

Unit 2, cont.

September 14

HTML, Validating your pages,
Publishing your site

Tables in HTML

- Tables *should* only be used for data
- However, they are also used for layout of web pages
- Try to keep tables for layout simple
 - Don't nest tables within tables with tables, etc.
 - Creates havoc for trying to fix/read HTML
- Important tag to know how to use

<table> Tag

- Indicates the beginning of a table (imagine that!)
- Commonly used attributes:
 - “border” : takes a value in pixels and creates a border around the table
 - “width” : takes a value in pixels or percent of horizontal space available
 - Better to specify a percent than pixels
 - “cellspacing” : space between table cells, in pixels or percent
 - “cellpadding” : space between the cell border and the contents of the cell, also in pixels or percent

More <table> Attributes

- Other attributes include:
 - “rules” : borders between cells
 - You’ll most likely only use the
 - “none” value which has no borders between cells
 - “all” which has all borders between cells
 - See XHTML 1.0 reference for the values it can take
 - “bgcolor” : will change the background color of the table

Tags inside <table>

- Three most important and widely used tags are <tr>, <td>, and <th>
- <tr>, or table row
 - Defines a row inside the <table> tag
 - The only attribute you'll probably use is the bgcolor attribute
 - Will change the background color for that row

<td> Tag

- <td> denotes a table data cell
- Is contained within <tr> tags
- Common attributes:
 - rowspan: how many rows the cell should span
 - colspan: how many columns the cell should span
 - bgcolor: background color of that cell
 - nowrap: takes the value of “nowrap” if you do not want the contents of the cell to automatically do line wrapping
 - This can create tables which require horizontal scrolling to view --- annoying!

<th> Tag

- <th> is the tag which specifies a table header
- Acts much like the <td> tag
 - However, this is logical markup
 - A table header does not usually contain table data
- You should use table headers when creating tables used for actual data
 - i.e. not for layout purposes

Table Examples

- In class tables

Tag Types

- Have seen that some tags can only go inside other tags
- It isn't arbitrary
- 4 main classes of tags:
 - Top level tags
 - Head tags
 - Block level tags
 - Inline tags

Top Level Tags

- Define overall structure of the document
- The one's you're familiar with
 - `<html>` specifies this is an html document
 - `<head>` contains the head information which is usually not displayed as part of the page
 - `<body>` where the “meat” of your page goes

Head Tags

- Head tags go inside the `<head>` top level tag
- Give information *about* the page, not things to display *on* the page
- Examples:
 - `<title>` (mandatory for this course!) Gives the page's title at the top of the browser
 - `<link>` we'll use this to apply style sheets later
 - `<meta>` provides *metadata*, such as keywords, author, etc.

Block Level Tags

- Probably most widely used
- Block level tags take up vertical space on the page
- Go inside the `<body>` tag
- Some you've seen:
 - `<p>` paragraph
 - `<hr>` horizontal rule
 - `<h1>` heading tag
 - `` ordered list

Inline Tags

- Found inside block-level tags
- Called “inline” because they are placed within the text
 - Contents of inline tags appear *beside* the last text
 - vs. contents of block-level tags which appear *below* last block-level tag
- Change the way part of a line looks
- Examples:
 - `` insert an image
 - `` emphasis
 - `<a>` insert a hyperlink

Lang Attribute

- Specifies the language of the text in the element
- Can be applied to most tags
- Helps search engines to categorize the page properly
- Is also used by browsers to render the page properly
 - Could even translate it automatically

How to use Lang Attribute

- Can be added to almost any tag
- If added to `<html>` it will specify that the entire document is in a certain language
- Other common places to use lang attribute are:
 - `<p>` : paragraph
 - `<q>` : quote
 - `<blockquote>` : long quote

Examples of Lang Attribute

- For the entire document:
 - `<html lang = “en”>`
 - The page is all in English
- For a paragraph
 - `<p lang = “fr”>`
 - The paragraph is in French
- Other common ones you might use are:
 - zh Chinese
 - ja Japanese

xml:lang to Specify Language

- Future versions of XHTML will use “xml:lang” to specify the language instead of just the lang attribute
- Can use both specifications without problem:
 - `<html lang = “en” xml:lang=“en”>`
- Good idea to do so

Mistakes in HTML

- Common browsers often display HTML which actually contains errors
- Just because it looks right, doesn't mean it is right
- Not all browsers are this “smart” to overlook mistakes in HTML documents
- Example:
 - `<i>Badly nested tags</i>`
 - Most browsers will display this as:
 - ***Badly nested tags***
 - It was what was meant, but not correct!

HTML Validators

- HTML validators, or just validators, are programs used to find errors in HTML
- Check the web page against the formal HTML definition and reports errors (and warnings sometimes)
- In order to know which version of HTML the page should be checked against, it must be specified in the HTML document

Document Type Declaration

- *Document type* declaration is used to specify the version of HTML that is being used
- Must be the first line of the HTML document
 - Before <html> tag

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

- States that the document is written in XHTML 1.0 as defined by the W3C
- ALSO that it is *strict* XHTML

Transitional Doctype

- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
- This is what you'll use for assignment 1
- The transitional type allows you to use deprecated tags and attributes
 - Strict does NOT
- ALL of your html documents MUST have a specified doctype to validate

Namespace Attribute

- In the `<html>` tag you must add:
 - “xmlns = <http://www.w3.org/1999/xhtml>”
- Your `<html>` tag would then read:
`<html xmlns = “http://www.w3.org/1999/xhtml”>`
- So what the heck is this?
 - Basically it specifies where the definition for `xhtml` is coming from
 - `xmlns` comes from XML NameSpace
 - In this case its from the `w3c`

How to Check a Page

- Use a validator from the list on the course website
- In class example

How to Upload a Page

- 1. usermin upload/download on at:
 - <http://cmpt165.cs.sfu.ca>
- 2. Use an SCP file transfer client
 - Like WinSCP
 - Some can be found on Course Software of course webpage
- 3. Manually using scp
 - Primarily for those using linux/unix

Usermin Demonstration

Other Important Publishing Concerns

- All of your html files that you want to be seen need to be put in the public_html directory
- This includes any directories, etc. Your entire website should be contained within public_html
- Your homepage will be <http://cmpt165.cs.sfu.ca/~username>

How to Create the Homepage

- Your homepage should be a file named index.html
- Why?
 - Servers are configured to look for a default file name to load
 - cmpt165.cs.sfu.ca has a default of “index.html”
- With the default page specified, you don't have to type:
 - <http://cmpt165.cs.sfu.ca/~username/index.html>
 - Automatically loaded with: <http://cmpt165.cs.sfu.ca/~username>
- Imagine if for every website you also had to know the default file name
- Using something like index.html makes it easier

Questions