**Course Title** – Beginning Web Programming  
**Course Prefix and Number** – ITSE 1311  
**Department** - Computer Science  
**Division** – Technology & Business  

**Course Type:** (check one)  
- Academic General Education Course (from ACGM – but not in WCJC Core)  
- Academic WCJC Core Course  
- WECM course (This course is a Special Topics or Unique Needs Course: Y☐ or N☒)  

Semester Credit Hours # : Lecture hours# : Lab/other hours #  
3:2:2  

**Equateed Pay hours for course - ☒**  

**Course Catalog Description** - Skill development in web page programming including mark-up and scripting languages.  

**Prerequisites/Corequisites** - None  

**Approvals – the contents of this document have been reviewed and are found to be accurate.**  

<table>
<thead>
<tr>
<th>Prepared by</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stephanie Dees</td>
<td>[Signature]</td>
<td>11/25/2009</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Department Head</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stephanie Dees</td>
<td>[Signature]</td>
<td>11/25/2009</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Division Chair</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stephanie Dees</td>
<td>[Signature]</td>
<td>11/25/2009</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vice President of Instruction or Dean of Vocational Instruction</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Signature]</td>
<td>[Date]</td>
<td>12/1/09</td>
</tr>
</tbody>
</table>
I. Topical Outline – Each offering of this course must include the following topics (be sure to include information regarding lab, practicum, clinical or other non lecture instruction):

<table>
<thead>
<tr>
<th>Topics Covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Brief description of Common Tags (Chapters 1-3) Note compliance with XHTML &amp; XML specifications</td>
</tr>
<tr>
<td>➢ Storyboarding:</td>
</tr>
<tr>
<td>➢ Page Footers:</td>
</tr>
<tr>
<td>➢ Lists (Chapter 4)</td>
</tr>
</tbody>
</table>

| ➢ | Description: identifies a numbered list of items |
• **Type:** container
• **Attributes:** including, but not limited to the following:
  • compact deprecated
  • start deprecated in favor of: style
  • type deprecated in favor of: style="list-style-type"
  • value deprecated in favor of: style

• `<ul>` unordered list `<ul>`
  • **Description:** identifies a list of items with no specific order implied
  • **Type:** container
  • **Attributes:** including, but not limited to the following:
    • compact deprecated
    • type deprecated in favor of: style="list-style-type"
    • example:
      • OLD: `<ul type="square">`
      • NEW: `<ul style="list-style-type: square">`

• `<li>` list item `<li>`
  • **Description:** identifies a specific item in an ordered, unordered, menu, or directory list
  • **Type:** container
  • **Attributes:** including, but not limited to the following:
    • type deprecated in favor of: style="list-style-type"
    • value deprecated in favor of: style

• **Notes:**
  • can contain text images and other lists
  • if within directory or menu list, then it **cannot** contain other lists or blocked elements

□ Combining/Nesting

• `<dl>` definition list `<dl>`
  • **Description:** identifies a term and its definition as part of a glossary-like list
  • **Type:** container
    • <dt>definition term</dt>
      • **Description:** identifies term in a definition list
    • <dd>definition definition</dd>
      • **Description:** identifies definition in a definition list

• `<menu>` menu list `<menu>`
  • **Description:** identifies a single column list of no implied order
  • **Type:** container
  • **Note:** deprecated in favor of unordered list

• `<dir>` directory list `<dir>`
  • **Description:** identifies a multicolumn directory list of no implied order
  • **Type:** container
  • **Note:** deprecated in favor of unordered list

• **Style – wider range of values**
  • example: 3 list-style’s:
    • list-style-position: outside: inside
    • list-style-image: url("pepper.gif")
    • list-style-type: square

➢ Formatting (Chapter 5)

• **Terms**
  • glyphs – letters, numbers, symbols
  • fonts – typeface, weight, style, size
    • typeface
      • serif – line segments upper and/or lower end of letter/character
      • sans serif – without serif (easier to read)
      • cursive - handwriting
  • spacing
    • proportional – width varies based on letter width
    • monospace – each letter occupies same width
  • style sheets – set of design rules for HTML document
    • document-level style sheet – rules for document defined in the `<head>` element – affects text within `<body>`
    • container of document
    • external style sheets (.css, .css1, .css2, etc. – Ch. #8)
• <style> document style changes </style> p.111
  o Description: identifies formatting changes to elements that appear in the document’s <body> container
  o Type: container
  o Attributes: dir,lang,media,title,type
  o Placement: <head> container
  o Syntax:
    tag name [...tag name] { property:value; [...property:value;] }

  design considerations:
  o establish design before begin coding
  o comment lines – old browser compatibility
  o KISS
  o http://www.w3.org/markup & http://www.w3.org/style
  o point (pt) = 1/72 inch p.110

• text-align: left | center | right | justify p.115

• font-family: serif | sans-serif | cursive | fantasy | monospace | [Arial, Courier, … ]
  o names that contain spaces must be in quotes. (ex: P{font-family: “Times New Roman”})
  o use generic family names (w3c recommendations) – must not be in quotes:
    cursive, fantasy, monospace, sans-serif, serif

• font-size: absolute vs relative font size changes
  o relative:
    o larger – 20% larger
    o smaller – 20% smaller
    o percent (example: P{font-size: 40%} makes text 40% of current size)

• font-style: normal | italic | oblique

• font-weight: normal | bold | bolder | lighter | 100-900
  o 100 – 900 ; css2; 100-lightest; 900-darkest; 400-normal; 700-bold
  o relative: bolder | lighter -> increments of 100

• font-variant: normal | small-caps
  o example: (H4 {font-variant: small-caps} (This is small caps)

• font-stretch: normal | wider | narrower | ultra-condensed thru ultra-expanded

• font: font-style | font-variant | font-weight | font-size | font-family
  o CSS2 specs – shorthand notation
  o Example: h1 {font: italic bold 30pt sans-serif}

• <basefont> basefont</basefont>
  o Description: originally designed to change the size of the font within a document
  o Special note: deprecated (HTML 4.0) in favor of style

• <font>font of text</font>
  o Description: changes the color, typeface or size of the font
  o Special note: deprecated (HTML 4.0) in favor of style

  INLINE STYLES – 2 general forms
  o PHYSICAL STYLES – describe the way the text is to look in a browser
  o LOGICAL STYLES – describe the way text within the container is used
  o <abbr> - logical style for an abbreviation (124)
  o <acronym> - logical style for an acronym (124)
  o <address> - logical style for address (123)
  o <b> - physical style change for darker typeface (121)
  o <big>- physical style change to increase font size by 1 each time it is used (121)
  o <blockquote> - logical style for a long quote (125)
  o <cite> - logical style for a citation (125)
  o <code> - physical style change to a monospaced typeface (121)
  o <del> - logical style for deleted text (126)
  o <dfn> - logical style for a definition (124)
  o <em> - logical style for emphasized text (124)
  o <i> - physical style change to slanted typeface (121)
  o <ins> - logical style for inserted text (126)
  o <kbd> - physical style change to a monospaced typeface (121)
  o <q> - logical style for a quote (126)
  o <s> - deprecated physical style to draw line through text (121)
  o <samp> - logical style for sample output (124)
  o <small> - physical style change to decrease font size by 1 each time it is used (121)
  o <strike> - deprecated physical style to draw line through text (121)
  o <strong> - logical style for strongly emphasized text (123)
  o <sub> - physical style for setting glyph a half line below text (121)
  o <sup> - physical style for setting glyph a half line above text (121)
  o <u> - physical style change to a monospaced typeface (121)
  o <var> - logical style for a variable value (124)
• SPECIAL CHARACTERS
  o May be identified by referencing the Latin-1 character set
    ▪ Latin-1 character set also known as ISO 8859-1
    ▪ contains 236 characters commonly used in English, French, German, and other Indo-European languages
    ▪ basic character set for the Unicode characters (ISO 10646-1) identified in the HTML 4.0 specifications
  o Syntax: & (standard name | octothorp hex code);
    ▪ Example #1: &lt; (lt – standard name for ‘Less than’ symbol ‘<”)
    ▪ Example #2: &amp;#60; (hex. ‘060’ is hex code for ‘<”)
    ▪ Example #3; © or © (copyright symbol)
  o Other examples:
    ▪ Browsers cannot resolve “< “ (less than symbol followed by a space), so it may be simply ignored as an HTML tag, but displayed.
    ▪ Browsers cannot resolve “<tag>”; not a valid tag, so it may be ignored & not displayed.
    ▪ Browsers may attempt to resolve “<S and the>” as a valid tag & display the following text with the strikethrough font.

• &lt;pre&gt; pre-formatted text&lt;/pre&gt;
  o Description: identifies preformatted text that the browser is instructed not to alter
  o Type: container
  o Attributes: class, dir, id, lang, onClick, onDbClick, onKeyDown, onKeyPress, onKeyUp, onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, style, title, width
  o Special note: replaces obsolete <plaintext>, <listing>, and <xmp> containers.
  o turns off formatting

➢ Images – A Picture Is Worth A Thousand Words (Chapter 6)
• Image Formats
  o GIF – Graphics Interchange Format
    ▪ 8-bit color
    ▪ 3 formats:
      ▪ Plain
      ▪ Transparent
      ▪ Animated
    ▪ Lossless compression
  o JPEG – Joint Photographers Experts Group
    ▪ 8-bit or 24-bit color
    ▪ Lossy compression (therefore, smaller than GIF)
  o PNG – Portable Network Graphics
    ▪ newer
    ▪ 8-bit or 24-bit color
    ▪ Lossless compression
    ▪ open standard
  o SVG – Scalar Vector Graphics
    ▪ specifications released by W3C in October 2001
    ▪ small
    ▪ easily scaled
    ▪ may be animated
    ▪ XML-compliant
  o Others (not recommended for HTML documents):
    ▪ PDF – Portable Data Format
    ▪ TIFF – Tagged Image File Format
    ▪ BMP – Bit Mapped Picture
    ▪ PCX – older format developed by Zsoft
• image editors:
  o StarOffice Image (freeware)
  o Paint Shop Pro (shareware)
  o Adobe’s Photoshop
• &lt;img /&gt;
  o Description: insert graphs, photographs, line art, or other images into screen display.
  o Type: empty
  o Attributes - including, but not limited to the following:
    ▪ src – source; required
    ▪ alt – description; required
    ▪ longdesc – long description
    ▪ height and width – define size in pixels
    ▪ (horizontal) align = “left” | “right” p.147
    ▪ style attribute float property recommended
      ▪ ex: style="float:left"
    ▪ no “center” – use &lt;center&gt; or &lt;div&gt;
    ▪ (vertical) align = “top” | “middle” | “bottom” p.152
    ▪ border
    ▪ vspace and hspace
• **Line Breaks**  
  o `<nobr>no line break</nobr>` p.153

• **Centering Images**  
  o `<center>center text</center>` p.154
  - **Description:** center element within screen display.
  - **Type:** container
  - **Attributes:** none
  - **Note:** deprecated
  o Centering a paragraph
    - `<p align="center">`

• **Image Borders**  
  o `border-style: dotted | dashed | solid | double | groove | ridge | inset | outset | hidden | none` p.156
  o `border-width: thin | medium | thick | absolute | inherit` p.156
  o `border-width: 4 pixels in I.E. & 3 pixels in Nav. & Opera` p.156
  o `border-color: color# | transparent | inherit` p.156

• **Image Margins**  
  o `margin: margin-width in pixels | inherit` p.158
  o `vspace & hspace – deprecated in favor of style` p.158

• **Images as Bullets**  
  o `list-style-image: url | none | inherit` p.160
  o `definition list – alternative to list-style-image property` p.160

• **Image Links**  
  o `<a href="music.html">` <img src="movies.gif" alt="Music" /> `</a>` p.161

### Tables – Data in Rows and Columns (Chapter 7)

• `<table>` **table** `</table>`  
  o **Description:** defines the rows, columns, and caption that make up a table.
  o **Type:** container
  o **Attributes:** including, but not limited to the following:
    - `border`  
      - `ex: <table border="5">` defines 5-pixel wide border
      - default – no border if:
        - no border= attribute is defined or
        - border="0"
      - `style="border-style: solid; border-width: 5px;"`
    - `rules` – none,rows,cols,all,groups
      - `ex: <table rules="rows">`
    - `frame-box, border, void,above,below,hsides,vsides,lhs,rhs`  
      - `ex: <table frame="void">` defines no sides (default)
    - `width` – pixels or percent screen width (percent recommended)
      - `ex: <table width="500px">`
      - `ex: <table width="25%">`
    - `bgcolor vs. style="background-color: #FFFFFF;"`
    - `summary = text string (for aural browsers)`
      - `ex: <table summary="This is a test">`
    - `border-width (style="border-width: 5px;">` p.178
      - `border-style (style="border-style: solid;">` p.178
      - `border-color (style="border-color: #FFFFFF;">` p.180
    - `float – left, right, none (style="float: right;">` p.180
    - `cellpadding – pixels – space within a cell`
    - `cellspacing – pixels – space between cells`

• `<tr>` **table row** `</tr>`  
  o **Description:** defines a table row.
  o **Type:** container
  o **Attributes:** including, but not limited to the following:
    - `align – left | right | center` p.184
    - default left for data and center for headers
    - `valign – center | top | bottom | baseline` p.184
    - `char & charoff – not yet fully supported`

• `<td>` **table data** `</td>`  
  o **Description:** defines table data (one cell).
  o **default:** left align regular font
  o **Type:** container

• `<th>` **table header** `</th>`  
  o **Description:** defines table header (one cell).
  o **default:** center align bold font
  o **Type:** container
- **Attributes**: including, but not limited to the following:
  - align
  - bgcolor
  - valign
  - colspan & rowspan
  - abbr, headers, scope, axis – not yet supported
  - style
  - width & height (deprecated in favor of style)
  - whitespace: nowrap (deprecated in favor of style)

- `<caption>`
  - **Description**: defines a table caption
  - p.193

- **Order of Precedence for attributes in a table**
  - cell level / row level / table level / style sheet
  - p.187

- Centering a table: (see ‘Centering Images’, p.154)
  - `<center>...</center>`
  - `<p align="center">...</p>`
  - p.187

- `<colgroup>`
  - **Description**: creates structural divisions in a table that can be independently formatted
  - **Type**: container
  - **Attributes**: including, but not limited to:
    - span – number of columns to span
    - width – number of pixels | percentage | proportional
  - ex: `<colgroup span="2" width="30">`<colgroup>
  - p.195

- `<col>`
  - **Description**: allows grouping of attributes for table columns
  - **Type**: empty
  - **Attributes**: including, but not limited to:
    - span – number of columns to span
    - valign – center | top | bottom | baseline
    - width – number of pixels | percentage | proportional
  - ex: `<col span="2" width="30"/>` defines minimum width
  - p.197

- `<thead>`
  - **Description**: not yet implemented in popular browsers
  - p.199

- `<tfoot>`
  - **Description**: not yet implemented in popular browsers
  - p.200

- `<tbody>`
  - **Description**: not yet implemented in popular browsers
  - p.201

- **Styles – Some Have It and Some Don’t (Chapter 8)**
  - **Cascading Style Sheet**
    - p.208
      - cascading order of precedence:
        - inline
        - document-level within `<style>` container (in `<head>` container)
        - external files (.css files)
        - default browser values
  - **External Style Sheets (.css filename extension)**
    - strongly recommended for XML and XHTML specifications
  - **<link>**
    - **Description**: identifies external files that relate to the document in various ways.
    - **Type**: empty
    - **Note**: must be coded within the `<head>` container
    - **Attributes**: including, but not limited to the following:
      - type="text/css"
        - defines ‘text’ and follows W3C’s cascading style sheet specifications
      - rel="stylesheet" – defines that external file is related to the current document as a stylesheet
      - href – identifies the path to the stylesheet
    - ex: `<link rel="stylesheet" href="abc.css" type="text/css" />`
    - p.211
  - **Classes of Styles**
    - a **class** is a name given to a set of properties assigned to different HTML elements. Class names always begin with a period (.). Classes may be:
      - generic – applied to a variety of tags
      - tag-level – assigned to a specific tag
    - **Syntax**:
      - `[tag name] . class name { property:value; [...property:value;] }
  - **Common Measurement Values**
    - em – width of letter ‘M’ in font-size of active font
  - **Style Sheet Elements Divide and Span**
    - p.215
  - `<div>`
    - **Description**: used to divide document into large sections spanning several elements, optionally applying formatting
  - p.218

- Administrative-Master Syllabus
- form approved June/2006
- revised Nov. 2009
• Common Style Properties p.219
  o Text Manipulation and Alignment p.220
    ▪ color
    ▪ letter-spacing
    ▪ line-height
    ▪ text-align
    ▪ text-decoration
    ▪ text-indent
    ▪ text-shadow
    ▪ text-transform
    ▪ vertical-align
    ▪ word-spacing
  o Background properties p.222
    ▪ background-attachment
    ▪ background-color
    ▪ background-image
    ▪ background-position
    ▪ background-repeat
    ▪ background
• Changing Link Colors (not recommended) p.224
  o a:link (Example in body element: a:link {color: #FFFFFF;})
  o a:visited
  o a:active
• Box Definitions p.224
  o CSS Box Model – rules for placing white space in an HTML element containers’ contents within padding, within a border around the padding, within the margins defining the outside edge of the content
  o Four values of properties - order assumed: top, right, bottom, left
    ▪ width
    ▪ height
    ▪ visibility
    ▪ margin
    ▪ white-space
    ▪ border
    ▪ padding
    ▪ float
    ▪ clear
• Paged Media p.233
  o page-break-before
  o page-break-after
  o page-break-inside
  o page
  o size
  o orphans
  o widows
  o marks
  o mark-offset

➤ Multimedia: Watch Your Inclusions (Chapter 9) p.242
  o Multimedia – sounds, video, animation and other elements of a page that are not static images and text.
  o Inline using image tag
    ▪ Example: <img src="bug.png" alt="Ladybug" />
  o External using Anchor tag
    ▪ Example: <a href="bug.png">Ladybug</a>
  o Inserting Image Using Object Tag (NEW! – only works in Opera – p.245)
    ▪ Example: <object data="bug.png" type="image/png">
      ▪ Description: initiates the execution of a Java applet
      ▪ Type: container
      ▪ Attributes: including, but not limited to:
        ▪ code - path and name of applet file (required)
        ▪ height - required
        ▪ width - required
      ▪ Note: deprecated in favor of <object> element
      ▪ Example: <applet code="UFO_Attack.class" width="200" height="400"></applet>
      ▪ NOTE: MAY NOT HAVE AUTHORITY TO EXECUTE APPLET IN COMPUTER LAB. (See INTERNET OPTIONS/Security Tab/Custom Level/ActiveX Controls & Plugins/Automatic Prompting for ActiveX Controls/Enable - & BE SURE IT ALLOWS YOU TO SAVE CHANGE (Check to see if modification was changed, i.e., that you have authority to change settings) - Fall 2004
    ▪ <object> insert object </object>
      ▪ Description: identifies image, html text, sound video, or other external element for inclusion within the page
  o <applet> java applet</applet> (p.243)
    ▪ Description: initiates the execution of a Java applet
    ▪ Type: container
    ▪ Attributes: including, but not limited to:
      ▪ code - path and name of applet file (required)
      ▪ height - required
      ▪ width - required
    ▪ Note: deprecated in favor of <object> element
    ▪ Example: <applet code="UFO_Attack.class" width="200" height="400"></applet>
    ▪ NOTE: MAY NOT HAVE AUTHORITY TO EXECUTE APPLET IN COMPUTER LAB. (See INTERNET OPTIONS/Security Tab/Custom Level/ActiveX Controls & Plugins/Automatic Prompting for ActiveX Controls/Enable - & BE SURE IT ALLOWS YOU TO SAVE CHANGE (Check to see if modification was changed, i.e., that you have authority to change settings) - Fall 2004
  o <object> insert object </object>
    ▪ Description: identifies image, html text, sound video, or other external element for inclusion within the page
**Java**

- **Type**: container

- **Attributes**: including, but not limited to:
  - **classid** - identifies name/location of executable objects like applets and scripts
  - **data** - identifies name/location of data objects – images, sounds, or HTML text
  - **codetype** - optional/recommended; used with classid object
  - **type** - optional/recommended; used with data object
  - **standby** - message displayed while file is downloading
  - **declare** - retrieve large object before it is needed (not yet recognized by popular browsers)

- **Note**: Objects may be nested, allowing display options based on browser’s ability

- **Examples**:
  - Applet:
    ```html
    <object classid="UFO_Attack.class" codetype="application/java" width="200" height="400">Attack game cannot be found.</object>
    ```
  - Embedding other HTML documents:
    ```html
    <object data="abc.htm" type="text/html" height="60">
    ```

- **Note**: Browser must support ActiveX control.

- `<param />`

  - **Description**: identifies values needed by object at run time.
  - **Type**: empty

- **Attributes**: including, but not limited to:
  - name & value - assign a value to a program variable
  - valuetype - data, ref, or object -
  - codebase - coding shortcut to identify object paths

- **Sound Files** - including, but not limited to the following:
  - **µ-law (mu-law)**- less popular – UNIX (.au)
  - **RIFF WAVE** – Resource Interchange File Format - M.S. & IBM (.wav)
  - **AIFF and AIFC** – Apple (.aiff)
  - **MPEG Audio** - Moving Picture Experts Group; best algorithm; small (.mp2, .mp3)
  - **MIDI** – Musical Instrument Digital Interface (.mid, .midi)

- **Aural Style Sheets**

  - **Properties**: speak, volume, cue, pause, sound orientation
  - **Example**: `<style type="text/css" media="aural">`

  - **Not yet well-supported by current browsers**

- **Adding Video**

  - **MPEG Video** – most common; expensive to create (.mpg, .mpeg)
  - **QuickTime** – created for Apple; also QTFW for Windows (.qt, .mov)
  - **AVI** – Audio/Video Interleave; MicroSoft; VfW (.avi)
  - **Streaming Audio/Video** – movie or audio clip starts as soon as the browser plug-in receives a minimal buffering of the file.

- `<embed>`

  - **Non-HTML 4 specification**
  - **popular tag** – supported by Navigator 2.0 later & now IE 6.0

  - **Example**: `<embed src="abc.mp3" height="0" width="0" autostart="true" loop="true" />
  ```

  - **NOTE**: INFORMATION ON `<embed>` NOT INCLUDED IN TEXTBOOK.

- **Frames - Divide and Conquer** *(Chapter 10) (p.267)*

  - `<frameset>`/`frame setting`</frameset>

  - **Description**: describes how the browser window is subdivided into frames.

  - **Type**: container

  - **Attributes**: including, but not limited to the following:
    - rows and cols - either one required; best to define as percentages
      - **Example**: `<frameset cols="40%,*">` makes 2 columns, 40% and 60% of width of frame
      - **Example**: `<frameset rows="40%,30%,*">` makes 3 rows; 40%, 30% and 30% of height of frame
    - onload and onunload - used to activate a script or applet
      - **Example**: `<frameset rows="15%,*,*" onload="func1()">` activates a script called 'func1' when loaded
    - IE's ‘view source’ reveals the source of the main frame document defined in the `<frameset>` tag. To view the source of the document contained within the frame, right-click on the document within the frame, then select View Source.
    - Each page in the frame should have a way to link back to the main page
    - May be nested

  - `<frame />`  

    - **Description**: identifies the contents of a frame.

    - **Type**: empty

    - **Attributes**: including, but not limited to:
      - **src** - URL of displayable object that initially appears in the frame - important! (p.276)
      - **name** - name of the frame – optional, required if to be used as the target of links (p.280)
      - **scrolling** - restrict scrolling with `scrolling="no"`
• noresize - restrict the user from resizing with noresize="yes"
• id – not interchangeable with 'name'; not recommended. (p.279)
• marginheight and marginwidth - add spacing along edge of frame
• title – short; 2-3 words; recommended
• longdesc – longer description; non-visual browsers; recommended
  • example: <frame src="abc.html" longdesc="abcdesc.txt" />
• frameborder - suppress with frameborder="0" (1=on;0=off)
• <noframes> no frames</noframes> (p.282)
  • Description: provides content for browsers that do not recognize the <frameset> element.
  • Type: container.
  • Attributes: none.
  • Should always be coded to provide support for noncompliant browsers
• target attribute in the <a> anchor tag to direct the browser to load the document referenced in the href into the window or frame specified. (p.283)
  • Example: <a href="blues.htm" target="body">Blues</a>
    where 'body' represents the name attribute of a frame
• <base /> (p.285)
  • Description: specifies a document’s base URL explicitly. Relative path references then uses the base URL as a reference point. In other words, a default target.
  • Type: empty
  • Attributes: href & target
  • Placement: within the <head> container
  • if <base target="body" /> is coded in the <head> container, then each <a href="xxx.htm"> in the <body> container will default to target="body"
• Special Targets
  • Reserved names: _blank, _self, _parent, _top
• <iframe> inline frame</iframe>
  • Description: inserts a frame in the same line with text and other objects.
  • Type: container
  • Attributes: including, but not limited to:
    • src, height, width, name, align
• SUMMARY: Use frames only if you must
  • can cause navigation problems
  • compatibility problems with non-frame-compliant browsers
• Alternatives to Frames
  • Tables as frames
• Forms and Form Processing (Chapter 11)
• <form> contents of the form</form> (p.301)
  • Description: indicates that the contained data are part of a form.
  • Type: container
  • Attributes: including, but not limited to the following:
    • action - required;
      • method - required; "post" sends regular data to server; "get" for search engines (default)
  • Common Gateway Interface (CGI) example:
    • <form action="http://www.xcom/cgi-bin/x.cgi" method="post">
  • Electronic Mail Action example:
    • <form action="mailto:abc@abc.net" method="post">
• <input /> (p.307)
  • Description: creates an input element (control) within a form.
  • Type: empty tag.
  • Attributes: including, but not limited to:
    • name – variable name; required for 'text' type
    • type – type of control
      • type="text"
        • size - length of text-entry field; number of characters
        • maxlength - limits actual number of characters user can enter
        • value – assign a default value
      • type="password" - each character typed appears as special character (usually asterisks, "*")
      • type=file" - identify file to be sent back (p.314)
      • type=checkbox" - one or more may be selected (p.313)
        • checked - may specify default
      • type="radio" - only one may be selected
        • checked - may specify default
  • Local Action Controls
    • type="reset" - clears selections; resets all fields in the form (p.319)
    • type="submit" - encode & send information to cgi script or email identified in "action"
      • type="button" - trigger for JavaScript only (HTML 4.0) (p.321)
• Graphical Buttons
  o type="image" – increases load time; advise not to use (p.322)
• Hidden Data Fields (p.324)
  o type = “hidden” – neither the user nor the browser sees this field
    o Example: <input type="text" name="question1" size="15" maxlength="12" />

• <button>button text and/or images</button> (p.324)
  o disabled – testing purposes only (p.325)
  o tabindex – define tab order (sequence)
  o Intrinsic Events – trigger JavaScript methods

• <textarea>optional text string</textarea>
  o Description: creates an area for multi-line text input box.
  o Type: container.
  o Attributes: including, but not limited to:
    ▪ rows and cols - good practice to always use both
    ▪ wrap - good practice to always use (p.327)
      ▪ wrap="virtual" – text continuous except for user-entered CrLf’s
      ▪ wrap="physical" – recommended; wrap text with CrLf @ wrap
      ▪ wrap="off" – no wrap; text continuous except for user-entered CrLf’s

• <select>Set of option elements</select>
  o Description: creates a list box made up of the enclosed <option> elements.
  o Type: container.
  o Attributes: including, but not limited to:
    ▪ name
    ▪ size - # of choices shown in initial list box (p.330)
    ▪ multiple – allows multiple inputs

• <option>option text</option>
  o Description: defines one element of a menu list created with a <select> statement.
  o Type: container.
  o Placement: coded only within <select> element
  o Attributes: including, but not limited to:
    ▪ value – user-selected value of ‘name’ attribute from <select> element.
    ▪ selected – set as default

• <optgroup>collection of related options</optgroup> – not widely supported
• <fieldset> [<legend>]<input>[s]</fieldset>
  o Description: defines a group of related form controls
• <legend>Text</legend>
  o Description: provides a caption for the <fieldset> container
• <label>Text</label>
  o Description: provides a text label for the associated input control form

➤ CGI, Metatags, and Other Tricks of the Trade (Chapter 12)
• CGI – Common Gateway Interface – collection of programs the Web server uses to communicate with
  scripts that are kept on the server.
  o Common uses: processing forms, updating databases, searchable document support
  o Language: Perl, C, shell script or other
  o Platform: Unix
• Searchable Documents
  o A document that runs a search on your web site
  o Sensitive files could be accidentally exposed
• <isindex /> (p.354)
  o Description: indicates that the document contains a tool to perform searches.
  o Type: empty tag.
  o Note: Deprecated in favor of <input>
• <base /> (p.354)
  o Description: changes the way relative addresses are specified for all of the <a>, <img>, <link>, and
    <form> elements in the whole document. Reference a series of objects not on the same server as your page.
  o Type: empty tag.
  o Placement: must be inside the <head> container
  o Attributes:
    ▪ href – only required attribute
    ▪ target
• <meta /> (p.355)
  o Description: supplies additional information about the document.
Type: empty tag.

Attributes: including, but not limited to:
- `http-equiv` (p.356)
  - expires – set time/date after which a document has expired
  - content-type – load a specific character-set before loading page
  - content-language – ‘language-dialect’ selection for page
  - window-target – used to prevent document from appearing in a frame
  - pics-label – Platform for Internet Content Selection – for censorship
  - refresh – specify when a page is ‘refreshed’ or reloaded from the server
    - guidelines
      - give user enough time to read contents
      - use quick loading pages
      - give the user a way to escape

- `URL`
- `name` - description, keywords, author
  - others: Format, Language, Coverage, Source, Other contributor, Date, Resource identifier, Publisher, Relation, Rights management, scheme

- JavaScript Programs for Your Pages (Chapter 13) (p.373)
  - `<script>`/`<script>` (p.377)
    - Description: encloses the actual elements of the scripting language - JavaScript, Vbscript, Tcl, etc.
    - Type: container
    - Attributes: including, but not limited to:
      - `src` - required for external script
        - Example: `<script src="myscript.js">`
      - `type` - text/javascript, text/tcl, text/vbscript
        - Example: `<script src="netscript.js" type="text/javascript">`
      - `<meta />` - may be within `<script>` container
        - Example: the following will override meta tag in head container
          `<meta http-equiv="Content-Script-Type" content="text/javascript"/>
          defer. “true” tells the browser to wait for the code in the script to execute before loading the page
      - `language` - vbscript, JavaScript
        - deprecated, but recommended in addition to ‘type’ attribute
  - Basic Programming Structures (p.380)
    - Sequence
    - Selection (p.381)
    - Iteration (p.383)
  - Object – real life entities; data and code that are together as one
  - Object-Oriented Concepts in JavaScript (see Table 13.1, p.385)
    - The Date Object – `getDay`, `getHours`, `getMinutes`, `getSeconds`, ...
    - The Document Object – `lastModified`, `open`, `close`, `clear`, `write`, `writeln`, ...
    - The Window Object - `Status Bar`, `Dialog Boxes`, `New Windows`, `onLoad` and `onUnload` (p.391)
    - The Navigator Object - `appName`, `appVersion`, `platform`, `javaEnabled`, ...
  - Intrinsic Events - `onload`, `onunload`

- Dynamic HTML: Charismatic Pages (Chapter 14)
  - Image Maps – allow user to select different parts of an image (hot spots) to cause an event to happen.
    - Server-side vs Client-side
      - Server-side - increases Net traffic; it’s use is discouraged
      - Client-side – faster; easier to test; no interface with server
  - Image coordinates for Hot Spots
    - `<img>`
      - usemap – target name for client-side image map
      - `Example`: `<img src="new.gif" usemap="#new" border="0" />`
    - `<map>`/Contents of Image Map`</map>`
      - Description: encloses client-side image map data specified by the `usemap` attribute.
      - Type: container
      - Attributes: Including, but not limited to:
        - title – tool tip in newer browsers (N.N.6; O.6)
        - alt – older browsers’ tool tip
        - for compatibility, code both ‘title’ and ‘alt’
    - `<area />`
      - Description: defines the coordinates and link for one region of a client-side image map.
      - Type: empty tag
      - Attributes: Including, but not limited to:
        - alt - descriptive text – use is recommended
        - shape – used with `coords` to define hot spots
• circle, rect, poly, default – not recognized in I.E.
  • coords – required; defines boundaries of hot spot.
    • circle=”x,y,r”
    • poly=”x1,y1,x2,y2,x3,y3,…”
    • rect=”x1,y1,x2,y2”
  • href – required; (may have nohref coded)
  • nohref – defines a hot spot with no link

Example:
<area shape=”rect” coords=”317,7,366,231” alt=”Book” title=”Book” href=”book.html” />

XML – The Next Best Thing (Chapter 15)
• XML – eXtensible Markup Language
  • Extensibility – new elements can be added without changing the entire DTD
  • Portability across platforms
• DTD Models – sets of definitions for different individuals and groups
  • 5 most common: XML/EDI, OFX, GedML, dbXML, DocBooks
• XHTML – eXtensible Hypertext Markup Language
  • HTML with XML
• What’s Next?
  • HumanML – designed to represent human characteristics through XML
    • Examples of human characteristics: attitude, emotions, physical descriptors

Pragmatic Hypertext: It Ain’t All Pictures! (Chapter 16)
• Electronic Books (eBooks) – now available for many handhelds and PDA platforms; expected to be widely used across platforms in the future
• Web-Based Reference Manuals
• Online Help
• Wireless Web – mobile computing, wireless Web
  • WAP – Wireless Application Protocol
  • WML – Wireless Markup Language
• Aural Data – VoiceXML – next logical step to replace text messages on cell phones & handhelds

Instructional Methods:
• Lecture
• Lab Assignments
• Exams

II. Course Learning Outcomes

<table>
<thead>
<tr>
<th>Course Learning Outcome</th>
<th>Method of Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create, maintain, and troubleshoot web pages</td>
<td>Individual Web Site Design Project. At least 75% of students will score C or higher; grading is based on Website Design Project Specifications.</td>
</tr>
</tbody>
</table>

III. Required Text(s), Optional Text(s) and/or Materials to be Supplied by Student.


IV. Suggested Course Maximum - 20

V. List any specific spatial or physical requirements beyond a typical classroom required to teach the course.
• Computer for each student with Internet access and ability to install/use: text editor/HTML Validator(s) and various browsers

VI. Course Requirements/Grading System – Describe any course specific requirements such as research papers or reading assignments and the generalized grading format for the course

Exam(s)
Weekly Hands-on Lab Assignments
VII. Curriculum Checklist

☐ - Academic General Education Course (from ACGM – but not in WCJC Core)
   No additional documentation needed

☐ - Academic WCJC Core Course
   Attach the Core Curriculum Checklist, including the following:
   • Basic Intellectual Competencies
   • Perspectives
   • Exemplary Educational Objectives

☒ - WECM Courses
   If needed, revise the Program SCANS Matrix & Competencies Checklist.