Unit 3 – Links and Text

Objectives	- Learn how to add hyperlinks to your web page. - Learn the differences between inline and block-level elements - Learn how to format text. - Learn the XHTML 1.0 specifications.
Required Reading	- Lesson 5 – Adding Links to Your Web Pages, Lemay, p. 95 - Lesson 6 – Formatting Text with HTML and CSS, Lemay, p. 127 - Chapter 7 - Web Typography, Sklar, p. 191 - Text Formatting with CSS (http://www.w3schools.com/css/css_text.asp)
Supplemental Reading	Optional: - W3Schools Link Information (http://www.w3schools.com/html/html_links.asp) - W3CSchools Formatting Text Information (http://www.w3schools.com/html/html_formatting.asp)
Assignment	Create a new page (mini_project1.html) which should contain some formatted text, links for other pages and email. Read the mini project one instruction for detailed instruction. **You will submit your mini_project1.html page after finishing unit 4.**

I. Important Concepts

1. Links

1.1 Linking to another web site:

To create a link to another web site, use the anchor tag (**<a href="">**), followed by the text you wish the user to click as the hyperlinked text, then end the anchor tag with </a>. Note that you only need the "/a" part of the tag as the ending tag.

Here is an example of the link to the University of Missouri's web site:

<a href="http://www.missouri.edu/**">**Mizzou Home Page**</a>**

The text between <a href> and </a> anchor tags, Mizzou Home Page, is what appears as the hyperlinked text to be clicked.

You can also include an image tag inside the <a href> tags. This lets you click an image that is hyperlinked to another web page. You will learn more about that in future lessons, along with how to add an image to a web page using image tags.

1.2 Opening linked pages in a new browser window:

To make a linked page/file open in a new browser window, use the **target attribute** and set the value to "**_blank**" in the anchor tag. For example, to link to the University of Missouri's web site in a new browser window:

<a href="http://www.missouri.edu/" target="_blank">Mizzou Home Page</a>

1.3 Linking to another HTML file in your directory:

To make a second page for your web site that you will link to, simply open a new file in your text editor, use the basic tags as you did for the assignments.html file and save this new file as mini_project1.html.

Absolute path: You'll notice that when you link to another person's web site, you include the entire pathname in the <a href> tag. This is called an **absolute path**. For example, the link to “http://www.missouri.edu/~yourpawprint/somepage.html” (without quotes) would link to the “somepage.html” file, that locates in the WWW directory of your bengal space.

Relative path: When you've created a second page and want to link to it, you can use what's called a **relative path**. When you create a relative path, you specify the path to the new linked file RELATIVE to the location of the file you're linking from (the file you're adding the <a href > tag to). For example, the link to “somepage.html” (without quotes) would link to the “somepage.html” file, as long as **it was in the same directory as the file that this code was in**.

You can use the absolute pathname, but it's more efficient for the server to use the relative one because it doesn't need to look up the domain name and root (or top) directory again. If you create subdirectories and begin moving files around in them, you will encounter fewer problems with broken links by sticking to relative paths.

1.4 Linking to a specific point in your HTML page:

With links and anchors, you can create internal hyperlinks within a page which, when clicked, will jump to specific parts of the page. These hyperlinks tell the browser to show the anchored portion of the document at the top of window when the link is clicked. This allows you to set up a table of contents for a long page that might otherwise require a lot of scrolling. Sometimes it makes sense to keep things together on a longer page and this method can be a good solution in that situation.

To create a link to a section within a page that's named curriculum.html, you could use something like this:

- First, in the link that can lead you to the specific part of the page, you need to use the tag and attribute: <a href="curriculum.html#science">Science Section</a>

- Then, anchor the section (the specific part of that page) that you want to jump to by using this tag and attribute: <a name="science">Science Resources</a>

Using a top anchor for a long page: It is good design practice on a long page to include a "top" link after each section you jump to: <a href="curriculum.html#top">Return to top</a>. You will have just one anchor tag named "#top" that you will place near the top of the page (i.e. <a name="top">The first part of your page</a>). It can be the first tag within your body tag.

Jumping to a specific point in a second page: For example, let's say that I have a page called general.html on which I describe general curriculum information. I have a second page called curriculum.html that gets into specifics on various subjects. I might set up a link on general.html that leads to science information appearing near the bottom of curriculum.html. To do that, I would put: <a href="curriculum.html#science">Science Section</a> on the general.html page.

On curriculum.html I would include the anchor tag at the spot I want my viewer to end up on:
<a name="science">Science Resources</a>

1.5 Adding a link for creating an email message:

Another use of the anchor tag is adding a link which, if your viewer's browser mail preferences are set, will open a separate window. Your viewer can then compose and send an email. The following tag allows this:

<a href="mailto:zone@missouri.edu">E-mail the Zone</a> (you can substitute your e-mail address and text.)

Moving the mouse over "E-mail the Zone" causes the e-mail address to appear in the status bar at the bottom of the browser window. It can be better to repeat the e-mail address as the hypertext link. This can be helpful for those using a text-only browser or if someone prints the page. Printing a page with the first tag would show only "E-mail the Zone". An alternative way to create a mailto link is to use the following example, which would show "E-mail zone@missouri.edu" if it were printed.

<a href="mailto:zone@missouri.edu">E-mail zone@missouri.edu </a>

2. Inline vs. Block-Level Elements

2.1 What does inline or block-level mean:

Most html elements permitted within the body tags are classified as either inline or block-level elements. Generally, inline elements may **only contain text and other inline elements**. When rendered visually in a web browser, inline elements do not usually begin on a new line. Block-level elements typically **contain inline elements and other block-level elements**. When rendered visually, block-level elements usually begin on a new line.

2.2 Rules for using inline and block-level elements:

It is **not allowed** to have **block-level elements within inline elements**. Let’s look at an example below. The **font tag** is classified as an inline element; the **p tag** is classified as a block-level element. If you have three paragraphs on your page and you want to set the font to Arial for all the paragraphs, you might try the following code and thought the font size of the text in these paragraphs would be changed to Arial.

```
<font face=”Arial”>
  <p>…………….</p>
   <p>…………….</p>
  <p>…………….</p>
</font>
```

However, according to the concept explained above regarding inline and block-level elements, it is NOT permitted to write the html code this way. You will get an error message when you try to validate this page using xhtml specification because it violated the rules (you will learn more about validation in unit 5).

So, how to correct the above example? You will need to use a font tag each time when you want to change the font setting in each paragraph (or within a block-level element).

```
<p><font face=”Arial”>…………….</font></p>
<p><font face=”Arial”>…………….</font></p>
<p><font face=”Arial”>…………….</font></p>
```

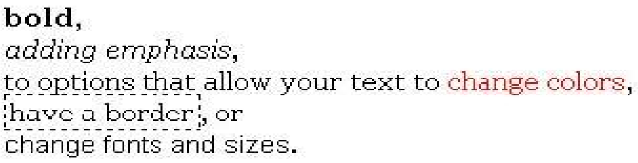
2.3 Resources for understanding more about inline and block-level elements:

Here are two websites listing the html inline and block-level elements:

http://www.htmlhelp.com/reference/html40/block.html

http://www.htmlhelp.com/reference/html40/inline.html

3. Formatting Text and CSS



bold,
adding emphasis,
to options that allow your text to change colors,
have a border, or
change fonts and sizes.

Here are some commonly used attributes for the font tag,

```
<font size=”…”>xxx</font>
<font face=”…”>xxx</font>
<font color=”…”>xxx</font>
```

However, the font tag is deprecated in the latest versions of HTML (HTML 4 and XHTML). The World Wide Web Consortium (W3C) has removed the <font> tag from its recommendations. In future versions of HTML, style sheets (CSS) will be used to define the layout and display properties of HTML elements.

3.1 Formatting text using CSS:

Based on the rule about inline vs. block-level elements we have discussed above, you may be thinking…how many font tags you need to have if you are creating a sophisticated page with a variety colors and text formatting?

A more efficient way to format text is to use CSS. CSS can get very complicated. There are three ways to write CSS rules**: inline, embedded, and external CSS**. In this unit, only the inline style will be introduced. For this unit, I encourage you to use inline style CSS (i.e. the style attribute, <p style="….">), as discussed on page 151-153. While we will not require you to use CSS, it is best that you learn it and get in the habit of using it, as CSS is the way of the future.

Inline style CSS: Adding a style attribute to an html tag is called inline style. You can see examples in Lesson 6 (format the text decoration and font properties by using CSS on pages 133-135 and 151-153 of the textbook). The style attribute can be added in both the inline and block-level elements.

- **If you want to change some text within a paragraph**, you can use

  <span style="….">xxxx**<span>** to do character formatting.

- If you want the entire block of text to have the same font setting, you can use

  <**p style="…">**xxxx**</p>** to format the paragraph.

Here is a **generic CSS rule** for inline style.

<tag style="property: value">…..</tag>

You can also combine CSS rules for one tag by adding a semicolon at the end of the first rule and typing the second rule starting with a property name. For example, **<p style="font-size: 14px; font-family: Arial">….</p>**

More information regarding embedded CSS and advanced techniques will be introduced in Unit 10.

4. A checklist for you to self-check your pages

In order to make your pages xhtml compliant, here are some rules listed in the following table to help you self-check your pages. You are encouraged to check your pages against this list. The correctness of coding by complying XHTML specifications is the key to make sure your page would be rendered the same across different operating systems and web browsers.

Major XHTML Specifications	Check the Following Things
The DOCTYPE declaration is required. (We will discuss more about DOCTYPE in unit 5)	Do you have a DOCTYPE declaration which should be something similar to the following example as the first thing on your page? <!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
The document needs to be well-formed: all the html, head, title, and body tags are required.	Do you have all the structural tags on your page? <html> <head> <title></title> </head> <body> </body> </html>
Tags and attributes must be in lowercase.	Do you have all the tag names and attribute names in lowercase?

All elements must be closed. For empty tags, such as the hr, br, and img tags, adding an extra space and a slash at the end of the tag: <br />	Does every tag have it corresponding closing tag? If you have 10 opening p tags, then you must have 10 closing p tags as well. If it is a self-close tag, do you close it by adding an extra space and a slash at the end of the tag?
All tags must be properly nested: the last tag opened should be the first tag closed: <p><a>…….</a></p>	Do you place all the tags in the appropriate order?
Attribute values must be quoted: <font size="5">	Do you have quotation marks around every value for your attributes?