

CGT 141/CPT 141 Lecture 2 Wk 1

Review of Structural and Formatting Tags

Three flavors of HTML (4.01)

- Based on Strict DTD – omits deprecated tags and attributes
- Based on Transitional DTD – less restrictive
- Based on Frameset DTD – includes Transitional DTD plus frames support
- Pay special attention to Chapter 8 discussion of this, as well as the information pertaining to `<!DOCTYPE>` in Chapter 9.

XHTML uses the same 3 flavors: strict, transitional, and frameset. These URL's have been known to change. You may want to look them up yourself first. In this course, we will be using the strict DTD.

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

The required structural tags of an HTML document are: `<html>`, `<head>`, `<body>`

Thus, a minimalist HTML document would look like:

```
<html>
  <head>
  </head>
  <body>

  </body>
</html>
```

While this page above would be completely blank, no title, no content... this following HTML document has both a title and content:

```
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is my first HTML page.</p>
    <p>I hope you enjoy the site.</p>
  </body>
</html>
```

HTML is composed of tags, that is, keywords that identify content in a document as of a particular type.

- For example `<p>` identifies a paragraph.
- Thus, HTML and XHTML use a page metaphor to classify the content of a web page

Tags have optional settings, called attributes that can be defined for them.

- To align a paragraph to the right of the page, you add an attribute:
 - o `< p align="right">`
- All attributes have a default value. The default alignment on a `<p>` tag, for example, is left alignment.
- Most tags have multiple attributes. They can be used in any order. The following two lines are exactly the same:
 - o ``
 - o ``
- When you do not set a value for an attribute, the user agent will render the element based on the default values for its attributes.

Anatomy of a tag (container):

- Tag Skeletons:
 - o `<opening>Content</closing>` (most common)
 - o `<opening>Content`
 - o `<opening>`
- Most tags have a opening and a closing element, with text content in-between:
 - o `<p align="right">This is a paragraph </p>`
 - o `<h1>This is a heading at level 1</h1>`
 - o `<h3>This is a heading at level 3</h3>`
- Some do not, often called “empty” or “standalone tags”:
 - o `<hr>` or `<br>`
 - o ``
- If you wish to set values for attributes, you do so in the opening of the tag:
 - o ``
- Tags may also be nested inside of one another. For example you can create a bolded paragraph by using:
 - o `<p><b>This is a bold paragraph</b></p>`
 - o Just make sure you nest properly. The following is incorrect:
 - `<p><b>This is an incorrectly nested set.</p></b>`

General Rules:

- Editing can be done with a simply text editor (such as Notepad or Wordpad). Simply add the extension .html (preferred) or .htm on the end of the ASCII text file.
- Line breaks, tabs or multiple spaces in the ASCII file are ignored by the user agent.
- All tags that can be closed, should be closed.
 - o Browsers will often allow you to be lazy in that you can get away with not closing a tag. Don't do it! It will cause you to develop bad habits that hurt you later (xHTML, XML).

- HTML is not case sensitive. But you should use all lowercase letters for all tags and attributes. HTTP compression is better (files are smaller) when tags and attributes are lowercase.
 - o You must use all lowercase letters for this course
- XHTML IS case sensitive.
- All attribute values should be quoted.
- Unrecognized tags or attributes the browser will often ignore. But make sure what you use is a valid entity.
- Comments can be added for internal documentation (to help you remember what you did and why).
 - o `<!-- This is an example of a comment -->`
- Do not use deprecated, nonstandard or browser-specific tags.
 - o For left, center, or right alignment, try the following:
 - `<span align="center"> some text </span>`
 - Where center could also be right or left
- Special characters must be entered using **escape sequences**.
 - o Can be entered as ASCII codes or specific keywords
 - o General form for escape sequences/character entities:
 - `&#number;` or `&word;`
 - o Examples:
 - `&` or `&`
 - creates an ampersand (&) in text
 - ` ` or ` `
 - creates a non-breaking space
 - Listing of all escape sequences in Appendix F (pg. 577) of the book.

Hierarchy of HTML Tags

- All tags fall into one of the following groups:
 - o Structural tags – tags required in every document; they define the sections of an HTML document
 - o Head tags – tags that can be used in the `<head>...</head>` section.
 - o Block-level tags – tags designed to format blocks of text. Easily identifiable by the carriage return/line feed that follows them.
 - o Text-level tags – tags designed to format words or phrases; this can be further divided into two groups
 - Logical text tags – tags that are descriptive, as opposed to formatting, oriented:
 - `<strong>...</strong>`
 - `<emphasis>...</emphasis>`
 - Physical text tags – tags that deal with strict formatting
 - `<b>...</b>`
 - `<i>...</i>`

- Invisible tags – predominantly used with Cascading Stylesheets; has no visual effect on elements.
- o Empty/Standalone tags – these tags are often unclosed tags and are designed to do special things:
 - `<hr />` – horizontal lines
 - `<br />` – line break

Wellformedness constraints

- Throughout this course, we will be following the rules for wellformedness.
- Wellformedness refers to XML standards
- Conforming to these XML constraints are what turns a regular HTML document into an XHTML document.
- These are the rules that will be followed all semester long:
 - o There can only be **one root element** (sometimes called document element)
 - Example - `<html>` is the root element in this example
 - `<html>`

```

              <head></head>
              <body>

              </body>
            </html>
              
```
 - o Elements must be **properly nested**
 - Wrong: `<strong> <em> some text </strong> </em>`
 - Correct: `<strong> <em> some text </em> </strong>`
 - Browsers have been dumbed down enough to accept improper syntax in HTML, basically to make it ‘easier’ for anybody to create HTML.
 - o All tags must be **closed**
 - Wrong:
 - `<ul>`

```

              <li> one
              <li> two
              <li> three
            </ul>
              
```
 - Correct:
 - `<ul>`

```

              <li> one </li>
              <li> two </li>
              <li> three </li>
            </ul>
              
```

- o All single tag elements must be **closed**
 - The
, <hr>, and tags will be changed to:
 -
 becomes

 - <hr> becomes <hr />
 - becomes
- o All values must be in **quotes**
 - Wrong:
 -
 - Correct:
 -
 - The standard for this course and in general is to use double quotes. It will work with single quotes; however, double quotes are required for this course. This becomes important later when we begin nesting quotes inside of other quotes. Use double quotes.
- o All elements (tags) and attributes must be **lower case** and are **case sensitive**.
 - Wrong:
 - <TABLE border="1" width="680"> ... </TABLE>
 - <table BORDER="1" width="680"> ... </table>
 - <table> </TABLE>
 - <table> </Table>
 - Correct:
 - <table border="1" width="680"> ... </table>
 - <table> </table>
- o Attribute minimization is forbidden
 - Wrong:
 - <option selected>
 - <input disabled>
 - <input checked>
 - <frame noresize>
 - Correct
 - <option selected="selected">
 - <input disabled="disabled">
 - <input checked="checked">
 - <frame noresize="noresize">