

# NestedTags/Spaces/Comments

- Nested tags are possible but don't overlap sets of them. Avoid the following:

`<em><strong>Message</em></strong>`

- Browser Processing

- Multiple spaces are converted to one space.

John

Mary

Peter

John Mary Peter

- Line returns are ignored.

- Comments

- Represented by `<!-- -->` Note: (two sets of double -)
- Examples
- `<!--The html code example starts at this point-->`
- Comments can not be nested.

# HTML Editors

- Text Editor
  - Any text editor (e.g., wordpad, notepad, pico, etc.)
- HTML Editors
  - Utilities designed to write HTML
  - Examples: CoffeeCup HTML Editor, HTMLjive
- Authoring tools
  - Frontpage
  - Dreamweaver – Fairly complex.
  - NVU – Free and available for (Windows, Linux, Mac) <http://www.nvu.com/>
- List of editors can be found at:  
[http://dir.yahoo.com/Computers\\_and\\_Internet/Software/Internet/World\\_Wide\\_Web/HTML\\_Editors/](http://dir.yahoo.com/Computers_and_Internet/Software/Internet/World_Wide_Web/HTML_Editors/)
- Recommended:
  - Komodo Edit [http://www.activestate.com/Products/komodo\\_edit](http://www.activestate.com/Products/komodo_edit)

# Frequently Used Tags

- Heading tags
  - `<h1> text </h1>`  
`<h2> text </h2>` ... and so on until `<h6> text </h6>`
  - Higher numbers imply smaller headers.
- Paragraph tag
  - `<p> paragraph </p>`
- Code – Use to define computer code
  - `<code> </code>`
- Horizontal Line - `<hr />`

# Frequently Used Tags

- Emphasis
  - `<em> text here </em>` Text usually rendered in italics
  - `<strong> text here </strong>` Text usually rendered in bold
- Super/Sub script
  - `<sub> text here </sub>`
  - `<sup> text here </sup>`
- Quotations
  - `<q> quote here </q>`
- Line Breaks
  - `<br />`
- Verbatim (text displayed exactly as it appears)
  - `<pre> text here </pre>`
- **Example: HtmlDoc.html**

# Lists

- **Unordered lists**
  - `<ul>` `</ul>` tags to represent beginning and end.
  - `<li>` `</li>` to represent elements in the list.
  - **Example: Lists.html**
- **Ordered lists**
  - `<ol>` `</ol>` tags to mark beginning and end.
  - `<li>` `</li>` to represent elements in the list.
- **Definition lists**
  - Consist of terms and definitions like in a glossary.
  - Tags - `<dl>` `</dl>`
  - Terms specified using `<dt>` `</dt>` and definitions with `<dd>` `</dd>`
  - **Example: Lists.html**
- **Nested lists**

# Image Inclusion

- We can include an image using the img tag  
``
- **Example: Image.html**
- Although the width and height attributes are not required they are highly recommended. (They can also be set through CSS).

# Links

- Link – connection between web resources.
- Hypertext links are created using the <a> (anchor) tag.
- The link can be text :
  - `<a href="http://www.cnn.com">CNN Web Page</a>`
  - Notice that you **need to** specify the protocol (http://)
  - **Example: Links.html**
  - The URL can be absolute or relative.
- The link can be an image:

```
<a href="http://www.umd.edu"></a>
```

# Tables

- To define a table we use the `<table>` tag
  - Border attribute controls table's border.
  - By default borders are not visible.
- Basic tags are associated with tables
  - `<tr>` - defines a row.
  - `<td>` - defines a data element.
  - `<th>` - define a header data element.
  - `<caption>` - provides a caption for the table.
    - Must appear after the `<table>` tag.
    - Must be used only once.
- **Example: Tables.html**

# Character Entity References

- Special Characters can be specified by
  - Name specification - *&name;*
  - Numeric specification - *&#xxx;*
- Commonly used characters

<i>Copyright</i>	<i>&amp;copy;</i>
<i>Registered Trademark</i>	<i>&amp;reg;</i>
<i>&amp;</i>	<i>&amp;amp;</i>
<i>&lt;</i>	<i>&amp;lt;</i>
<i>&gt;</i>	<i>&amp;gt;</i>
<i>Non break space</i>	<i>&amp;nbsp;</i>
- **Example: CharacterReferences.html**
- Complete list at:  
<http://www.w3.org/TR/html4/sgml/entities.html>

# Block Elements/Inline Elements

- Comparison
  - Block elements begin on new lines whereas inline elements don't.
  - Block elements create larger structures (allow you to define the large structure of your document) whereas inline elements don't.
- Block Elements Examples
  - Paragraphs (<p>), Headings, Lists, Tables, Division (<div>), Block Quotations, Preformatted Text (<pre>)
- Inline Element Examples
  - Anchors (<a>), Images (<img>), Line Breaks (<br />)
- Block elements may contain other block elements, inline elements, and data. Some block elements may not contain other block elements.
- Inline elements may contain inline elements and data.

# Inline Elements in Block Elements

- Why the following example does not validate?
- `<img>` should be in a block element (e.g., `<p></p>`)
- Example: `validationProblem.html`

# Suggestions for Writing HTML Code

- Add the corresponding end tag immediately.
- Use indentation.
- Have a consistent style.
- Use comments to separate sections of your code.
- Validate your code as you develop it (not at the end).

# Googles Page Creator

- <http://pages.google.com/>
- You need a gmail account.
- Provides free hosting.
- Your address will be:
  - <http://YOURGMAILID.googlepages.com>

# CSS (Cascading Style Sheets)

- Official W3C standard for controlling presentation
- Specification: <http://www.w3.org/TR/CSS21/>
- Style Sheets
  - Text file with rules. It includes no html.
  - Style sheets files use a .css extension.
  - Allows you to apply typographic styles (font size, line spacing, etc.)
  - Allows you to apply spacing instructions.
  - Allows you to have page layout control.
  - Smaller html files by avoiding redundancy in style specification.
  - Easy update a collection of pages by updating only a single file.
  - Example: ExternalFile.css
- Why CSS? <http://www.csszengarden.com/>

# Rules

- Rule - Basic element of a style sheet.
- Rule - describes the formatting associated with a page element.
- Rule format

## *selector declaration*

*selector* – identifies what should be styled in a web document (e.g., h1, p).

*declaration* – what and how that portion of the web document should be modified.

- declaration - consists of *property: value* pair(s) enclosed in { }
- Examples:

```
h1 {color: green}
p {
    font-size: 10px;
    color: red;
}
```

- Notice there is a space after the colon (;)
- Popular properties – color, font-family, font-size, text-decoration
- HTML Dog CSS Properties –
- <http://www.htmldog.com/reference/cssproperties/>

# CSS Validator

- <http://jigsaw.w3.org/css-validator/>
- Notice you have three choices
  - by URI
  - by File Upload
  - by direct input

# Colors

- You can specify colors using one of the following predefined colors:

*yellow, white, teal, silver, red, purple, orange, olive, navy, maroon, lime, green, gray, fuchsia, blue, black, aqua*

- Source for colors

[http://www.w3schools.com/html/html\\_colors.asp](http://www.w3schools.com/html/html_colors.asp)

- You can specify a color by indicating the red, green and blue components. For example, all the following are equivalent:
  - red
  - `rgb(255,0,0)`
  - `#ff0000`

# Kinds of Selectors

- **Type Selectors** – Those based on the name of an HTML tag
  - `p { color: red; }`
- **Pseudo-classes** – attached to selectors to specify a state. Four popular pseudo-classes are
  - `a:link` – initial color of a link.
  - `a:visited` – color for a visited link.
  - `a:hover` – color when mouse hover over link.
  - `a:active` – color during the clicking of the link.
- **Class Selectors** – Allow us to apply the same CSS rule to different elements
  - Use to create a style you need to apply many times in your document.
  - Created with a period (also known as full stop).
  - Example: `classIdSelectors.html`, `Selectors.css`
- **ID Selectors** – Like class selectors but appear only once in the document
  - Used when you need to apply a style only once in your document
  - Created using `#`
- **Example:** `Selectors.html`, `Selectors.css`