Bay Area Video Coalition

HTML and CSS—Introduction

Classroom prep

1. MAMP
2. Aptana
3. Firefox, Chrome
4. Keynote
5. Install Firebug
6. Set PHP error reporting E_ALL | E_STRICT in /Applications/MAMP/bin/php/php5.3.6/conf/php.ini.

Course Outline—Day 1

I. Introductions
   - A. What does everyone know about HTML, CSS?
   - B. What does everyone hope to get from this class?
   - C. My background …

II. Getting started …

III. HyperText Markup Language
   - A. What is a text markup language?
     - 1. Plain text (emphasis all caps):
       This class is an Introduction to HTML and CSS.
     - 2. HTML (emphasis italics):
       This class is <em>HTML/CSS: Jumpstart</em>.
     - 3. RTF (emphasis italics):
       {\rtf1\ansi\deff0\deftab720{\fonttbl{\f0\fswiss MS Sans Serif;}{\f1\fdecor\fcharset2 Symbol;}{\f2\froman Times New Roman;}{\f3\froman Times New Roman;}}{\colortbl\red255\green0\blue0;}\deflang1033\pard\plain\f2\fs20 This class is \plain\f2\fs20\i HTML/CSS: Jumpstart \plain\f2\fs20 .\plain\f3\fs20 \par}
   - B. Physical vs. Semantic Markup
     - 1. Physical markup (pseudo-code):
       - a. Main heading:
         <indent left paper edge 0.5 inches><Times italic bold 16pt>HTML/CSS: Jumpstart <Times 10pt>
       - b. Good for printing, e.g. PostScript
     - 2. Semantic or logical markup (HTML)
       - a. Main heading:
         <h1> HTML/CSS: Jumpstart </h1>
       - b. Defines function or meaning of the text within the structure of the document
       - c. Good for multiple display methods
     - 3. What is HTML?
       - a. Based on Standard Generalized Markup Language
       - b. Designed for the Web and for display on different devices
       - c. Mainly a semantic markup language with a few physical tags
       - d. HTML also includes the ability to link from one document to another (hypertext), add images and other objects.
     - 4. HTML is not word processing or page layout
       - a. Rather than WYSIWYG, HTML documents consist of logical entities such as titles, paragraphs, lists, images, and other objects
       - b. Consequence is that each browser can display the objects in a way appropriate to its platform.
       - c. The logical structure is maintained whatever the means of display: graphical, text only, Braille, voice, etc.
     - 5. Specifying precise formatting
       - a. Precise physical formatting possible with HTML through CSS positioning
       - b. Cross-platform compatibility adversely affected
     - 6. Cascading Style Sheets—separate content from presentation
       - a. Allows physical characteristics to be assigned to logical elements
       - b. For CSS to work, the HTML must be correct

- c. Logical layout combined with CSS displays the document correctly whether on a CSS capable browser or not.
- d. Example:

```
<style type="text/css">
  H1 {
    font-family: Times, Times New Roman, serif;
    font-size: 16pt;
    font-weight: bold;
    margin-left: .5in;
  }
</style>
...
<h1>HTML and CSS</h1>
```

- C. Elements and tags
  - 1. Tags are the element name and attributes started and ended by the characters “<” and “>”.

```
<html lang="en"></html>
```

  - 2. The HTML language defines the set of elements and what attributes may be used for each.
  - 3. Container elements have a start and end tag:

```
<title>Introduction to HTML/CSS</title>
```

  - 4. Elements can contain other elements, character data, and comments—as defined.
  - 5. Empty tags consist of a single start tag:

```
<img src="imgs/logo.png" alt="360" height="90" alt="logo" />
```

  - 6. Tag names are lowercase.
  - 7. An element is the tag and its content taken together along with all its properties and methods.
  - 8. Types of elements
    - a. Flow
    - b. Phrasing
    - c. Metadata
- D. HTML is structured
  - 1. Element placement follows nesting rules: elements cannot overlap. Parent/child relationships apply.
  - 2. Some container tags exclude the use of other tags inside their element (the browser will ignore the illegal tag).
- E. Attributes
  - 1. name=”value”

```
<br clear="left" />
```

  - 2. affect tag function
  - 3. appear in the start tag
  - 4. values may be predefined or user-defined, numbers or strings, must be present and in quotes
  - 5. are all lower case in XHTML
- F. Selected core attributes
  - 1. `class`
  - 2. `id`
  - 3. `lang`
  - 4. `style`
  - 5. `tabindex`
  - 6. `title`
- G. Selected events
  - 1. `onclick`
  - 2. `onkeydown`
  - 3. `onkeypress`
  - 4. `onkeyup`

    - 5. `onmouseout`
    - 6. `onmouseover`

- IV. HTML elements by their function
  - A. Document type
    - 1. `<!DOCTYPE html>`
  - B. Root
    - 1. `<html>`
  - C. Document metadata
    - 1. `<head>`
    - 2. `<title>`
    - 3. `<meta />`
    - 4. `<link />`
    - 5. `<style>`
    - 6. `<base />` – base URL for links and anchors (href’s and src’s)
  - D. Scripting
    - 1. `<script>`
    - 2. `<noscript>`
  - E. Body elements in detail
    - 1. `<body>`
    - 2. `<section>`
    - 3. `<nav>` – navigation section
    - 4. `<article>`
    - 5. `<aside>`
    - 6. `<header>`
    - 7. `<footer>`
    - 8. `<hgroup>` – heading group
    - 9. `<h1>` – `<h6>` – headings
    - 10. `<address>` – contact information
  - F. Content grouping
    - 1. `<p>` – paragraph
    - 2. `<br />` – line break
    - 3. `<hr />` – horizontal rule
    - 4. `<pre>` – preformatted text
    - 5. `<blockquote>`
    - 6. Lists
      - a. `<ul>` – unordered list (bullet list)
      - b. `<ol>` – ordered list
      - c. `<li>` – list item
      - d. `<dl>` – definition list
      - e. `<dt>` – definition term
      - f. `<dd>` – definition description
    - 7. `<figure>`
    - 8. `<figcaption>` – figure caption
    - 9. `<div>` – generic flow container
  - G. Text-level semantics
    - 1. `<a>` – hyperlink
    - 2. `<em>` – emphatic stress
    - 3. `<strong>` – strong importance
    - 4. `<small>` – small print
    - 5. `<s>` – struck text
    - 6. `<cite>` – cited title of a work
    - 7. `<q>` – quoted text
    - 8. `<dfn>` – defining instance
    - 9. `<abbr>` – abbreviation
    - 10. `<time>` – date and/or time
    - 11. `<code>` – code fragment
    - 12. `<var>` – variable or placeholder text

13. `<samp>` – (sample) output
14. `<kbd>` – user input
15. `<sub>` – subscript
16. `<sup>` – superscript
17. `<i>` – offset text conventionally styled in italic
18. `<b>` – offset text conventionally styled in bold
19. `<u>` – offset text conventionally styled with an underline
20. `<mark>` – marked (highlighted) text
21. `<ruby>` – ruby annotation
22. `<rt>` – ruby text
23. `<rp>` – ruby parenthesis
24. `<bdi>` – BiDi isolate
25. `<bdo>` – BiDi override
26. `<span>` – generic span Images

H. Edits

1. `<ins>` – inserted text
2. `<del>` – deleted text

I. Embedded content

1. `<img />` – image
2. `<iframe>` – nested browsing context (inline frame)
3. `<embed />` – integration point for plugins
4. `<object>` – generic external content
5. `<param />` – initialization parameters for plugins
6. `<video>` – video
7. `<audio>` – audio stream
8. `<source />` – media source
9. `<track />` – supplementary media track
10. `<canvas>` – canvas for dynamic graphics
11. `<map>` – image-map definition
12. `<area />` – image-map hyperlink

J. Tables

1. `<table>` – table
2. `<caption>` – table title
3. `<colgroup>` – table column group
4. `<col />` – table column
5. `<tbody>` – table row group
6. `<thead>` – table heading group
7. `<tfoot>` – table footer row group
8. `<tr>` – table row
9. `<td>` – table cell
10. `<th>` – table header cell

K. Forms

1. `<form>` – user-submittable form
2. `<fieldset>` – set of related form controls
3. `<legend>` – title or explanatory caption
4. `<label>` – caption for a form control
5. `<input />` – input control
6. `<button>` – button
7. `<select>` – option-selection form control
8. `<datalist>` – predefined options for other controls
9. `<optgroup>` – group of options
10. `<option>` – option
11. `<textarea>` – text input area
12. `<keygen />` – key-pair generator/input control
13. `<output>` – result of a calculation in a form
14. `<progress>` – progress indicator

- 15. `<meter>` – scalar gauge
- L. Interactive elements
  - 1. `<details>` – control for additional on-demand information
  - 2. `<summary>` – summary, caption, or legend for a details control
  - 3. `<command />` – command
  - 4. `<menu>` – list of commands

Day 2

- V. Some CSS basics
  - A. how to style with CSS
    - 1. external style sheets
      - a. the link element
      - b. @import with the style element.
    - 2. the style element
    - 3. the style attribute—for the exception
  - B. comments in style sheets
- VI. CSS rules
  - A. selector—selects the affected element(s)
  - B. declaration—declares what to do to them
    - 1. property—predefined things that can be changed on an element
    - 2. value—how you want to change the element (not all values are predefined)
  - C. grouping declarations

```
h1 {
  color: green;
  text-align: center;
}
```

  - D. grouping selectors

```
h1, h2, h3 { … }
```

- VII. Common uses for CSS
  - A. fonts
    - 1. size (font-size)
    - 2. family (font-family)
    - 3. style (font-style)—normal, italic, etc.
    - 4. weight (font-weight)—normal, bold, etc.
    - 5. alignment (text-align)—left, center, right, etc.
    - 6. decoration (text-decoration)—none, underline, etc.
  - B. margins
    - 1. top, right, bottom, and left (margin-top, margin-right, etc.)
    - 2. using the em unit to specify sizes
      - a. It’s equal to the font height of the parent element.
      - b. Its use ensures the page is scalable.
      - c. It can be used to specify most sizes.
  - C. padding—it’s not the same as margins
  - D. links
    - 1. pseudo classes for the a element
      - a. :link
      - b. :visited
      - c. :hover
      - d. :active
  - E. browser default styles
    - 1. resource://gre/res/html.css
    - 2. user can override page styles
  - F. best practices
    - 1. Use relative size units (em, %, etc.) for browser-displayed pages.
    - 2. Use absolute units (pt, cm, etc.) for printing.
    - 3. Try to arrange content logically so it makes sense if styling is turned off.

    - 4. Start testing your pages early in different browsers on different platforms.
    - 5. Don’t use fonts which aren’t commonly available.
    - 6. Excepting very simple colors (e.g. black or white), use color numbers instead of names.
- VIII. Selectors (see Appendix D, Selector syntax)
  - A. type selectors (element_name)
    - 1. (X)HTML element tag names
    - 2. case-sensitive
  - B. class selectors[1]
    - 1. element attribute in HTML

      ```
      <p class="first_p">
      ```
    - 2. selector in style sheet

      ```
      .first_p
      ```
  - C. ID selectors
    - 1. must be unique to page
    - 2. attribute

      ```
      <div id="main_section">
      ```
    - 3. selector

      ```
      #main_section
      ```
  - D. combining selectors
    - 1. type and class
      - a. restrict the range of selection
      - b. example: `p.first`
  - E. contextual selectors
    - 1. descendant—two or more simple selectors together (h1 em)
- IX. CSS rule declaration values
  - A. length units
    - 1. relative (to font-size)— em ex
      - a. relative to the font size of the containing element
      - b. **except** when used as the value for a font-size property; it then refers to the font-size of its parent element.
    - 2. absolute— mm cm in pt pc
    - 3. pixel— px
  - B. percentages— %
  - C. keywords
- X. Box model
  - A. margins
    - 1. margins are always transparent
    - 2. negative lengths allowed
    - 3. collapsing margins—only if they touch (i.e. there is no borders or padding)
  - B. borders
    - 1. color
    - 2. style—none hidden dotted dashed solid double groove ridge inset outset
    - 3. width—thin medium thick <*length*>
  - C. padding
- XI. Visual formatting model
  - A. display
    - 1. block
    - 2. inline
    - 3. other: list-item | run-in | table
  - B. Normal flow
    - 1. Inline formatting
    - 2. Block formatting
  - C. Floats
    - 1. float—right or left
    - 2. clear—none, right, left, or both
  - D. Positioning

[1] Class and ID selectors are marked by the flag characters . and # respectively. They are case-sensitive and may only contain these characters: a to z, A to Z, 0 to 9, hyphen (-), or underscore (_). They must begin with a letter. They may not contain spaces.

- 1. position
  - a. static (default normal flow)
  - b. relative (normal flow)
  - c. absolute
  - d. fixed
- 2. box offsets: top, right, bottom, left
- E. z-index

- XII. More about the visual formatting model
  - A. width & height
  - B. min-width & min-height
  - C. width computation
    - 1. browser adjusts so that element’s total width = parent’s width (inherited width)
    - 2. border and padding never change, only margins and width
    - 3. images and objects have an intrinsic width, which is always used
  - D. line-height—use number instead of em or %.
  - E. vertical-align—raises inline elements above or below the text baseline.
    - 1. baseline—default
    - 2. sub—subscript
    - 3. super—superscript
    - 4. text-top— aligns top of element with top of the parent’s tallest letter
    - 5. middle—vertically centers element 0.5ex above baseline
    - 6. text-bottom—aligns bottom of element with bottom of the parent’s font
    - 7. top & bottom—aligns element with tallest or lowest thing in parent
      - a. avoid using together on same element
      - b. can lead to indeterminate layout
- XIII. Visual effects
  - A. overflow
    - 1. visible | hidden | scroll | auto
  - B. clip
    - 1. rect(<top>, <right>, <bottom>, <left>) | auto
  - C. visibility
    - 1. visible | hidden
- XIV. Lists
  - A. list-style
    - 1. type
    - 2. image
    - 3. position
- XV. Colors and backgrounds
  - A. color
  - B. background
    - 1. background-color
    - 2. background-image
    - 3. background-repeat
    - 4. background-position
    - 5. background-attachment
- XVI. Fonts
  - A. font-family
    - 1. Generic: serif, sans-serif, cursive, fantasy, monospace
  - B. font-style
  - C. font-variant
  - D. font-weight
  - E. font-size
  - F. font
- XVII.Text
  - A. text-indent
  - B. text-align
  - C. text-decoration
  - D. text-transform
  - E. white-space
    - 1. normal

- 2. pre
- 3. nowrap

XVIII. Tables

- A. border-collapse
    - 1. separate | collapse
- B. border-spacing—for separated borders
    - 1. <length> [<length>] (horizontal / vertical)
- C. empty-cells—for separated borders
    - 1. show | hide

XIX. User Interface

- A. cursor
    - 1. auto | crosshair | default | pointer
- B. outline
    - 1. outline-width
    - 2. outline-style
    - 3. outline-color

XX. Media

- A. Specifying media
    - 1. @import url(…) <*media type*>
    - 2. @media <*media type*>
- B. Media types
    - 1. all | braille | embossed | handheld | print | projection | screen | speech | tty | ty
- C. Paged media
    - 1. page boxes—@page
    - 2. page breaks
        - a. page-break-before
        - b. page-break-after
        - c. page-break-inside
    - 3. breaks inside elements: orphans and widows

XXI. CSS with JavaScript: dynamic HTML

- A. Popup boxes example