WEEK

2

COMP1223

Web Development Fundamentals

LECTURE NOTE

Book: Murach's HTML5 and CSS3

Week: 2

Chapters: 2 and 3

Author/s: Maziar Masoudi

External Sources:

- http://www.murach.com/books/htm5/index.htm
-

Table of Contents

Contents

Objectives 3

Syntax 4

- Basic struture of an HTML document 4
- How to code elements and tags 5
- How to code attributes 5
  - Empty tag with attributes 5
  - Boolean attribute 5
  - Common attributes to identify html elements 6
- HTML COMMENT and Withespace 6
- Coding Recommendations 6
- Validation 7
- Chapter 2 Summary 8

How to code the head section 9

- SEO guidelines for the title tag 9
- How to link to a favicon: 9
- Metadata 10
- SEO Guidelines 10
  - Summary 11

How to code text elements: 11

- Common Block elements for heading and paragraphs 11
  - SEO Guidelines 12
- Block elements for special types of text 12
- How to code inline elements 12
- How to code character entities: 13
- How to code the core attributes 13

Objectives

Knowledge

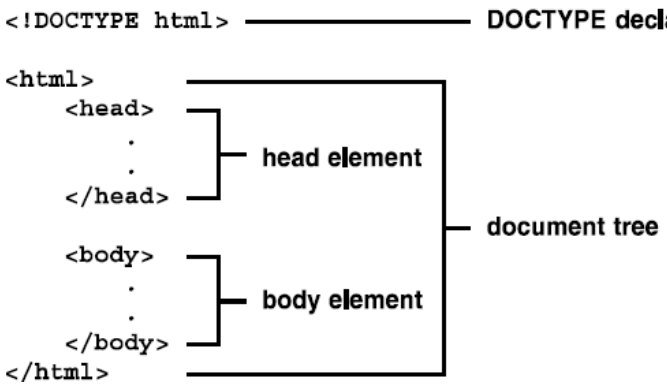
1. Describe the use of the head and body elements in an HTML document.
2. Describe these types of HTML tags: opening, closing, and empty.
3. Describe the use of attributes within HTML tags.
4. Describe the use of HTML comments and whitespace.
5. Describe three ways to run a web page and one way to retest a page after you've changed the source code for the page.
6. Describe two benefits of validating HTML files.
7. Describe the use of the title and meta elements in the head section of an HTML document.
8. Distinguish between a block element and an inline element.
9. Describe the use of these block elements: h1, h2, h3, and p.
10. Describe the use of these inline elements: br, i, b, sup, em, q, and strong.
11. Describe the use of character entities like   or ©.
12. Describe the use of these core attributes: id, class, and title.

Syntax

Every spoken language has a general set of rules for how words and sentences should be structured. These rules are collectively known as the language syntax. In **computer programming**, syntax serves the same purpose**, defining how declarations, functions, commands**, and other statements should be arranged. Read More: http://www.techterms.com/definition/syntax

📢 Code html5 in lowercase

Basic struture of an HTML document



📢 always code in lowercase (readablitiy)

- An *HTML document* contains *HTML elements* that define the content and structur of a web page.
- Each HTML5 document consists of two parts: the DOCTYPE declaration and the document tree.
- The *DOCTYPE declaration* shown above indicates that the document is going to use HTML5. You'll code this declaration at the start of every HTML document.
- The *document tree* starts with the html element, which marks the beginning and end of the HTML code. This element can be referred to as the *root element* of the document.
- The html element always contains one head element that provides information about the document and one body element that provides the structure and content of the document.

How to code elements and 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">
```

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>
```

Description

- Most HTML elements have an opening tag, content, and a closing tag. Each tag is coded within a set of brackets (<>).
- An element's *opening tag* includes the tag name. The *closing tag* includes the tag name preceded by a slash. And the *content* includes everything that appears between the opening and closing tags.
- Some HTML elements have no content. For example, the <br> element, which forces a line break, consists of just one tag. This type of tag is called an *empty tag*.
- HTML elements are commonly *nested*. To nest elements correctly, though, you must close an inner set of tags before closing the outer set of tags.

How to code 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">
```

Empty tag with attributes

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

Boolean attribute

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

Common attributes to identify 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">
```

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. They don't have to include the equals sign and a value that's the same as 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.

Description

- *Attributes* can be coded within opening or empty tags to supply optional values.
- A *Boolean attribute* represents either an on or off value.
- The id attribute is used to identify a single HTML element so its value can be used for just one HTML element.
- A class attribute with the same value can be used for more than one HTML element.

HTML COMMENT and Withespace

An HTML document with comments and whitespace

```
<!DOCTYPE html>
<!--
    This document displays the home page
    for the web site.
-->
```

Coding Recommendations

- Use whitespace to indent lines of code and make them easier to read.
- Don't overdo your use of whitespace, because it does add to the size of the file.

- An HTML *comment* is text that appears between the <!-- and --> characters. Since web browsers ignore comments, you can use them to describe or explain portions of your HTML code that might otherwise be confusing.
- You can also use comments to *comment out* elements that you don't want the browser to display. This can be useful when you're testing a web page.
- An HTML comment can be coded on a single line or it can span two or more lines.
- *Whitespace* consists of characters like tab characters, line return characters, and extra spaces.
- Since whitespace is ignored by browsers, you can use it to indent lines of code and separate elements from one another by putting them on separate lines. This is a good coding practice because it makes your code easier to read.

Validation

Should all HTML documents be validated, even though it's estimated that 99% of all web pages aren't? We say, yes! As we see it, validation is a useful practice that will solve some testing problems, and programs like Aptana and Dreamweaver make validation so easy that it's well worth doing. Besides that, validation may help your SEO results because clean code gets better results.

http://validator.w3.org

Exercise 2-2 , 2-3

Chapter 2 Summary

- An *HTML document* consists of a *DOCTYPE declaration* that indicates what version of HTML is being used and a *document tree* that contains the *HTML elements* that define the content and structure of a web page.
- The *root element* in a document tree is the html element, which always contains a head element and a body element. The head element provides information about the page, and the body element provides the structure and content for the page.
- Most HTML elements consist of an *opening tag* and a *closing tag* with *content* between these tags. When you *nest* elements with HTML, the inner set of tags must be closed before the outer set.
- *Attributes* can be coded in an opening tag to supply optional values. An attribute consists of the name of the attribute, an equal sign, and the attribute value. To code multiple attributes, you separate them with spaces.
- An *HTML comment* can be used to describe or explain a portion of code. Because comments are ignored, you can also use comments to *comment out* a portion of HTML code so it isn't rendered by the browser. This can be helpful when you're testing your HTML code.
- *Whitespace* consists of characters like tab characters, line return characters, and extra spaces that are ignored by browsers. As a result, you can use whitespace to indent and align your code.
- A *CSS rule set* consists of a selector and a declaration block. The *selector* identifies the HTML elements that are going to be formatted. Three of the common CSS selectors select by element (called a *type selector*), ID, and class.
- The *declaration block* in a CSS rule set contains one or more *declarations* that do the formatting. Each declaration (or *rule*) consists of a *property*, a colon, a *value*, and a semicolon.
- *CSS comments* work like HTML comments. However, CSS comments start with /* and end with */, and HTML comments start with <!– and end with –>.
- Aptana is a text editor that can be used to edit HTML or CSS code. To help you read the code, Aptana displays the syntax components with different colors. It also provides auto-completion lists, and error checking that detects common entry errors.
- When you start a new HTML or CSS file, it's best to start from a *template* or an old file that's similar to the new file that you're going to create.
- To *test* an HTML file, you run it on all of the browsers that your clients may use. Then, if you discover problems, you need to *debug* the code and test it again.
- To *validate* an HTML or CSS file, you can use a program or web site for that purpose. Often, a validation program will detect errors in a file, even though the web page displays the way you want it to on all browsers.

How to code the head section

```
<head>
    <title>San Joaquin Valley Town Hall | speakers and luncheons</title>
    <link rel="shortcut icon" href="favicon.ico">
</head>
```

SEO guidelines for the title tag

- Always code a title tag in the head section.
- The title should accurately describe the page's content, and it should include the one or two keywords that you want the page ranked for.
- The title should be interesting enough to entice the reader to click on it when it's shown in the search results for a search engine.
- The title should be unique for each page in your web site.
- Limit the length of your titles to around 65 characters because most search engines don't display more than that in their results.

- The title element specifies the text that's displayed in the browser's title bar.
- The title is also displayed in the tab for the web page, and it is used as the name of a favorite or bookmark for the page.
- A custom icon, called a *favicon*, is typically named favicon and must have the extension .ico to work correctly with Internet Explorer. A favicon typically appears to the left of the URL in the browser's address bar. It may also appear to the left of the title in a tab, and it may be used in a favorite or bookmark.
- To specify a favicon for a page, you use a link tag exactly like the one shown above.
- To create an ico file, you can use an icon editor, a program that converts an image to an ico file, or a web-based converter. You may also be able to find an icon on the Internet by searching for "web icons". For more information, see chapter 8.

How to link to a favicon:

```
<link rel="shortcut icon" href="favicon.ico">
```

favicon.ico Generator: http://www.favicon.cc/

Metadata

The Meta elements are used to specify metadata, which provides information about the content of the document.

Attribute	Description
charset	A required tag in HTML5 that specifies the type of character encoding to be used for the page. UTF-8 is the encoding that's commonly used for the World Wide Web.
name	Specifies the type of metadata being added to the document. The values "description" and "keywords" can be used to specify content that's used by some search engines.
content	Specifies the value to be used for the item specified by the name attribute.

```
<head>
      <meta charset="UTF-8">
      <meta name="description" content="Description of the document">
      <meta name="keywords" content="Related keywords">
      <meta name="author" content="Author name">
</head>
```

Refresh document every 30 seconds:

```
<head>
      <meta http-equiv="refresh" content="30">
</head>
```

SEO Guidelines

- Code the description metadata for each page of your web site. It should summarize the contents of the page, it should be unique for each page, and it can be longer than the title tag. When it is displayed in the search-engine results, it should encourage users to click on your link.
- Code the keywords metadata for each page of your web site. It should consist of no more than 10 keywords or phrases, and it should be unique for each page.

Summary

- The meta element provides information about the HTML document that's called *metadata*.
- The charset metadata is required for HTML5 validation.
- All or part of the description metadata may be displayed in the search results of some search engines.
- Although the keywords metadata has been de-emphasized by some search engines, it's still a good practice to include this data.

Links:

- Charset: http://www.w3schools.com/tags/att_meta_charset.asp
- Content: http://www.w3schools.com/tags/att_meta_content.asp
- http-equiv: http://www.w3schools.com/tags/att_meta_http_equiv.asp
- name: http://www.w3schools.com/tags/att_meta_name.asp
- Read More: http://www.w3schools.com/tags/tag_meta.asp
- Character encoding: http://www.iana.org/assignments/character-sets/character-sets.xhtml

How to code text elements:

Within the body of a document, you can code two types of element:

- Block elements:
  - Are the main building blocks of a web site and can contain other elements.
  - Each block element begins on a new line.
- Inline elements

Common Block elements for heading and paragraphs

Element	Description
`h1`	Creates a level-1 heading with content in bold at 200% of the base font size.
`h2`	Creates a level-2 heading with content in bold at 150% of the base font size.
`h3`	Creates a level-3 heading with content in bold at 117% of the base font size.
`h4`	Creates a level-4 heading with content in bold at 100% of the base font size.
`h5`	Creates a level-5 heading with content in bold at 83% of the base font size.
`h6`	Creates a level-6 heading with content in bold at 67% of the base font size.
`p`	Creates a paragraph of text at 100% of the base font size.

The based fount size and the spacing above and below headings and paragraphs are determined by the browsers, but you can change these values by using CSS.

SEO Guidelines

- Use the heading tags to show the structure and importance of the content on a page. Always start with an h1 tag and decrease one level at a time to show subsequent levels of importance.
- Don't use heading levels as a way to size text. Instead, use CSS to size the headings.

Block elements for special types of text

Element	Description
`pre`	Used for portions of code that are formatted with line breaks and spaces. Creates a block of preformatted text that preserves whitespace and is displayed in a monospaced font.
`blockquote`	Used for quotations. Can be used with the cite and <q> elements of figure 3-5.
`address`	Used for contact information for the developer or owner of a web site.

How to code inline elements

In contrast to a block element, an inline element doesn't starts on a new line. Instead, an inline element is coded within a block element.

Inline elements for formatting text

Element	Description
`i`	Displays the content in italics.
`b`	Displays the content in bold.
`sub`	Displays the content as a subscript.
`sup`	Displays the content as a superscript.
`br`	An empty element that starts a new line of text.

```
<p>This text contains <sup>superscript</sup> text.</p>
```

Definition and Usage

The <sup> tag defines superscript text. Superscript text appears half a character above the baseline. Superscript text can be used for footnotes, like WWW[1].

Tip: Use the <sub> tag to define subscript text.

Read More: http://www.w3schools.com/tags/tag_sup.asp

Inline elements for identifying content

Element	Description
abbr	Used for abbreviations.
cite	Used to indicate a bibliographic citation like a book title.
code	Used for computer code, which is displayed in a monospaced font.
dfn	Used for special terms that can be defined elsewhere (definitions).
em	Indicates that the content should be emphasized, which is displayed in italics.
kbd	Used for keyboard entries, which is displayed in a monospaced font.
q	Used for quotations, which are displayed within quotation marks.
samp	Used to mark a sequence of characters (sample) that has no other meaning.
strong	Indicates that the content should be strongly emphasized, which is displayed in bold.
var	Used for computer variables, which are displayed in a monospaced font.

Description

- An *inline element* is coded within a block element and doesn't begin on a new line.
- The formatting elements should be used when no special meaning is implied.
- The content elements should be used to convey meaning. Then, you can use CSS to format them.

How to code character entities:

Many of the web pages you develop will require special characters such as copyright symobol and opening and closing "curly" quotes. To display these special characters, you use **character entities**.

All character entities start with (**&**) and end with a semicolon (**;**)

&amp;	&	&lsquo;	‘ (opening single quote).
&lt;	<	&rsquo;	’ (closing single quote or apostrophe).
&gt;	>	&ldquo;	" (opening double quote).
&copy;	©	&rdquo;	" (closing double quote).

How to code the core attributes

HTML provides some core attributes that you can use with most elements.

- id: use the id attribute to uniquely identify an HTML element (CSS to format the element)
- class: similar to id attribute, except it doesn't have to be unique.
- title: specifies additional information about an element. (tootip)
- lang: identifies the language that the content of the element is written in. (<html lang="en">)

Attribute	Description
`id`	Specifies a unique identifier for an element that can be referred to by CSS.
`class`	Specifies one or more class names that can be referred to by CSS, and the same name can be used for more than one element. To code more than one class name, separate the class names with spaces.
`title`	Specifies additional information about an element. For some elements, the title appears in a tooltip when the user hovers the mouse over the element.
`lang`	Identifies the language that the content of the element is written in.

Accessibility guideline

- Always code the lang attribute on the html element to identify the language for the page.

Description

- The core attributes can be coded for most HTML elements.
- ID and class names are case sensitive, should start with a letter, and can include letters, numbers, underscores, hyphens, colons, and periods.
- The lang attribute is typically used to assist screen readers to read content correctly and to provide for searches that are restricted by language.

Read more on title: http://www.w3schools.com/tags/att_global_title.asp

Read more on lang: http://www.w3schools.com/tags/att_global_lang.asp