Introduction

Websites can be created by using one of many programming languages (e.g., HTML, JSP, PHP, ASP, ASP.NET, or Perl). Among those languages, HTML is the most basic text-based language that has been used in web design since 1989. HTML consists of two parts: 1) content that will be displayed in a web browser, and 2) markup or tags, which are encoded information that are generally hidden from web page viewers. This three-part handout will help users create a basic website using fundamental HTML knowledge that they can build on with more advanced techniques. The first part covers web design basics, including the use of divisions to arrange the page layout, menu bar to link multiple pages, and CSS3 to enhance web page elements.

Downloading the Data Files

This handout includes sample data files that can be used for hands-on practice. The data files are stored in a self-extracting archive. The archive must be downloaded and executed in order to extract the data files.

- The data files used with this handout are available for download at http://www.calstatela.edu/its/training/datafiles/html5p1.exe.
- Instructions on how to download and extract the data files are available at http://www.calstatela.edu/its/docs/download.php.

Requirements

- A text editor, preferably Notepad++ or something similar. Notepad that is provided with Windows will also work.
  - Notepad++
    - http://notepad-plus-plus.org/download
    - http://portableapps.com/apps/development/notepadpp_portable (Portable version that can be placed on a flash drive)
- A current web browser (e.g., Mozilla Firefox, Google Chrome, or Internet Explorer 9).
  - Mozilla Firefox http://www.mozilla.com/en-US/firefox/new/
  - Google Chrome http://www.google.com/chrome/intl/en/landing_chrome.html?hl=en
  - Internet Explorer 9 (Only on Windows 7 and Windows Vista Service Pack 2) http://windows.microsoft.com/en-US/internet-explorer/downloads/ie

    NOTE: Anything below Internet Explorer 9 is not compatible with the new HTML5 features. While any on the list above is acceptable, this handout was developed based on the steps administered in Notepad++ and Mozilla Firefox.

About Notepad++

Notepad++ is a lightweight, but powerful text editing tool (see Figure 1). It is built primarily for programming, not writing essays. It has built-in tag highlighting and automatic indentation which helps improve organization and readability.

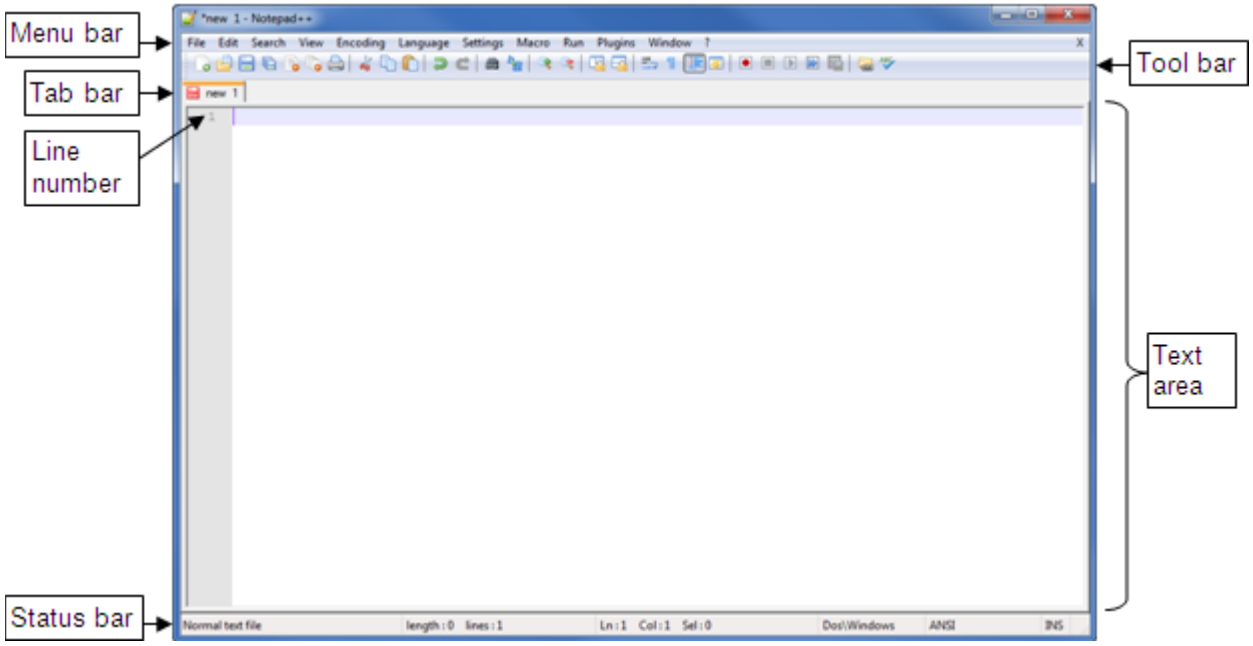

Figure 1 – Notepad++ Program Window

About HTML and CSS

HyperText Markup Language (HTML) and Cascading Style Sheets (CSS) are two of the core technologies for building web pages. HTML is the language for describing the structure of web pages. CSS is the language for describing the presentation of web pages, including colors, layout, and fonts. The separation of HTML from CSS makes it easier to maintain websites and share style sheets across web pages.

HTML5

HTML5 is still a work in progress and has been since June of 2004. Most of the tags that were compatible in previous versions of HTML are still compatible. However, they are no longer used because better and cleaner methods have since been found. HTML5 is still not final, but is already making a huge impact on the web.

Basic HTML Tags

HTML uses *tags* to describe the structure of web pages. Web browsers use the tags in an HTML document to interpret the content and display it as a web page. For every HTML based web page, there are several tags that are always inserted into the document: <!doctype>, <html>, <head>, <title>, and <body> (see Figure 2).

```
1  <!doctype html>
2  <html lang="en">
3      <head>
4          <title>Title of the Page(Change me!)</title>
5          <meta charset="utf-8">
6      </head>
7      <body>
8      </body>
9  </html>
```

Figure 2 – Source Code for a Blank HTML Page

HTML Tag Syntax

HTML tags are keywords surrounded by angle brackets (< and >). HTML tags normally come in pairs (e.g., <body> and </body>). The first tag in a pair is called the *opening tag*, and the second tag is called the *closing tag*. The opening tag consists of the tag name enclosed in angle brackets (e.g., <body>). The closing tag is the same as the opening tag except it has a forward slash (/) before the tag name (e.g., </body>).

Head and Body Tags

Web pages are divided into two sections: the head and the body. The *head* section is where you define the title of the web page, provide metadata about the page (e.g., keywords, character encoding), add style sheets, and insert scripts. The *body* section encloses the contents of the web page, the part that your visitors will see, including text, hyperlinks, images, tables, lists, etc.

CSS

CSS is used to control the style and layout of web pages. CSS can be added to an HTML document in the following three ways:

- An *external style sheet* (CSS file) is ideal when the style is applied to many web pages. It enables you to change the appearance and layout of an entire website just by editing one file. Each page must link to the style sheet using the <link> tag in the head section.
- An *internal style sheet* can be used if one single document has a unique style. Internal styles are defined in the head section using the <style> tag.
- An *inline style* can be used if a unique style is to be applied to one single occurrence of an element. To use inline styles, use the *style* attribute in the relevant tag.

CSS Syntax

A CSS rule is made up of two parts: a selector and one or more declarations. The selector designates or targets the HTML element you want to style. The declaration instructs the web browser to display all elements identified by the selector in a particular way. Each declaration consists of a property (the style attribute you want to change) and a value, and always ends with a semicolon. Declarations are surrounded by curly brackets. The rule in Figure 3 takes all paragraphs in the HTML document and displays them centered and in the color red.

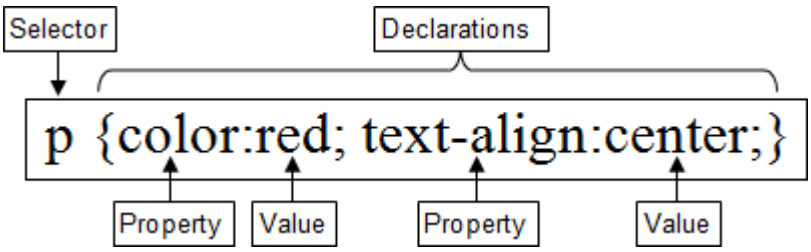

Figure 3 – CSS Syntax

ID and Class Selectors

In addition to setting a style for an HTML element, CSS allows you to specify your own selectors called ID and class. The *ID* selector is used to specify a style for a single, unique element. It uses the *id* attribute of the HTML element, and is defined with a number sign (#). The *class* selector is used to specify a style for a group of elements. Unlike the ID selector, the class selector is most often used on several elements. This allows you to set a particular style for many HTML elements with the same class. The class selector uses the HTML *class* attribute, and is defined with a period.

CSS Selectors

CSS selectors offer various ways to select your HTML tags for styling (see Table 1).

NOTE: For more information about CSS selectors, visit http://www.w3.org/TR/CSS2/selector.html.

Table 1 – CSS Selectors

Selector	Syntax	Description	Example
ID Selector	#box{property:value;}	Will affect any tag with the ID *box* applied to it.	 This text is affected.
Class Selector	.box{property:value;}	Will affect any tag with the class *box* applied to it.	 This text is affected.
Tag Selector	p{property:value;}	Will affect all tags in the HTML document.	 This text is affected.
Child Selector	p>a{property:value;}	Will affect all <a> tags where the parent is a tag.	 <a> This text is affected.
Pseudo Selector	a:hover{property:value;}	Will only apply the defined style on mouse over.	<a>Hover over this text.

Creating a New Web Page

Every website is built inside a directory on a web server, and each web page is a separate file in that directory. When a visitor enters a Uniform Resource Locator (URL) in a web browser's Address bar without a file name at the end (e.g., http://www.calstatela.edu), the server looks for a default file and displays that automatically. On most web servers, the default page in a directory is named *index.html*.

Creating a Home Page

When you start building your website, you should first create your home page and name it *index.html*.

To create a home page:

1. Launch **Notepad++**. In the ITS Training Program computer labs, click the **Start** button, point to **All Programs**, point to **Notepad++**, and select **Notepad++**.
2. If the text area is not blank, click the **New** button  on the **Tool** bar.
3. Click the **File** menu and select **Save As**. The **Save As** dialog box opens.
4. In the **File name** box, type **index.html**.
5. Click the **Save as type** arrow and select **All types (*.*)** from the list.
6. Click the **Desktop** button in the left pane of the **Save As** dialog box.
7. Click the **Create New Folder** button  , rename the new folder to **Website**, and then press the **Enter** key.
8. Double-click the newly created folder, and then click the **Save** button.

Basic Web Page Structure

This section covers the most basic elements you need to create the structure of your HTML5 document. The first thing you need to add in your document is a *doctype* declaration which lets web browsers know what version of the markup language the page is written in. Next, you need to add the <html> tag which tells web browsers that this is an HTML document. You also need to add the <head> tag which is a container for all the head elements, and the <body> tag which contains all the contents of an HTML document. In addition, you can define the character encoding that is used in the document (e.g., utf-8).

To create a basic web page structure:

1. In **Notepad++**, enter the following code in the text area.

```
<!doctype html>
<html lang="en">
    <head>
        <title></title>
        <meta charset="utf-8">
    </head>
    <body>

    </body>
</html>
```

NOTE: Take note of the indentation.

Indentation

Indenting the source code of any programming language is absolutely crucial. It makes the code cleaner and much easier to read. The Tab key is most often used to indent source code. As a rule of thumb, you should indent every time you insert an element into another element; this is known as nesting.

Example:

```
<body>
    <p>
        This is a nested paragraph.
    </p>
</body>
```

In the example above, the <p> tag is nested in the <body> tag. Every tag in an HTML document is nested within the <html> tag, except for the <!doctype> which must be placed at the top of the HTML document, before the <html> tag.

Adding a Page Title

The <title> tag defines the title of the page and is required in all HTML documents. The title appears in the Title bar of web browsers. It also provides a title for the page when it is added to favorites, and displays a title for the page in search engine results.

To add a page title:

1. Place the cursor between the opening **<title>** tag and the closing **</title>** tag, and type **HTML Workshop**.

Saving and Viewing the Web Page

Saving the page right away serves many purposes. First, it is a good habit to save your work periodically so that nothing is lost in the event of a system error. Second, it gives a name to the file that is being worked on so that it can be referenced from other files. Third and most importantly from a web design standpoint, it allows links from one page to another to be relative (i.e., instead of *http://www.mywebsite.com/webpage_2.html*, links will look like *webpage_2.html*). This helps make it easier to seamlessly work on a website offline as well as online, as links are referencing files in the same folder or subfolders.

To save and view the web page:

1. Click the **File** menu and select **Save**. Or, press **Ctrl+S**.
2. Double-click the **index.html** file located in the **Website** folder to view the web page in a web browser window.

Adding an Internal Style Sheet

In this lesson, the <style> tag is added to the head section of the document which will later be used to add styles.

To add the <style> tag:

1. Add a new line below the **<meta>** tag.
2. Type **<style>**, press the **Enter** key four times to create several blank lines, and then type **</style>** (see Figure 4).

   NOTE: Press the **Tab** key to apply the appropriate indentation.

```
<!doctype html>
<html lang="en">
    <head>
        <title>HTML Workshop</title>
        <meta charset="utf-8">
        <style>

            CSS goes here!

        </style>
    </head>
    <body>

    </body>
</html>
```

Figure 4 – <style> Tag Added to the Head Section

Setting the Background Color

The background of a page sets the undertone for the entire website. When setting a background, it is important to have something simple. The background should enhance the page content, not overpower it. Two types of backgrounds can be used: a single solid color or an image. In this lesson, a background color is set using CSS.

To set a background color:

1. Place the cursor on the first blank line between the opening **<style>** tag and the closing **</style>** tag, and then type the following CSS rule to change the background color to light grey.

```
body{
    background-color:#CCCCCC;
}
```

NOTE: Be sure to apply the appropriate indentation; line up the closing curly bracket with the **body** selector (see Figure 5).

```
<style>
    body{
        background-color:#CCCCCC;
    }
</style>
```

Figure 5 – CSS Rule for Setting the Background Color

2. To see the result, save the file, and then refresh the page in the browser window.

Creating a Wrapper

Creating a wrapper is an essential part of creating any web page. The wrapper helps contain all of your content and gives a more reliable reference for positioning elements on your page.

To create a wrapper:

1. Place the cursor on the blank line between the opening **<body>** tag and the closing **</body>** tag, and then type the following code.

```
<div class="wrapper">


</div>
```

NOTE: Be sure to apply the appropriate indentation. Also, create several blank lines between the opening and closing tags for content.

Creating a Class for the Wrapper

A *div* element with a *wrapper* class does not do anything if the *wrapper* class is not defined.

To define a new class:

1. Add a new line before the closing **</style>** tag, and then type the following CSS rule.

```
div.wrapper{
    width:1000px;
    height:800px;
    margin-left:auto;
    margin-right:auto;
    border:1px solid black;
}
```

NOTE: The **width** property sets the width of the **div** element to 1,000 pixels; **height** sets the height of the element to 800 pixels; **margin-left:auto;** and **margin-right:auto;** are used to center the element; and **border** creates a border around the element that is 1 pixel wide, solid (as opposed to dotted or dashed) and black.

NOTE: Using **margin-left:auto;** and **margin-right:auto;** is currently the most effective and compatible way to center elements on a web page.

2. To see the result, save the file, and then refresh the page in the browser window.

Checkpoint One

This is the first of three checkpoints within this handout. Make sure that the document that is being created is similar to the *checkpoint_1.html* file located in the data files folder (see Figure 6).

```
<!doctype html>
<html lang="en">
    <head>
        <title>HTML Workshop</title>
        <meta charset="utf-8">
        <style>
            body{
                background-color:#CCCCCC;
            }
            div.wrapper{
                width:1000px;
                height:800px;
                margin-left:auto;
                margin-right:auto;
                border:1px solid black;
            }
        </style>
    </head>
    <body>
        <div class="wrapper">

        </div>
    </body>
</html>
```

Figure 6 – Source Code in the checkpoint_1.html File

Creating the Web Page Structure within the Container

Up to this point, the web page that is being created has a colored background and the outline of a box, which is the wrapper or container. In this section, a banner and a navigation bar will be added to the page.

Adding a Banner

A *div* element can be placed at the top of the web page to serve as a container for text or an image that represents the website.

To add a banner:

1. Place the cursor on the blank line between the opening ** ** tag and the closing ** ** tag, and then type the following code.

   ** **

2. To define a new class for the banner, add a new line before the closing **</style>** tag, and then type the following CSS rule.

```
div.banner{
    width:750px;
    height:90px;
    position:relative;
    top:0px;
    margin-left:auto;
    margin-right:auto;
    border:1px solid black;
}
```

NOTE: The **position:relative;** declaration is a bit complex and is used to give you more of a choice on how you want the positioning of your element to react to its surroundings; **top:0px;** moves the element so that the top of the element is at the top of the parent element, which in this case is the wrapper.

Adding a Navigation Bar

Another *div* element can be placed below the banner to serve as the navigation bar to reach other pages on your website.

To add a navigation bar:

1. Add a new line after ** **, and then type the following code.

```
<div class="nav"></div>
```

2. To define a new class for the navigation bar, add a new line before the closing **</style>** tag, and then type the following CSS rule.

```
div.nav{
    background-color:#5C5C5C;
    width:750px;
    height:30px;
    margin-left:auto;
    margin-right:auto;
    position:relative;
    top:3px;
    border:1px solid black;
}
```

NOTE: By declaring **position:relative;**, the navigation bar will appear naturally after the banner. Also, **3 pixels** of space are added between the top of the navigation bar and the bottom of the banner.

3. Press **Ctrl+S** to save the changes.

Checkpoint Two

This is the second checkpoint within this handout. Make sure that the document that is being created is similar to the *checkpoint_2.html* file located in the data files folder (see Figure 7).

NOTE: The code shown in Figure 7 is only the code added since checkpoint one.

```
div.banner{
    width:750px;
    height:90px;
    position:relative;
    top:0px;
    margin-left:auto;
    margin-right:auto;
    border:1px solid black;
}
div.nav{
    background-color:#5C5C5C;
    width:750px;
    height:30px;
    margin-left:auto;
    margin-right:auto;
    position:relative;
    top:3px;
    border:1px solid black;
}
```

```
<body>
    <div class="wrapper">
        <div class="banner"></div>
        <div class="nav"></div>
    </div>
    </div>
</body>
```

Figure 7 – Source Code in the checkpoint_2.html File

Adding the Main Content Containers

Up to this point, the web page that is being created includes a container and the upper portion of the page. In this section, additional important containers will be added to the page.

Adding the Main Content

In this lesson, a *div* element is placed below the navigation bar to serve as a container for the main content of the web page.

To add the main content:

1. Add a new line after ** **, and then type the following code.

   ** **

2. To define a new class for the main content, add a new line before the closing **</style>** tag, and then type the following CSS rule.

   ```
   div.main{
       position:relative;
       top:6px;
       width:750px;
       height:600px;
       margin-left:auto;
       margin-right:auto;
       border:1px solid black;
       background-color:white;
   }
   ```

   NOTE: By declaring **position:relative;**, the main content will appear naturally after the navigation bar. Also, **6 pixels** of space are added between the top of the main content and the bottom of the navigation bar.

Adding the Footer

A footer can be added at the bottom of the web page to help give some background information about the page or to provide a large area for useful links.

To add a footer:

1. Add a new line after ** **, and then type the following code.

   ** **

2. To define a new class for the footer, add a new line before the closing **</style>** tag, and then type the following CSS rule.

   ```
   div.footer{
       background-color:black;
       height:180px;
       width:752px;
       margin-left:auto;
       margin-right:auto;
       position:relative;
       bottom:8px;
       z-index:1;
       color:white;
   }
   ```

   NOTE: The **z-index:1;** declaration gives the footer a z-index of 1, meaning that it will be above all other elements. The z-index is a way to change the layering of elements without having to rearrange the tags. Elements without a z-index defined in the style default to 0.

Adding Text to the Web Page

Most web pages are made up of a lot of text. There are many things you can do to make your web page text look more interesting.

To add text:

1. Add a new line between the opening ** ** tag and the closing ** ** tag, and then type the following code.

   ```
   <p class="welcome">
       This is some text.
   </p>
   ```

   NOTE: The <p> tag defines a paragraph.

2. To define a new class for the paragraph, add a new line before the closing **</style>** tag, and then type the following CSS rule.

   ```
   p.welcome{
       margin-left:14px;
       margin-right:15px;
       text-indent:16px;
   }
   ```

   NOTE: The CSS rule above will only affect **<p>** tags with the class **welcome** applied to them.

3. To see the result, save the file, and then refresh the page in the browser window.

Checkpoint Three

This is the third checkpoint within this handout. Make sure that the document that is being created is similar to the *checkpoint_3.html* file located in the data files folder (see Figure 8).

NOTE: The code shown in Figure 8 is only the code added since checkpoint two.

```
div.main{
    position:relative;
    top:6px;
    width:750px;
    height:600px;
    margin-left:auto;
    margin-right:auto;
    border:1px solid black;
    background-color:white;
}
div.footer{
    background-color:black;
    height:180px;
    width:752px;
    margin-left:auto;
    margin-right:auto;
    position:relative;
    bottom:8px;
    z-index:1;
    color:white;
}
p.welcome{
    margin-left:14px;
    margin-right:15px;
    text-indent:16px;
}
```

```
<body>
    <div class="wrapper">
        <div class="banner"></div>
        <div class="nav"></div>
        <div class="main">
            <p class="welcome">
                This is some text.
            </p>
        </div>
        <div class="footer"></div>
    </div>
</body>
```

Figure 8 – Source Code in the checkpoint_3.html File

Basic Styling of Text

HTML text can be formatted and stylized much like in other programs such as Microsoft Word. To add more styles to the text, try adding some of the declarations listed below to the text class (p.welcome).

- text-decoration:underline;
- text-decoration:line-through;
- text-decoration:overline;
- color:red;
- color:blue;
- color:#00FF00;
- font-size:20px;

Appendix

Table 2 – Commonly Used HTML Tags

Tag	Description
<a>	Defines a hyperlink.
<b>	Makes text bold.
<body>	Defines the document's body.
 	Inserts a single line break.
	Renders as emphasized text.
<form>	Used to create an HTML form.
<h1> to <h6>	Define headings.
<head>	Container for all the head elements.
<hr>	Creates a horizontal line.
<html>	Tells the browser that this is an HTML document.
<i>	Renders text in italics.
<img>	Defines an image.
<input>	Specifies an input field.
	Defines a list item.
<menu>	Used to create a list of menu choices.
	Defines an ordered list.
<p>	Defines a paragraph.
<table>	Defines an HTML table.
<td>	Defines a standard cell in an HTML table.
<th>	Defines a header cell in an HTML table.
<title>	Defines the title of the document.
<tr>	Defines a row in an HTML table.
	Defines an unordered (bulleted) list.

Table 3 – Commonly Used HTML Color Codes

Color	Code	Color	Code
Black	#000000	Green	#008000
Silver	#C0C0C0	Lime	#00FF00
Gray	#808080	Olive	#808000
White	#FFFFFF	Yellow	#FFFF00
Maroon	#800000	Navy	#000080
Red	#FF0000	Blue	#0000FF
Purple	#800080	Teal	#008080
Fuchsia	#FF00FF	Aqua	#00FFFF

For more HTML color codes, visit:

- Webmonkey (http://www.webmonkey.com/2010/02/color_charts/)
- Colourlovers (http://www.colourlovers.com/blog/2007/06/30/ultimate-html-color-hex-code-list/)
- ComputerHope (http://www.computerhope.com/htmcolor.htm)