This puts a limit on how descriptive your file names can be and forces you to be more creative with naming. It is an ideal standard for web server compatibility.

### New School — Clear & Mnemonic

Modern web servers support longer file names and extensions. Filenames can be more logical and intuitive, denoting their content while remaining short. Basically, use keywords, i.e. **policy\_manual.htm**. Modern search engines also look at filenames for relevance during searches. If your filename runs together as one word, **policymanual**, it will be interpreted as such, so name wisely.

### What Characters to Use

For web server compatibility and search engine optimization, do NOT use spaces in filenames. Use lowercase letters, numbers (only when necessary), and the underscore character in place of spaces. Why? If you are randomly capitalizing file names, you will have trouble trying to recall your conventions later, just like your customers will. Some web servers are case-sensitive, so if capitalization is messed up, customers will receive an error page. Numbers should be used for dates and only when necessary to avoid confusion in naming conventions.

### ... And Another Thing

The home page of a directory should be named **index.htm**. With most web servers, this is the default file used when only a domain or directory is requested.

# MCM 545 Web Design

with your host Joshua Logsdon

## File Management Primer

### Sorting It All Out

If you have a mess of files, consider making a subdirectory for file types that may require their own managing, like images. If you have a group of files that belong together, consider creating a subdirectory for the group, like a policy directory for many policy files and resources.

Sorting files can be a tedious task, but a tidy and organized site is well worth it. Every time you have to revisit a project, you will thank yourself for keeping it consistent. **Think through and map your file management** to catch potential problems before they become a hassle.

### Shared Resources vs. One-timers

By identifying shared resources and keeping them in an easily accessible place, you will cut down on the site's size and keep your viewer's from downloading duplicate files. Another benefit is that you will only have to update shared resources once instead of once for each instance.

### Don't Go Overboard

Having good naming conventions can help from having to store a particular set of files in a separate subdirectory or having overly-nested directories.