Chapter 2

How to code, test, and validate a web page

Web Design and Digital Development 1

Objectives

Applied

1. Use a text editor to create and edit HTML and CSS files.
2. Test an HTML document that's stored on your computer or a local server by loading it into a browser.
3. Validate an HTML document using a web site like W3C Markup Validation Service.

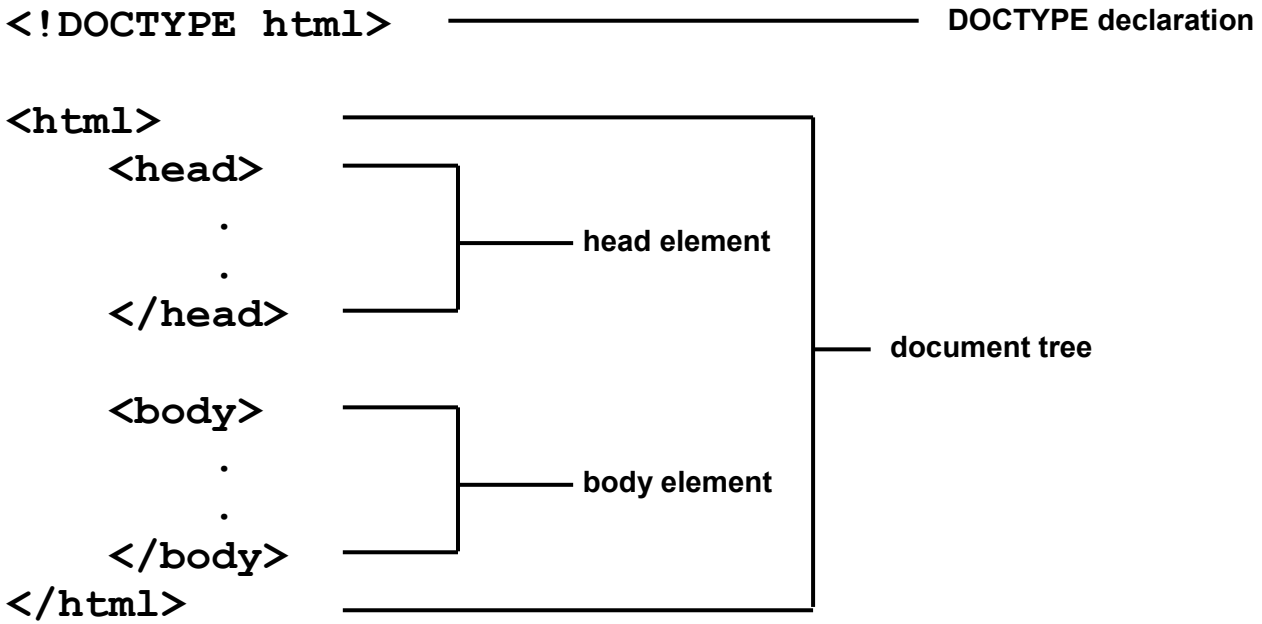
Knowledge

1. Describe the use of the head and body elements in an HTML document.
2. Describe these types of HTML tags: opening, closing, and empty.
3. Describe the use of attributes within HTML tags.
4. Describe the use of HTML comments and whitespace.

Objectives (cont.)

5. Describe the components of a CSS rule set.
6. Describe the use of these types of CSS selectors: type, id, and class.
7. Explain how and why you would start a new HTML or CSS file from a template.
8. Describe three ways to run a web page and one way to retest a page after you've changed the source code for the page.
9. Describe two benefits of validating HTML files.

The basic structure of an HTML5 document



Review the basic document structure and tag definitions located on page 43 in the textbook.

A simple HTML5 document

Note: No styles are applied.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>San Joaquin Valley Town Hall</title>
    </head>
    <body>
        <h1>San Joaquin Valley Town Hall</h1>
        <p>Welcome to San Joaquin Valley Town Hall.</p>
        <p>We have some amazing speakers in store for you
           this season!</p>
        <p><a href="speakers.html">Speaker
           information</a></p>
    </body>
</html>
```

Always Include a Meta tag

“Always Include the Character Encoding”
You should always include character encoding for your web pages, even if you never use any special characters. If you don't, your site becomes vulnerable to a cross site scripting attacks using UTF-7.

The attacker sees that your site has no character encoding defined, so it makes the browser think that the character encoding is UTF-7. Then the attacker injects UTF-7 encoded scripts into the web page, and your site is hacked.
The Character Encoding Should be the First Line of Your HTML After the Root and Head Elements**.**

Coding Recommendations

- Always code HTML statements in lower case.
- To be XHTML compliant:

  Always close tags

  End empty tags with a forward slash - /
- Don't use XHTML for new work

~Lecture Examples

- In your work drive/space
  - Create a new folder: c2_html
    - Place the **logo.gif** file, located on D2L, into it

~Lecture Examples cont.

- Select/create a new html file:
  - Save it in the c2_html (site root) folder
  - Save it as: c2_lecture_examples.html
    - Indent code for readability and maintainability
    - It should now be listed in the file/assets area.
  - Select the code view tab
- Open Chrome or Firefox
  - Open the file
- Add basic HTML (doctype, tags, etc)

~Lecture Examples cont.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>Title Goes Here</title>
    </head>
    <body>
    </body>
</html>
```

Coding simple HTML tags

Two elements with opening and closing tags

```
<h1>San Joaquin Valley Town Hall</h1>
<p>Here is a list of links:</p>
```

Two empty tags

```
<br>
<img src="logo.gif" alt="Murach Logo">
```


 and <img /> XHTML Compliant

Correct and incorrect nesting of tags

Correct nesting

```
<p>Order your copy <i>today!</i></p>
```

Incorrect nesting

```
<p>Order your copy <i>today!</p></i>
```

~2. Let's add some code to our page/file (E.g. page 45).

>

Review and Explain Code:

- Elements/Tags
  - Two sided: <tag_name >content</tag_name>
  - Empty: <hr> and <img>
- Content
  - Remember, HTML tags provide hints on content
- HTML and white space:
  - The browser ignores all white space
    - So, you can use white space to make your code more readable (I.e. maintainable)

Adding attributes to HTML tags

How to code an opening tag with attributes

An opening tag with one attribute

```
<a href="contact.html">
```

An opening tag with three attributes

```
<a href="contact.html" title="Click to Contact Us"
class="nav_link">
```

How to code an empty tag with attributes

```
<img src="logo.gif" alt="Murach Logo">
```

~Add and modify code in our example
(page 47 part 1) (~review)

How to code a Boolean attribute

```
<input type="checkbox" name="mailList" checked>
```

Attributes for identifying HTML elements

An opening tag with an id attribute

```
<div id="page">
```

An opening tag with a class attribute

```
<a href="contact.html" title=
   "Click to Contact Us" class="nav_link">
```

~Add and modify code to our example
(page 47 part 2) (~review)

Coding rules

- An attribute consists of the attribute name, an equals sign (=), and the value for the attribute.
- Attribute values don't have to be enclosed in quotes if they don't contain spaces.
- Attribute values must be enclosed in single or double quotes if they contain one or more spaces, but you can't mix the type of quotation mark used for a single value.
- Boolean attributes can be coded as just the attribute name. To code multiple attributes, separate each attribute with a space.

Our coding recommendation

- For consistency, enclose all attribute values in double quotes.
- Always validate the HTML for W3C compliance

A document with comments and whitespace

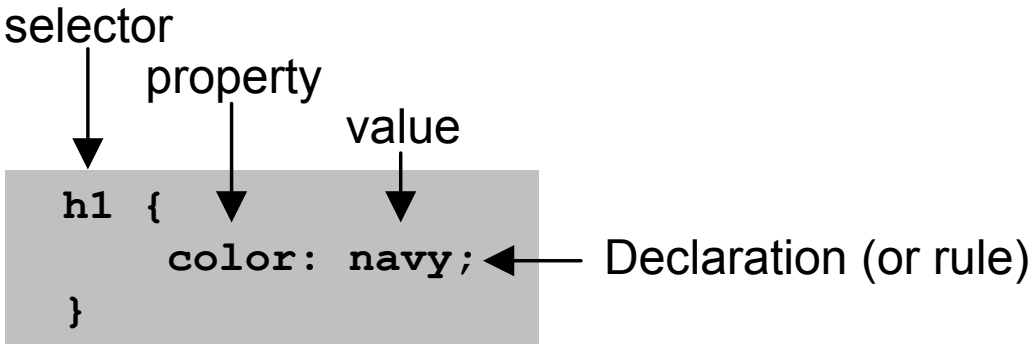
```
<!DOCTYPE html>
<!--
     This document displays the home page
     for the web site.
-->
<html>
     <head>
          <title>San Joaquin Valley Town Hall</title>
     </head>
     <body>
          <h1>San Joaquin Valley Town Hall</h1>
          <h2>Bringing cutting-edge speakers to the valley
          </h2>
     <!-- This comments out all of the unordered list
          <ul>
               <li>October 19, 2011: Jeffrey Toobin</li>
               <li>November 16, 2011: Andrew Ross Sorkin</li>
               ...
          </ul>
     The code after the end of this comment is active -->
```

~Adding comments and an un-order list to our example. (page 49) (~review)

- Professor's Note:
  - In this chapter we are taking a brief and overall look at general html components and tags.
  - In future lectures, we will look at each element in greater detail, definition, and application.

i.e. Cascading Style Sheet

The parts of a CSS rule set



For now, we are only concerned with syntax, not purpose.

CSS is used to structure your web page or; it can be used to modify any HTML element/tag's attributes and format content.

Note the { }

Styles can be applied to content by element/tag name selector.

A simple CSS document with comments

```
/*************************************************
* Description: Primary style sheet for valleytownhall.com
* Author:      Anne Boehm
*************************************************/
/* Adjust the styles for the body */
body {
    background-color: #FACD8A;    /* a shade of orange */
}

/* Adjust the styles for the headings */
h1 {
    color: #363636;
}
h2 {
    font-style: italic;
    border-bottom: 3px solid #EF9C00; /* bottom border */
}
/* Adjust the styles for the unordered list */
ul {
    list-style-type: square;   /* Change the bullets */
}
```

Note: Comment syntax, in a CSS file, is not the same as HTML comment syntax.

~Let's add an <h2> tag to our example (51 part 1 ~review – we'll apply some styles later)

Elements that can be selected by type, id, or class

```
<body>
    <div id="main">
        <h1 class="base_color">Student materials</h1>
        <p>Here are the links for the downloads:</p>
        <ul id="links">
            <li><a href=
                "exercises.html">Exercises</a></li>
            <li><a href=
                "solutions.html">Solutions</a></li>
        </ul>
       <p id="copyright" class=
          "base_color">Copyright 2012</p>
    </div>
</body>
```

~Let's add some html tags containing class and id identifiers to our example (51 part 2 ~review)

CSS rule sets that select by type, id, and class

Type

```
body {
    font-family: Arial, sans-serif;
}
```

ID

```
#main {
    width: 300px;
    padding: 1em;
}
#copyright {
    font-size: 75%;
    text-align: right;
}
```

Class

```
.base_color {
    color: blue;
}
```

Review the style selector types.

~Now, let's add and apply some styles to these new and different selectors. (51 part 3 ~review) (add one at a time)

Note:

- Textbook pages 54 – 70
  - Describe how to use create and view web pages and CSS files

Common coding errors

HTML

- An opening tag without a closing tag.
- Misspelled tag or attribute names.
- Quotation marks that aren't paired.
- Incorrect file references in link, img, or <a> elements.
- Case sensitivity for file names

HTML and CSS require an exact syntax. Close does not count.

Normally, when a tag is in error. The rest of the page applies formatting rules of the last correct style.

A syntax color coded IDE can help locate errors.

Common coding errors

CSS

- Braces that aren't paired correctly.
- Missing semicolons.
- Misspelled property names.
- Id or class names that don't match the names used in the HTML.

How to use the W3C Markup Validation Service

- Go to this URL:

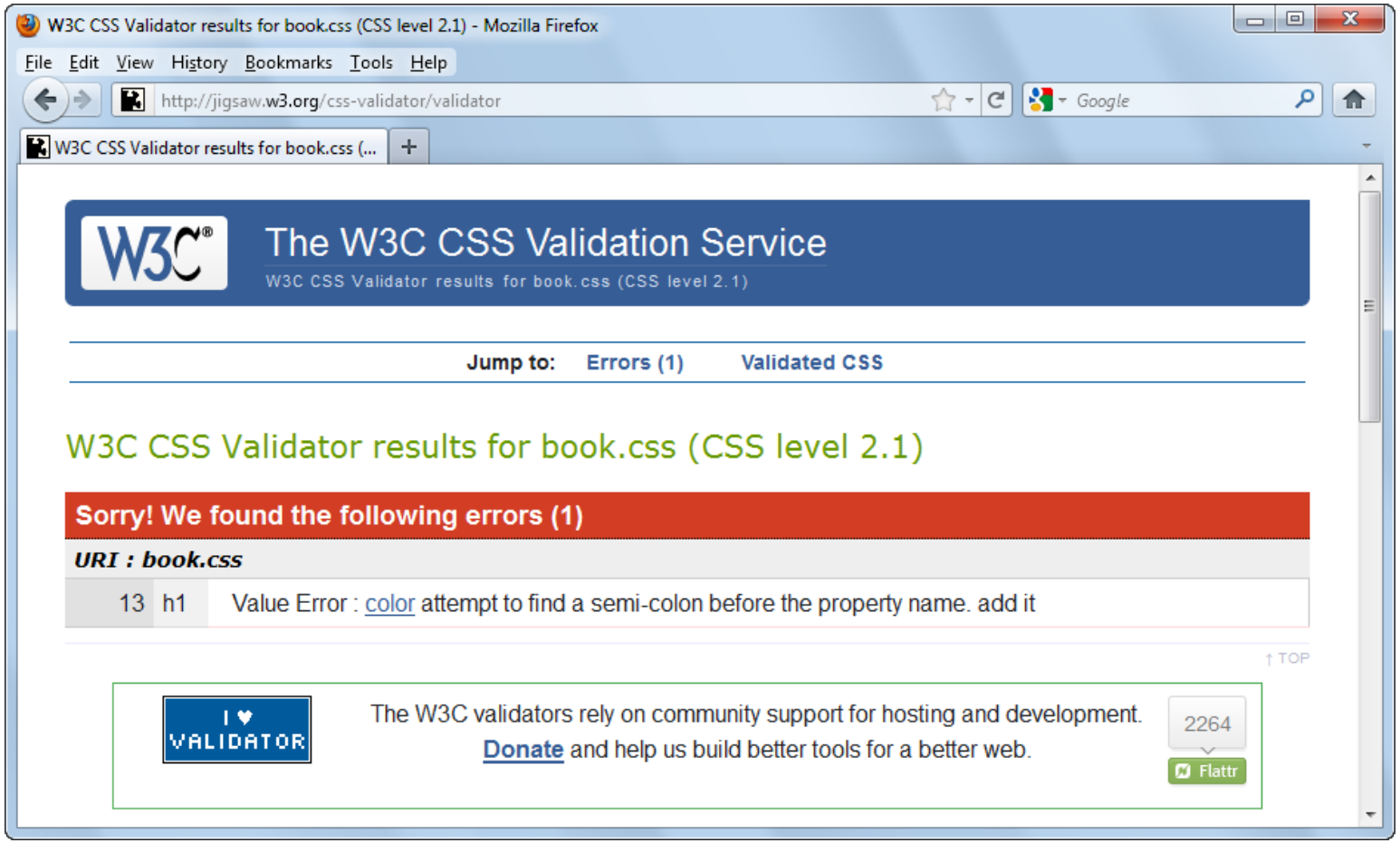
  `http://validator.w3.org/`

- Identify the file to be validated, and click the Check button.

~Let's validate our code.

http://validator.w3.org

The CSS Validation Service with errors displayed



How to use the W3C CSS Validation Service

- Go to the URL that follows, identify the file to be validated, and click the Check button:

  `http://jigsaw.w3.org/css-validator/`

~Optional: Create an external css and move the styles to it.

```
<link type="text/css" rel="stylesheet" href="chapter2_styles.css" />
```

End of Lecture

- Review chapter 2 assignment.
- Read chapter 3 for next week's lecture.
- Questions?