A publication of TeachUcomp Incorporated.

Mastering HTML5 and CSS3 Made Easy™

TEACHUCOMP, INC.®

...it's all about you

Copyright:

Trademark Acknowledgements:

Apple, Macromedia, Dreamweaver, CoffeeCup Software, eBay, Google, Intuit, Quicken, QuickBooks, QuickBooks Pro, QuickBooks Premier, Turbo Tax, EasyStep, QuickReports, and QuickZoom are registered trademarks of Intuit, Inc. Windows, Windows 95, Windows 98, Windows NT, Windows Me, Windows XP, Windows 7, Windows 8, Microsoft Word 97, Microsoft Word 2000, Microsoft Word XP, Microsoft Word 2003, Microsoft Word 2007, Microsoft Word 2013, Microsoft Excel 97, Microsoft Excel 2000, Microsoft Excel XP, Microsoft Excel 2003, Microsoft Excel 2007, Microsoft Excel 2013, and Outlook are registered trademarks of Microsoft Corporation. Other brand names and product names are trademarks or registered trademarks of their respective holders.

Disclaimer:

While every precaution has been made in the production of this book, TeachUcomp, Inc. assumes no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein. These training materials are provided without any warranty whatsoever, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. All names of persons or companies in this manual are fictional, unless otherwise noted.

TeachUcomp, Inc.
Phone: (877) 925-8080
Web: http://www.teachucomp.com

Introduction and Overview

Welcome to TeachUcomp, Inc.'s Mastering HTML5 and CSS3 Made Easy™ course. This course introduces the student to HTML (Hypertext Markup Language) the language used to create webpages on the Internet and Cascading Style Sheets.

The purpose of this course is to educate the student in the basic language skills necessary in web design. Several HTML editors exist today that use WYSIWYG (What You See Is What You Get) interfaces that do much of the work for you. Popular software applications such as Adobe's Dreamweaver and WordPress allow people to create blogs and webpages while knowing very little HTML. However, it's important to first understand the underlying language and components of web design in order to discover the true power of those applications.

Whether you are designing a website for your small business, learning to use WordPress, or you simply want to jazz up your eBay listings, having a solid foundation in HTML is the place to begin.

We will begin this course by discussing the basic components and structure of HTML as well as learning the terminology. Then, we'll advance through topics to cover CSS (Cascading Style Sheets), forms, and adding media such as video and audio files to a site.

TABLE OF CONTENTS

Table of Contents

Menu Bar or Ribbon:

When menu items from the menu bar or ribbon are referenced, the main menu title will be displayed, followed by a “|”, followed by the menu item.

Example: Edit| Copy.

Keyboard Shortcuts:

When keyboard shortcuts are referenced, the keyboard combination will be displayed as the first key which is held down, followed by a “+”, followed by the second key which is pressed and released quickly.

Example: CTRL+A

HTML Tags:

HTML tags generally appear as pairs, (with a few exceptions such as meta tags). The two tags are often used in conjunction, separated by text, to start and stop a specific action. In this manual, tags are often listed together and separated by an ellipsis (…) which represents the webpage content between the tags.

Example: <body>…</body>

Whenever attributes are available for a tag, the attribute will be referred to in UPPERCASE within explanations, but in lowercase as it would appear in normal usage in examples.

Example: <p align="center"> Your Text Here </p>

CHAPTER 1- Getting Acquainted with HTML

Getting Acquainted with HTML

1.1- Introduction to the Internet:

What began as a military research project in the 1960s, has evolved into the phenomenon today known as the World Wide Web (“the Web”) – connecting almost 2.5 billion users across the planet through a vast network of computers. The Internet has transformed the way people share information and revolutionized commerce by opening the global marketplace to businesses and individuals alike. Pages on the Internet contain objects such as text, images and audio/video. These webpages are saved on servers, which are large computers used to store and transfer information. Once a webpage is created and uploaded to the server, it can be viewed by others on the Internet.

You connect to the Internet through an Internet Service Provider(ISP). There are thousands of ISPs, ranging from large commercial companies to smaller service providers in your area. When you connect to the Internet, your computer communicates with other computers to exchange data.

In addition to your computer’s processor speed and other factors, your Internet connection speed plays an important role in your experience on the Internet. The speed of your connection means how much data can be exchanged in a unit of measure called BPS (Bits Per Second). The faster your connection, the faster webpages load on your computer for you to view.

Internet connections run on a collection of protocols (a set of rules) that govern how computers communicate with one another. This set of rules is called Transmission Control Protocol/Internet Protocol (TCP/IP). Each page on the Internet has a unique address, called a URL (Uniform Resource Locator). When you know a Web address, you generally use HTTP (Hypertext Transmission Protocol) to reach a specific address (http://wwww.teachucomp.com). HTTPS (Hypertext Transmission Protocol, Secure) is used when handling secure transactions that require information to be encrypted such as credit card information in a commerce transaction. FTP (File Transfer Protocol) is used to physically transfer files to a different computer. When you upload your webpages to a Web server, you use FTP.

Webpages are viewed using a Web browser. A browser is simply a program that reads, interprets and displays webpages on your computer. Microsoft’s Internet Explorer and Google Chrome are two of the most popular. However, many more exist such as Mozilla’s Firefox, Apple’s Safari, and Opera. Each of these browsers can interpret data in a slightly different manner, making this as well as connection speeds important considerations in Web design.

With the larger range of radio wavelengths being utilized for mobile broadband and our ability to miniaturize electronics, mobile phones and tablets now make up a large contingent of devices being used to access the internet.

1.2- Introduction to HTML Terminology:

HTML is the language used to create webpages. HTML stands for Hypertext Markup Language. HTML documents written with this text (or coding) tell browsers how to interpret and display the data. HTML documents can be identified by the file extension .htm or .html. HTML standards are set and maintained by an international group of industry leaders such as Microsoft and Apple, called The World Wide Web Consortium (W3C). The rules that govern how HTML is written are called syntax. As new Web technologies emerge, HTML evolves through the W3C with newer versions such as 5.0 that introduced new standards discussed in upcoming chapters.

1.2- Introduction to HTML Terminology (cont.):

The “Markup Language” component of HTML refers to the insertion of instructions, called tags. ***Tags*** tell the Web browser how to interpret the data. Tags follow a standard format. Each tag begins with a “less than” symbol (<), immediately followed by the tag text, and ending in a “greater than” symbol (>). Spelling is critical, as tags not recognized by a Web browser are ignored. Tags in HTML can be written in either upper or lowercase. However, it is considered generally good practice to type your tags in lower case.

Tag Example: *<body>*

Most tags have a similar counterpart, called an “end tag” which stops the effect of the tag. An end tag is identical to the start tag counterpart, but also includes a forward slash “/”immediately before the tag text. Some tags do not have a closing counterpart, such as the Image tag (<img>). A closing tag must always include the forward slash “/”.

End Tag Example: *</body>*

Elements are simply the different components or pieces of your webpage (text, images, etc.). For example, <body>…</body> are tags that identify the body text on a webpage.

Once you have identified the element, you can then apply ***attributes*** such as color and alignment to the element. Many attributes require that you also set a value, such as a measurement or specification. For example, if you wanted to center a paragraph on your webpage, the code would read:

Attribute Example: *<p style=“text-align:center”>Your paragraph text.</p>*

In this example, STYLE is the attribute for the paragraph’s text and TEXT-ALIGN:CENTER is the value. Values are always surrounded by quotation marks. You can apply multiple attributes, separating them with space between.

Entities are special characters you can add to a page such as a copyright or trademark symbol. Entities begin with an ampersand (&) and end with a semicolon. For example, the code to insert a registration mark (®) looks like:

Entity Example: &reg;

As HTML became more prevalent and the need increased for additional structure for HTML documents, the W3C introduced XML (Extensible Markup Language), a meta-markup language used to create other languages and allow developers to define their own tags. XML isn’t quite as flexible as HTML, so the W3C rewrote HTML in XML, creating XHTML.

With the advent of HTML5, the W3C recommends using HTML5, as its inclusion of the extensive use of Cascading Style Sheets (CSS) and JavaScript (JS) has eradicated the need for XHTML. While XHTML is still being supported by browsers, its similarity to the latest iteration of HTML makes it redundant.

1.2- XML, XHTML, DHTML (cont.):

DHTML stands for Dynamic Hypertext Markup Language. DHTML is an extension of HTML that allows webpages to react to a user's input and enables the inclusion of such things as animation and dynamic menus. DHTML uses a combination of HTML, Cascading Style Sheets (CSS) and JavaScript – some of which we cover in later chapters. While DHTML is an extension of HTML it is not considered a "language" in itself as it encompasses several languages under the "umbrella" of the term Dynamic.

1.3- Options for writing HTML:

When it is time for you to start creating your HTML documents, you must decide which program you will use to write your code. Your choices are varied and include:

Simple Text Editors:

Most computers come with a simple text editor pre-installed such as Microsoft Window's Notepad or Apple's TextEdit. When Web development was in its infancy, simple text editors were widely used. Unlike the WYSIWYG programs of today, there are no "bells and whistles" and they allow for straightforward HTML coding.

Word Processing Programs:

Commercial word processing programs such as Microsoft Word and Google Docs can also be used to write HTML. However, these applications have limited file conversion and editing capabilities. In addition, they can sometimes insert extraneous code within your files which can interfere with HTML.

Advanced HTML Editors:

Advanced HTML editors are programs specifically designed for writing HTML code and creating webpages. Programs such as Adobe's Dreamweaver and CoffeeCup HTML Editor use a WYSIWYG (What You See is What You Get) graphical interface. With these powerful programs, you can create webpages while knowing very little about HTML. You can also easily switch to text-based editing within these applications, underscoring the importance of understanding HTML.

Proprietary Sources:

Many popular sites these days such as eBay and Etsy allow users to customize their content using HTML. These sites will provide you with the work area in which to write your HTML code.

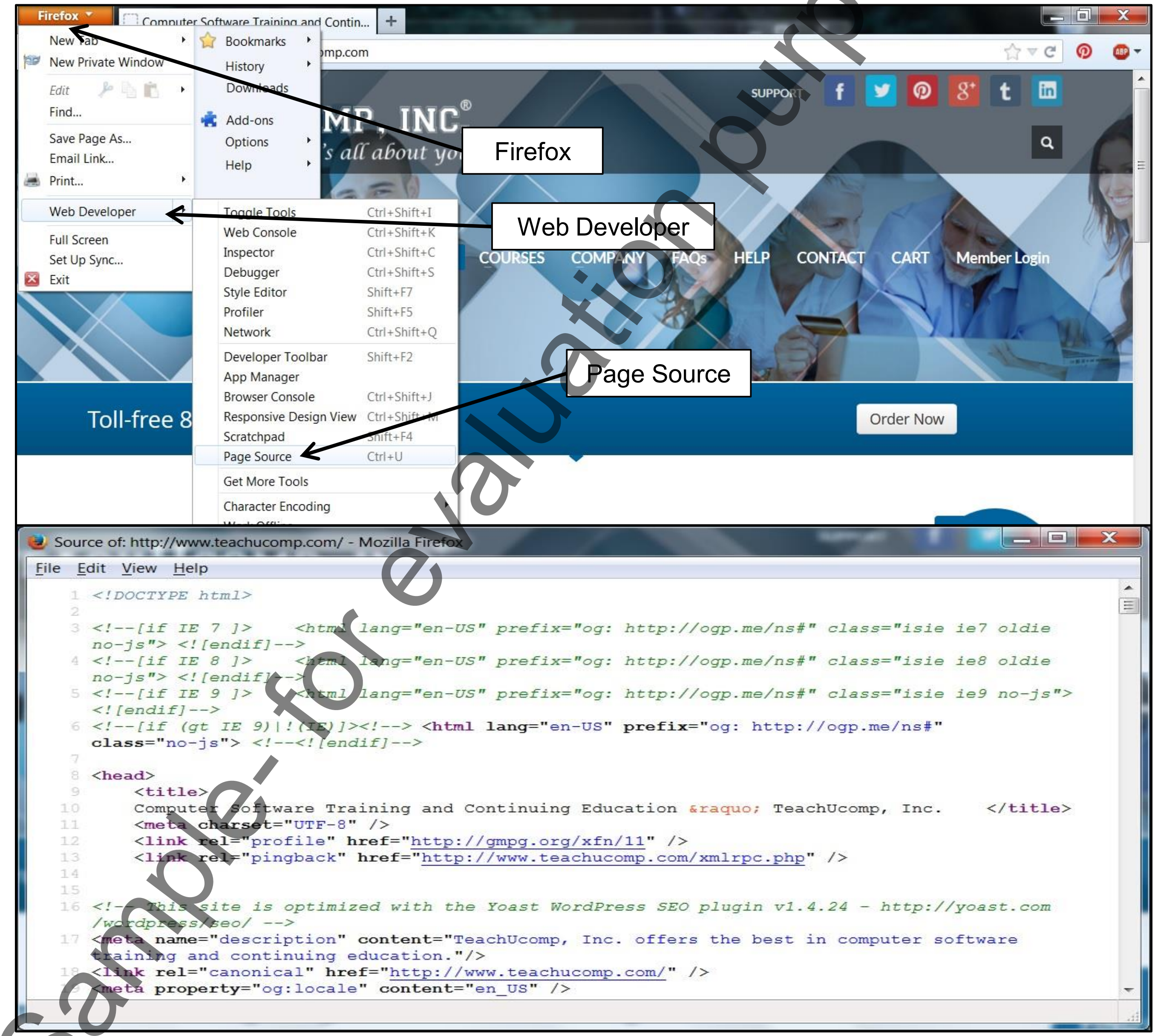
1.4- Unicode Transformation Format:

Unicode Transformation Format (UTF) is a set of standardized characters used to create HTML documents and was created to replace existing character sets to make a universal character set across all platforms. HTML5 supports both UTF-8 and UTF-16. UTF-8 is the accepted character set for encoding e-mail and all webpages while UTF-16 is used for major operating systems and environments like Windows and Java.

1.5- HTML Resources:

There is a wealth of information available at your fingertips to help guide you in writing your own HTML code – in the form of webpages that currently exist on the internet. You can view the HTML code of a webpage (unless the developer has protected the code), allowing you to learn from others and spark new ideas in your own development efforts. If you like, you can save the webpage and even print the HTML code, providing you instructions on how to replicate the desired effects on your site.

To view the source code in a webpage, when using Internet Explorer select View| Source, when using Firefox select Firefox| Web Developer| Page Source. Alternatively the keyboard shortcut CTRL + U can be used in most browsers, the exception being Apple's Safari browser, which requires CTRL+ALT+U.



```
<!DOCTYPE html>

<!--[if IE 7 ]>     <html lang="en-US" prefix="og: http://ogp.me/ns#" class="isie ie7 oldie no-js"> <![endif]-->
<!--[if IE 8 ]>     <html lang="en-US" prefix="og: http://ogp.me/ns#" class="isie ie8 oldie no-js"> <![endif]-->
<!--[if IE 9 ]>     <html lang="en-US" prefix="og: http://ogp.me/ns#" class="isie ie9 no-js"> <![endif]-->
<!--[if (gt IE 9)|!(IE)]><!--> <html lang="en-US" prefix="og: http://ogp.me/ns#" class="no-js"> <!--<![endif]-->

<head>
    <title>
    Computer Software Training and Continuing Education &raquo; TeachUcomp, Inc.    </title>
    <meta charset="UTF-8" />
    <link rel="profile" href="http://gmpg.org/xfn/11" />
    <link rel="pingback" href="http://www.teachucomp.com/xmlrpc.php" />


<!-- This site is optimized with the Yoast WordPress SEO plugin v1.4.24 - http://yoast.com/wordpress/seo/ -->
<meta name="description" content="TeachUcomp, Inc. offers the best in computer software training and continuing education."/>
<link rel="canonical" href="http://www.teachucomp.com/" />
<meta property="og:locale" content="en_US" />
```

CHAPTER 2- New for HTML5

2.1- What's Different in HTML5?

2.2- <!DOCTYPE> in HTML5

2.1- What's Different in HTML5?:

There are many changes to HTML5 that make it much easier to work with. The semantics and structure are clearer and more straight forward than previous versions of HTML. New Elements have been added and redundant Tags have been removed. The insertion of media has been streamlined and it has been updated to work across a multitude of platforms, i.e. home computers, tablets and mobile phones. We will cover the new additions in the relevant Chapters that follow in this course.

Most importantly, if you are familiar with HTML4 and new to HTML5, it's important to note that the following are Tags that are no longer supported in HTML5. This means that most browsers will give you errors or your webpage will not look as you had intended. We recommend reviewing any previous code on your webpage and removing or replacing these tags.

<acronym>
<applet>
<basefont>
<big>
<center>
<dir>
<font>
<frame>
<frameset>
<noframes>
<strike>
<tt>

2.2- <!DOCTYPE> in HTML5:

The <!DOCTYPE> tag is not actually a "Tag" in HTML5. It is a declaration to let the browser you are using know what version of HTML you have written your code in. This **MUST** be the first thing on any page you are creating before the <html> tag. <!DOCTYPE> is not case sensitive and it does not have an end tag. The beginning of the pages you create should look this way:

```
<!DOCTYPE html>
```

followed by your code.

CHAPTER 3- Designing a Webpage

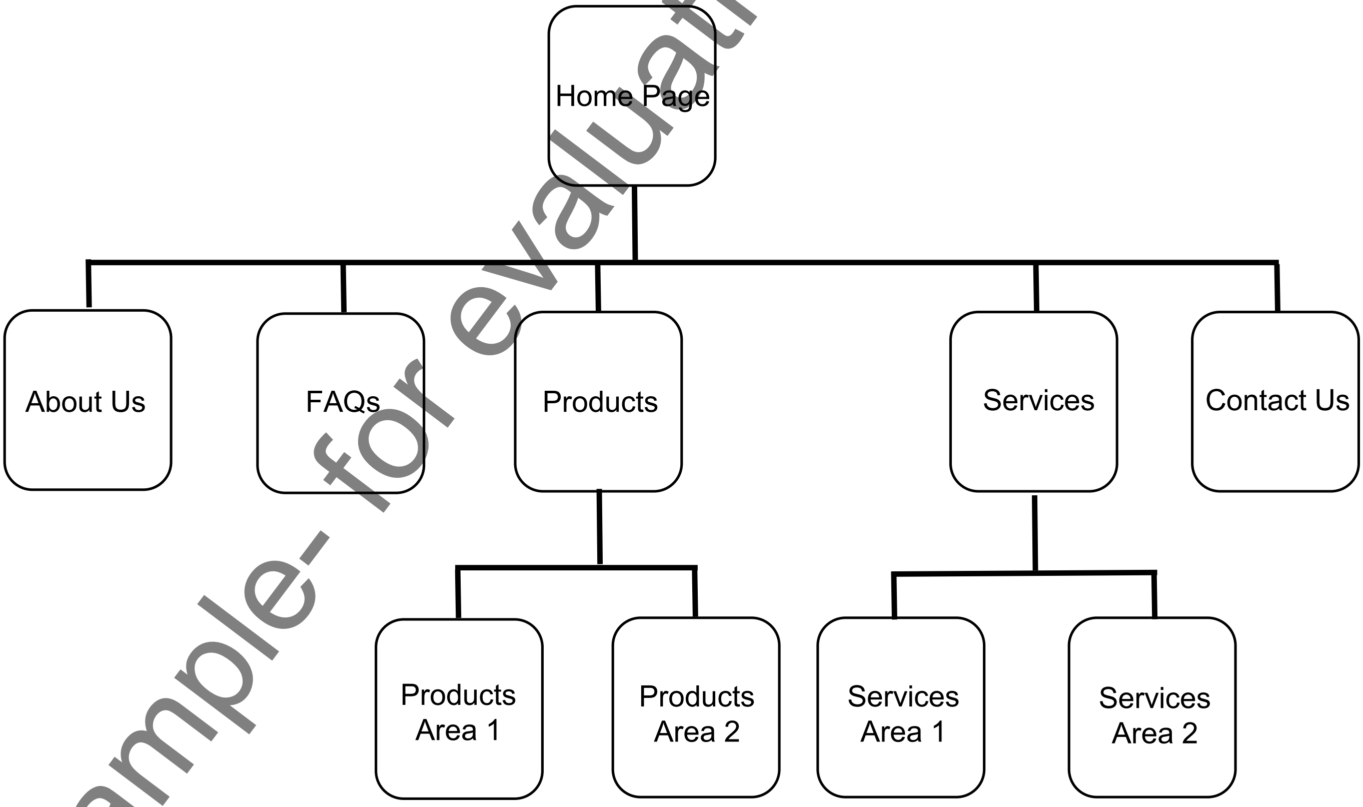
3.1- Design Considerations and Planning:

Before you sit down to design your first webpage, you should have an idea of what you want your end result to look like. Take out a piece of paper and pen or pencil and draw a rough sketch of your page. Lay out where you want to place your pictures, text and other elements. Your draft will be your blueprint as you develop your page. If you are designing an entire website (that is, a collection of webpages at a single URL), your planning is especially important.

In addition to deciding on how you want each page to look, you have to decide how to organize the pages so that information is easily accessible to people as they view your site. Many websites have common characteristics (home page, contact page, FAQ page, etc.). A home page is probably the most basic. It's the main page of your site that usually provides a brief introduction to the entire site. Home pages are usually named index.htm or index.html and provide links to the rest of the pages.

As you are getting started, visit a few of your favorite websites. Take note of things you like and don't like. How is the information laid out? Can I easily navigate from one page to another? How could the information be better organized? In addition to layout considerations, this is also a great way to inspire you in design details. What makes the site more user-friendly? Are there creative uses of text or images that are particularly appealing?

Once you have decided on your content and layout, make a simple flowchart of your site. It will give you a visual map to work from. Below is an example of a simple website flowchart.



3.2- Basic Tags and Document Structure:

Once you have decided on your content and layout, it's time to design your first HTML page. Start by opening the text editor of your choice. While every webpage is different in terms of content and layout, each page has a basic structure and tags as the building blocks. Every HTML document uses the basic structural tags below:

```
<!DOCTYPE html>
 <html>
  <head>
    <title>
    Your Browser Title text goes here
    </title>
  </head>
  <body>
  The contents of your webpage goes here
  </body>
</html>
```

3.3- <HTML>…</HTML>:

The <html> and </html> tags are the main tags and identify your page as an HTML document. When a Web browser reads your document, it knows that everything between these two tags is an HTML document. Each HTML page you create should start with <!DOCTYPE html>, <html> and end with </html> as shown in the example below.

Beginning of every page:	<!DOCTYPE html>	Tells browser version of HTM
Start Tag:	<html>	
End Tag:	</html>	
Attributes:	None	
Example:	(Start of HTML document) <!DOCTYPE html> <html> ... </html> (End of HTML document)	These tags must be the first and last tags used in your HTML document.

3.4- <HEAD>…</HEAD>:

The <head>…</head> tags are used to define your document header. This is where you add basic information about your webpage such as the title and metadata (author information, keywords, etc). These document header tags immediately follow the opening HTML tag <html> in your document.

Start Tag:	<head>	
End Tag:	</head>	
Attributes:	None	
Example:	<!DOCTYPE html> <html> <head> <title> Browser Title</title> </head> <body> … </body> </html>	These tags define the heading area of the document

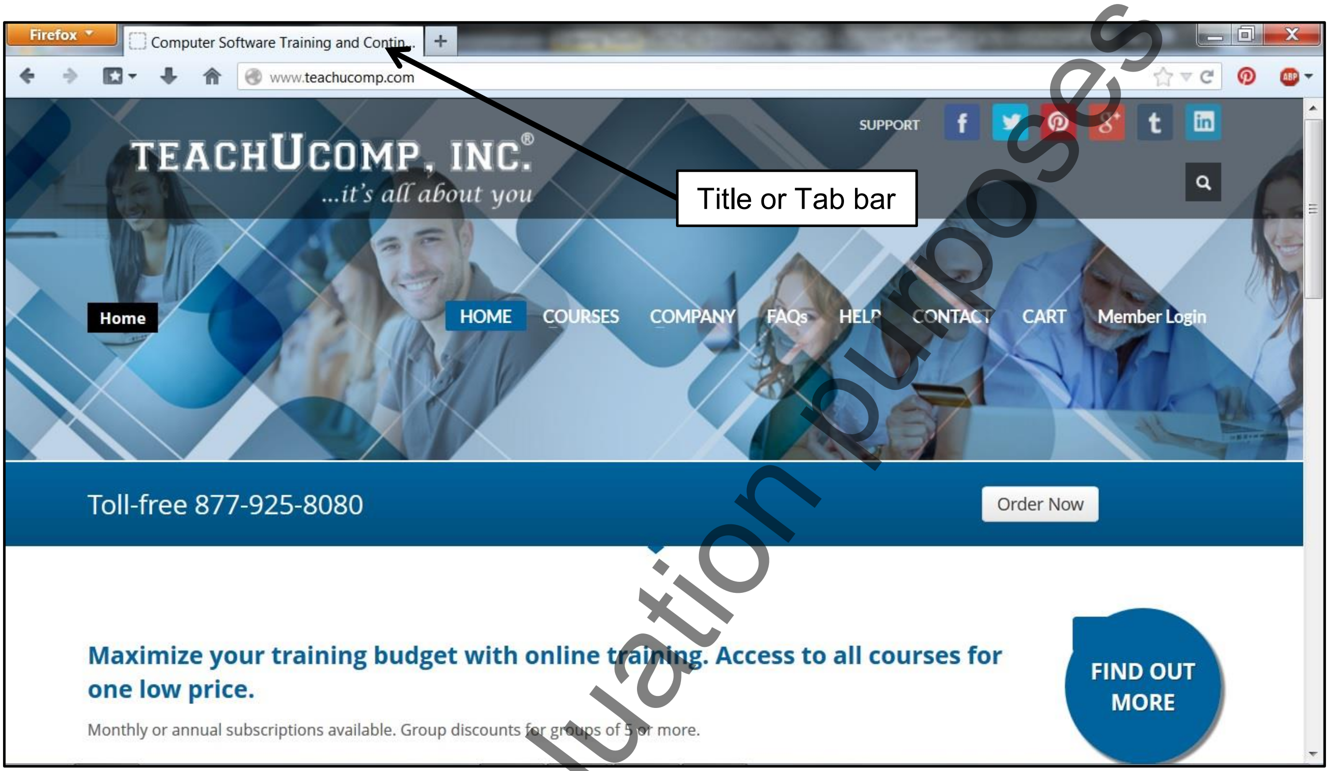
3.5- <TITLE>…</TITLE>:

Title tags allow you to give your HTML page a name which is displayed in your browsers' title or tab bar, depending if you have more than one tab open. It is also the name that appears when a user adds your webpage to their "Favorites" or "Bookmarks" folder. A "Favorites" or "Bookmarks" folder is a collection of pages that a user saves to quickly locate pages at a later time. Your title helps users identify your page, so it should be concise and relevant.

Start Tag:	<title>	
End Tag:	</title>	
Attributes:	None	
Example:	<!DOCTYPE html> <html> <head> <title> TeachUcomp, Inc. Home Page</title> </head> <body> … </body> </html>	These tags define the information to be displayed in a browsers' title bar.

3.5- <TITLE>…</TITLE> (cont.):

The following pictures show where your title tags appear in a browser.

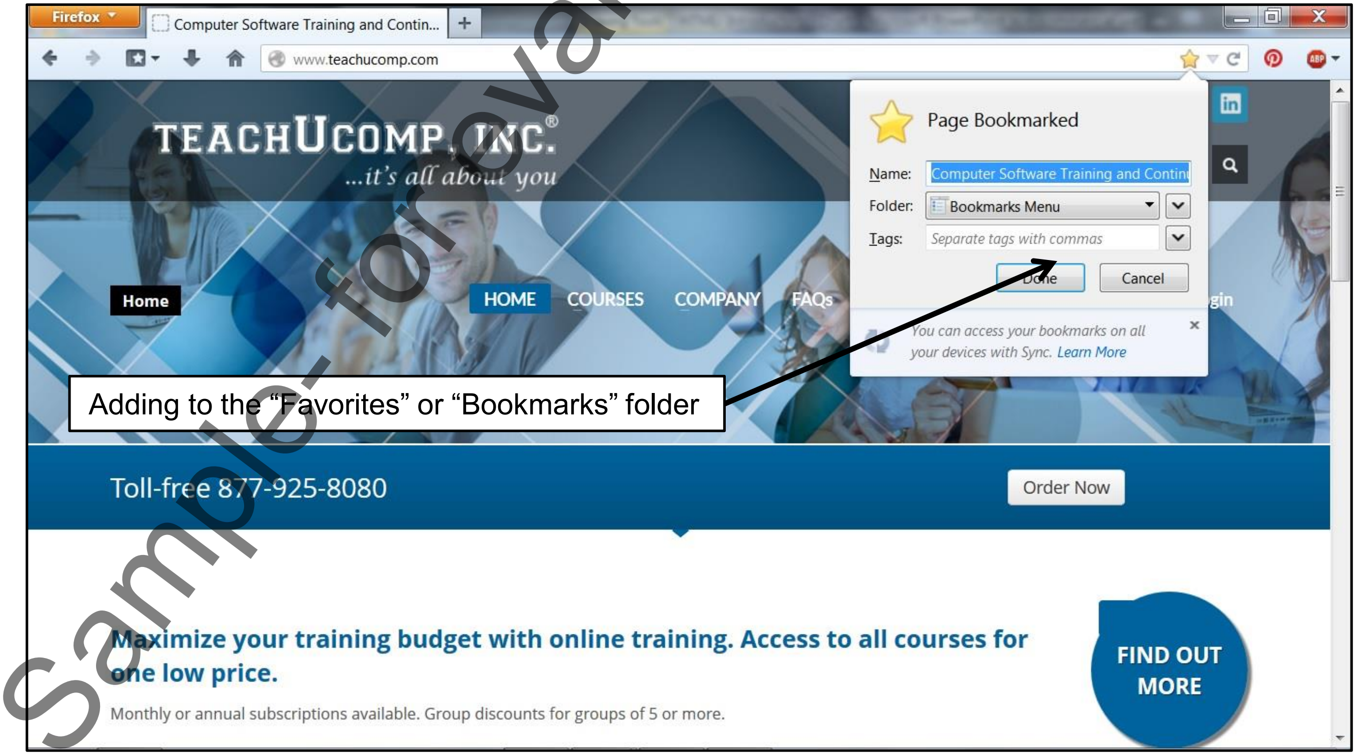


Title or Tab bar

Toll-free 877-925-8080

Maximize your training budget with online training. Access to all courses for one low price.

Monthly or annual subscriptions available. Group discounts for groups of 5 or more.



Name: Computer Software Training and Contin

Folder: Bookmarks Menu

Tags: Separate tags with commas

You can access your bookmarks on all your devices with Sync. Learn More

Adding to the “Favorites” or “Bookmarks” folder

Toll-free 877-925-8080

Maximize your training budget with online training. Access to all courses for one low price.

Monthly or annual subscriptions available. Group discounts for groups of 5 or more.

3.6- <BODY>…</BODY>:

The <body>…</body> tags identify the area in your webpage where all of your content is stored. Everything you place between these tags appears in your HTML document. Within these tags, you will place your text, images, lists, tables, forms and other elements. In addition, you can use attributes in this area to define default colors for the text and links.

Start Tag:	<body>	
End Tag:	</body>	
Attributes:	text=	Specifies a default text color
	link=	Specifies the default text color used to identify a unvisited links.
Example:	<body> …(Main browser window content) </body>	These tags define information to be displayed in the browser's main window.

3.7- Metadata:

Metadata is a term used to describe the details of your webpage, such as a page description, keywords, copyrights and more. Metadata does not appear on the actual webpage when it is displayed in a browser. Instead, it is used to identify information about your webpage. More importantly, search engines such as Google and Yahoo use the information in your metadata tags (primarily keywords and page descriptions) to help determine where your pages will be ranked in search results.

For example, by placing the term "HTML tutorial" in the keywords metadata and "HTML Tutorial by TeachUcomp, Inc." in the description metadata, search engines will know that the webpage is relevant when a user type in "HTML tutorial" in a search engine. To insert multiple keywords, use a comma and space to separate the keywords. While there are many other factors search engines use to determine ranks, your metadata is an important first step in optimizing your pages.

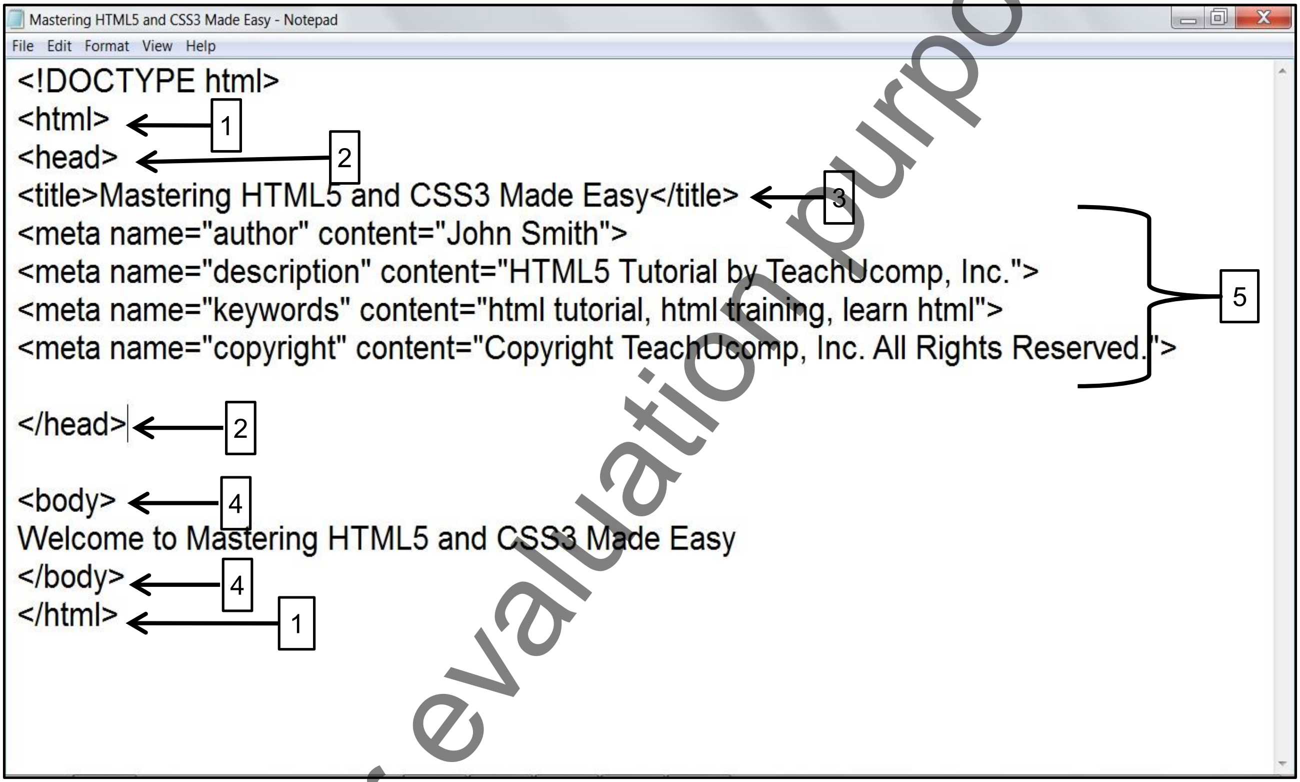
Metadata is placed between the <head>…</head> tags (generally following the <title>…</title> tags) and is also used by others who view your source code to learn about you, your pages and any other information you may want to relay.

Tag:	<meta…>	Does not require a closing tag.
Attribute:	name=	
	author	Specifies the author of the HTML page.
	description	Define a broad description of the HTML page.
	keywords	Identifies Keywords relevant to the content of the HTML page. Used by search engines to index and rank pages.
	copyright	Specifies copyright information.
Examples:	<meta name="author" content="John Smith">	Identifies John Smith as the author of this HTML page.
	<meta name="description" content="HTML Tutorial by TeachUcomp, Inc.">	Specifies the description of the HTML page.
	<meta name="keywords" content="html tutorial, html training, learn html">	Specifies the keywords relevant to the HTML page.
	<meta name="copyright" content="Copyright TeachUcomp, Inc. 2014. All rights reserved.">	Specifies the copyright statement.

3.7- Metadata (cont.):

As we have discussed, below are the basic structure tags of an HTML document.

1. <html> tags.
2. <head> tags.
3. <title> tags.
4. <body> tags.
5. Metadata.



```
<!DOCTYPE html>
<html>
<head>
<title>Mastering HTML5 and CSS3 Made Easy</title>
<meta name="author" content="John Smith">
<meta name="description" content="HTML5 Tutorial by TeachUcomp, Inc.">
<meta name="keywords" content="html tutorial, html training, learn html">
<meta name="copyright" content="Copyright TeachUcomp, Inc. All Rights Reserved.">

</head>

<body>
Welcome to Mastering HTML5 and CSS3 Made Easy
</body>
</html>
```

3.8- Saving an HTML Page:

If you are using an advanced editor such as Dreamweaver or CoffeeCup, your documents will be saved as HTML pages by default. If you are using a simple text editor such as Notepad or Wordpad, save your files as a text document under “Save as type.” Type your file name, followed by the .htm or .html extension.

Your file name *must not* contain any blank spaces. Choose a file name that is concise, descriptive and easy to remember. This will make it easier for users as well as yourself when you go back to edit pages in the future.

ACTIONS-
DESIGNING A WEBPAGE

CREATE AN HTML PAGE:

1. In a new document in your HTML text editor, type the declaration: <!DOCTYPE html>
2. On a new line, type the start tag: <html>
3. On a new line, type the end tag: </html>
4. Everything you place between these two tags will be part of your HTML document.

ADD A DOCUMENT HEADER TO AN HTML PAGE:

1. On a new line immediately following the <html> start tag, type the start tag: <head>
2. On a new line, type the end tag: </head>
3. All information between these two tags will be included in your header.

ADD A TITLE TO AN HTML PAGE:

1. Between the <head>…</head> tags, type the start tag: <title>
2. Type the text you want displayed as your page title in browsers.
3. Type the end tag: </title>

ADD THE BODY OF AN HTML PAGE:

1. On a new line following the end tag </head>, type the start tag: <body>
2. On a new line, type the end tag: </body>
3. Everything between these tags will be the body of your HTML page.

ADD METADATA TO YOUR HTML PAGE:

1. On a new line following the end tag </title> and between the <head>…</head> tags, type the tag <meta name="?" [where "?" is the metadata to insert ("author", "description", "keywords", or "copyright")]
2. Type a space and then: content=
3. Type the metadata in quotes
4. Close the tag with: >

SAVING AN HTML PAGE IN A SIMPLE TEXT EDITOR:

1. Select "File| Save As…" from the Menu Bar or Ribbon.
2. In the Dialog Box, navigate to the correct folder where you want to save your HTML document.
3. Use the drop-down under "Save as type" and select ".txt."
4. In the "File Name" box, enter a descriptive name for your HTML document, with a .html or .htm extension.
5. Click "Save".

EXERCISES:
DESIGNING A WEBPAGE

Purpose:

1. To create and save the basic structure of an HTML page.

Exercises:

1. Open your HTML editor and begin a new page.
2. Type: <!DOCTYPE html>
3. Press “Enter”.
4. Type: <html>
5. Press “Enter”.
6. Type: <head>
7. Press “Enter”.
8. Type: <title>
9. Type: My Webpage
10. Type: </title>
11. Press “Enter”.
12. Type: <meta name=“author” content=“Me”>
13. Press “Enter”.
14. Type: <meta name=“description” content=“My practice page”>
15. Press “Enter”.
16. Type <meta name=“keywords” content=“practice html, my practice page”>
17. Press “Enter”.
18. Type <meta name=“copyright” content=“Copyright Me. All Rights Reserved”>
19. Press “Enter”.
20. Type: </head>
21. Press “Enter”.
22. Type <body>
23. Press “Enter”.
24. Type: “Welcome to my Webpage”
25. Press “Enter”.
26. Type: </body>
27. Press “Enter”.
28. Type: </html>
29. In your HTML editor, Select “File| Save As…” from the Menu Bar or Ribbon.
30. In the Dialog Box, navigate to the correct folder where you want to save your HTML document.
31. Use the drop-down under “Save as type” and select “.txt”.
32. In the “File name” box, enter “my-practice-page” with a .html or .htm extension.
33. Click “Save”.

CHAPTER 4-
Page Formatting

4.1- Adding a New Paragraph:

Now that we understand the basic building blocks of an HTML page, let’s examine page formatting. When you are working in a word processing program such as Microsoft Word, you press the “Enter” (or Return) key on your keyboard to begin a new paragraph. Web browsers do not recognize those line breaks. Instead, you must insert paragraph tags <p>…</p> each time you want to start a new paragraph.

Start Tag:	<p>	
End Tag:	</p>	
Attributes:	None	
Example:	Welcome to Mastering HTML5 and CSS3 Made Easy <p>This is how you create new paragraphs.</p> <p>See how easy it is to do?</p>	
Result:	Welcome to Mastering HTML5 and CSS3 Made Easy This is how you create new paragraphs. See how easy it is to do?	

4.2- Adding a Line Break

By default, browsers ignore many formatting keystrokes that we take for granted. Examples include the “Enter” and “Tab” keys and multiple uses of the spacebar. To accomplish the same tasks in HTML, we use page formatting tags.

Web browsers wrap text automatically to the next line when the current line reaches the right side of the browser. If you want to avoid wrapping and begin text on a new line, you use the <br> tag. The <br> tag does not have an end tag. You can also add additional lines between paragraphs by using the <br> tags. Each <br> tag you enter creates another blank line.

Start Tag:	<br>	
End Tag:	None	
Attributes:	None	
Example:	<p>This is where your fist line of text goes<br><br>Your second sentence would begin on the next line. </p>	Each <br> tag begins a new line.
Result:	This is where your first line of text goes. Your second sentence would begin on the next line.	

4.3- Inserting Blank Spaces:

Any blank spaces that you type in your text (beyond a single space between words) are ignored by browsers. You must code your desired blank spaces into your document. You can insert blank spaces into any lines of text. Since blank spaces are not available on your keyboard, you must use the entity **&nbsp;** for each space you wish to add. For example, if you wanted to add multiple spaces between specific words, type in the appropriate amount of entities without any spaces between them.

Entity:	&nbsp;	
Example:	<p>This would add five &nbsp;&nbsp;&nbsp;&nbsp;&nbsp; blank spaces.</p>	
Result:	This would add five blank spaces.	

4.4- Preformatted Text:

If you want to avoid some of the repetitive coding that comes with multiple spacing in your document, you can use the preformatted tags <pre>…</pre>. The <pre>…</pre> tags allow you to maintain the integrity of the spacing and formatting of your text so that browsers will display it as you have typed it.

For example, let’s say you want to create two columns of information with specific spacing. You place the <pre>…</pre> tags before and after your text to display the columns just as you have typed them.

Start Tag:	<pre>	
End Tag:	</pre>	
Attributes:	None	
Example:	<p>Today’s Specials:</p> <pre> Soup: New England Clam Chowder Salad: Beefsteak Tomato Entrée: Surf & Turf </pre>	Maintains the integrity of the formatted text.
Result:	Today’s Specials: Soup: New England Clam Chowder Salad: Beefsteak Tomato Entrée: Surf & Turf	

4.5- Changing a Page’s Background Color:

HTML coding uses hexadecimal values to assign colors. Hexadecimal values are assigned by typing a number sign (#) followed by a six character value (see Appendix). For very basic colors, you can also replace the six character value with the color name (red, yellow, green, etc.) see Appendix.

By default, the background of your pages will be white. To change the background color, place the attribute BGCOLOR within the <body> start tag of your page. Use caution when changing the background color of your page, as it can sometimes make it harder for users to read your text.

Attribute:	Bgcolor=	
Example:	<!DOCTYPE html> <html> <head> <title>TeachUcomp, Inc. Home Page</title> </head> <body bgcolor=“#0000FF”> … </body> </html>	Changes the background color of the page to BLUE.

4.6- Div Element:

The <div> tag is used to define a section or block on a webpage and can also be used in conjunction with CSS, for styling purposes, on large blocks or sections of content.

Start Tag:	<div>	
End Tag:	</div>	
Attributes:	None	
Example:	<!DOCTYPE html> <html> <body> <p>This is my first paragraph.</p> <div style="color:#0000FF"> <h3>This heading is set to blue in the div element.</h3> <p>This text is also set to blue.</p> </div> <p>This text is no longer inside the div element.</p> </body> </html>	Changes the text inside the <div>…</div> to the color blue.

ACTIONS-
PAGE FORMATTING

ADD A NEW PARAGRAPH:

1. Just prior to the text you want to turn into a new paragraph, type the start tag: <p>
2. At the end of the new paragraph's text, type the end tag: </p>

ADD A LINE BREAK:

1. At the point you want to enter a line break, type the tag: <br>

INSERTING BLANK SPACES:

1. For each blank space you wish to enter, type the entity: &nbsp;

ADD PREFORMATTED TEXT:

1. On the line immediately preceding the text you want to maintain formatting, type the start tag: <pre>
2. Type your text.
3. On the line immediately following your text, type the end tag: </pre>

CHANGE A PAGE'S BACKGROUND COLOR:

1. Within the <body> start tag, type the attribute: bgcolor=
2. Without a space between, type the hexadecimal or name of the color you wish to assign, in quotes.
3. Close the start tag by typing >

ADD A DIV ELEMENT:

1. At the point in your page you would like a separate section, type the start tag: <div>
2. Type or place the coding for your text, images, etc.
3. On the line immediately following your additions, type the close tag </div>

EXERCISES-
PAGE FORMATTING

Purpose:

1. To add a new paragraph to the HTML document created in the previous lesson and change the paragraph and page properties..

Exercises:

1. Open your HTML page created in the previous lesson.
2. Within the <body>…</body> tags of your HTML page and on the line after the text “Welcome to my Webpage”
3. Press “Enter”.
4. Type: <p>This text will serve as a test paragraph.</p>
5. Press “Enter”.
6. Type: <br>
7. Press “Enter”.
8. Type: After adding a line break, now I will enter five &nbsp;&nbsp;&nbsp;&nbsp;&nbsp; blank spaces.
9. Press “Enter”.
10. Type <div style=“color:#0000FF”>
11. Press “Enter”.
12. Type <pre>
13. Press “Enter”.
14. Type: I can also add   preformatted text.
15. Press “Enter”.
16. Type </pre>
17. Press “Enter”.
18. Type </div>
19. Within the <body> start tag type: <body bgcolor=“#FFFF00”>
20. Save Document

CHAPTER 5-
Text Items and Objects

5.1- Headings:

Headings are used to help organize information on your webpage or to create a structural hierarchy or even add visual interest. HTML offers you six different levels of headings. These levels are the opposite of text levels. For headings, <h1> is the largest and <h6> is the smallest. To create a heading, place the <h?>…</h?> around the text of your heading, replacing "?" with a value from 1 to 6. Note that the value must appear in both the start and end tags.

Start Tag:	<h?>	
End Tag:	</h?>	
Attributes:	None	
Example:	<h1>This is the largest heading.</h1>	This tag causes the text between the tags to be displayed as a level 1 heading.
Result:	This is the largest heading.	

5.2- Comments:

You can insert comments into your HTML code as a form of documentation or as notes to yourself or other designers viewing your code. Comments only appear within the HTML code and are not displayed by the browser. One common use of inserting comments is to remind yourself of future editing tasks.

Start Tag:	<!-->	
Attributes:	None	
Example:	<!--Add photo here-->	
Result:		Nothing is displayed in the browser

5.3- Block Quotes:

Block quotes are typically used for quoted text from other sources. To create block quotes, use the <blockquote>…</blockquote> tags around the text. This sets off the paragraph from the rest of your webpage.

Start Tag:	<blockquote>	
End Tag:	</blockquote>	
Attributes:	None	
Example:	<blockquote> An example of using block quotes. This will set the text apart from the rest of the document. </blockquote>	These tags cause the text between the tags to be displayed as block quotes.
Result:	An example of block quotes. This will set the text apart from the rest of the document.	

5.4- Horizontal Lines:

You can insert a solid horizontal line (sometimes called a horizontal rule) into your webpage to break up information or add visual interest. HTML requires that horizontal rules occupy a line by themselves. To insert a horizontal line, place the <hr> tag (there is no end tag) where you want the line to appear in your document. In HTML5 the <hr> tag will still display a horizontal line in visual browsers, but it is more commonly used as a thematic break at the paragraph level. For example, a location change in a story or a change of theme.

Start Tag:	<hr>	
End Tag:	None	
Attributes:	None	

5.5- Special Characters:

There are times when it becomes necessary to display characters and symbols that are not available on a standard keyboard such as © or that may have special meaning in HTML (<,>,&, etc). To accomplish this, HTML uses the Special Character Entity tag. The tag takes the format of an ampersand (&), followed by the code and a semicolon. For example, the © (copyright) symbol can be displayed by using &#169;. There are literally hundreds of special character entities currently available. A few of the more common tags are listed below.

Description	Special Character	Code
Acute Accent	´	&#180;
Ampersand	&	&#38;
Cent Sign	¢	&#162;
Copyright	©	&#169;
Dagger	†	&#8224;
Degree Sign	°	&#186;
Division Sign	÷	&#247;
Euro	€	&#8363;
Fraction (one-half)	½	&#189;
Fraction (one-fourth)	¼	&#188;
Fraction (three-fourths)	¾	&#190;
Greater-than sign	>	&#62;
Left-angle quote	«	&#171;
Less-than sign	<	&#60;
Multiply sign	x	&#215;
Plus or minus sign	±	&#177;
Quotation Mark (left)	“	&#8220;
Quotation Mark (right)	”	&#8221;
Registered Trademark	®	&#174;
Right-angel quote	»	&#187;
Superscript one	¹	&#185;
Trademark	™	&#8482;

ACTIONS-
TEXT ITEMS AND OBJECTS

ADD A HEADING:

1. Just prior to the text you want to turn into a heading, type the first part of the start tag: <h
2. Type the value of the heading you want (1 for the largest and 6 for the smallest).
3. Close the tag by typing: >
4. Type your heading text.
5. At the end of the heading, type the first part of the end tag </h
6. Type the same value for the heading that you put in the start tag (step #2)
7. Close the tag by typing: >

ADD COMMENTS:

1. Click into the body of your HTML document where you want to add comments.
2. Type: <!--
3. Type your comments.
4. Type: -->

ADD BLOCK QUOTES:

1. On the line before the text you want to turn into block quotes, type the start tag: <blockquote>
2. Type your text.
3. On the line following your text, type the end tag: </blockquote>

ADD A HORIZONTAL LINE:

1. To insert a horizontal line in your page, type: <hr>

ADD SPECIAL CHARACTERS:

1. At the point in your text where you want to add a special character, type: &#
2. Type the proper code for the character you want to add.
3. Type: ;

EXERCISES-
Text Items and Objects

Purpose:

1. To add a comment, block quotes and special characters to the HTML document created in the previous chapters.

Exercises:

1. Open your HTML page created in the previous chapters.
2. On the line after the last text in the body, type: <h1>This is the largest heading.</h1>
3. Press “Enter”.
4. <!—Add photo here.-->
5. Press “Enter”.
6. Type: <blockquote>To be or not to be, that is the question.</blockquote>
7. Press “Enter”.
8. Type: <hr>
9. Press “Enter”.
10. Type: <p>I can add special characters, like the copyright sign:&#169;</p>
11. Press “Enter”.
12. Save Document

CHAPTER 6-
Creating Lists

6.1- Numbered (Ordered) Lists

6.2- Bulleted (Unordered) Lists

6.3- Nested Lists

6.4- Definition Lists

Creating Lists

6.1- Numbered (Ordered) Lists:

You can insert lists into your HTML documents to display ordered lists of items. Numbered lists (sometimes called Ordered Lists) are generally used to list items by their priority or their sequence in a process. For example, you might use a numbered list to display instructions in completing a specific task.

When you create a numbered list, you must use two different tags. First, you place the <ol>…</ol> tags around the text you want to become a numbered list. Second, you place the <li>…</li> tags around each line of the list. You have five choices when creating numbered lists: Upper-case letters (A,B,C), lower-case letters (a,b,c), upper-case Roman numerals (I,II,III), lower-case Roman numerals (I,ii,iii) and regular numbers (1,2,3) which is the default. You define the type of numbered list you want by using the TYPE attribute and placing it within the <ol> start tag. You can also begin the numbering (ordering) at any point in the sequence as well using the START attribute. For example, if you wanted to start your list with item D, you would use the attribute start="D" by placing it within the <ol> start tag as well..93

Start Tag:	<ol>	
End Tag:	</ol>	
Attributes:	type="1" *(Default)*	List sequence uses 1,2,3, etc.
	type="A" *(Upper case alphabet)*	List sequence uses A,B,C, etc.
	type="a" (Lower case alphabet)	List sequence uses a,b,c, etc.
	type="I" *(Upper case Roman numerals)*	List sequence uses I,II,III, etc.
	type="i" *(Lower case Roman numerals)*	List sequence uses I,ii,iii, etc.
	start="?" *(Starting value)*	Where "?" is the starting value.
Related Tags:	<li>…</li> *(Line item)*	
Example:	My list: <ol type="A"> <li>First Item</li> <li>Second Item</li> <li>Third Item</li> </ol>	This tag will cause the items surrounded by the line item tags <li> to be displayed as an ordered list.
Result:	My list: A. First Item B. Second Item C. Third Item	

6.2- Bulleted (Unordered) Lists:

The other type of list you can create is called a bulleted (also called unordered) list. Bulleted lists are typically used when items require the reader's attention, but do not need to be listed in any particular order.

When you create a bulleted list, you must use two different tags. First, you place the <ul>…</ul> tags around the text you want to become a bulleted list. Second, you place the <li>…</li> tags around each line of the list. You have three choices when creating bulleted lists: Circles (○), squares (■) and discs (●) which is the default. You define the type of bulleted list you want by using the STYLE attribute and the value of "list-style-type:"?" (where ? Is circle, square or disc) placing it within the <ul> start tag.

Start Tag:	<ul>	
End Tag:	</ul>	
Attributes:	style="list-style-type:disc"	● List sequence uses filled circle
	style="list-style-type:circle"	○ List sequence uses hollow circle
	style="list-style-type:square"	■ List sequence uses filled square
Related Tags:	<li>…</li>	
Example:	My list: <ul style="list-style-type:square"> <li>First Item</li> <li>Second Item</li> <li>Third Item</li> </ul>	This tag will cause the items surrounded by the line item tags <li>…</li> to be displayed as a list with square bullets.
Result:	My List: ▪ First Item ▪ Second Item ▪ Third Item	

6.3- Nested Lists:

A nested list is simply a numbered or bulleted list that is contained within another list. This can come in handy when you wish to display items at different levels within a list hierarchy. You have the flexibility of creating both numbered or bulleted lists as nested lists.

To create a nested list, be sure to use one set of either <ul>…</ul> or <ol>…</ol> tags and any attributes for each list (see sections on “Numbered Lists’ and “Bulleted Lists”).

Start Tag:	<ul> or <ol>	
End Tag:	</ul> or </ol>	
Attributes:		Choose from bulleted or numbered attributes (see those sections)
Related Tags:	<li>…</li> (Line item)	
Example:	Our Most Popular Tutorials: <ul> <li>QuickBooks</li> <li>Peachtree</li> <li>Microsoft Office</li> <ul style=“list-style-type:circle”> <li>Access</li> <li>Excel</li> </ul> <li>Photoshop</li> </ul>	These tags will create a nested list (Microsoft Office Applications listed separately from the rest)
Result:	Our Most Popular Tutorials: • QuickBooks • Peachtree • Microsoft Office ○ Access ○ Excel • Photoshop	

6.4- Definition Lists:

Definition Lists are used to display text in a form that resembles a dictionary or glossary of terms. When you create a definition list, you must use three different tags. First, you place the <dl>…</dl> tags around the text you want to become a definition list. Second, you place the <dt>…</dt> tags around each term of the list. Third, you place the <dd>…</dd> tags around each definition.

Start Tag:	<dl>	
End Tag:	</dl>	
Attributes:	None	
Related Tags:	<dt>…</dt> *(Definition Term)* <dd>…</dd> *(Definition)*	
Example:	What do the applications do? <dl> <dt>Access</dt> <dd>Create databases and programs to track and manage your information</dd> <dt>Excel</dt> <dd>Perform calculations, analyze information and manage lists in spreadsheets</dd> </dl>	These tags will create a definition list.
Result:	What do the applications do? Access Create databases and programs to track and manage your information. Excel Perform calculations, analyze information and manage lists in spreadsheets.	

ACTIONS-
CREATING LISTS

ADD A NUMBERED (ORDERED) LIST:

1. At the point where you want to add an ordered list, type the first part of the start tag: <ol type=
2. Type the value for the type of ordered list you want, in quotes (“1”,”A’,”a”,”I”,”i”)
3. Close the tag by typing: >
4. Press “Enter”.
5. Type: <li>
6. Type the text for the first item in your list.
7. Type: </li>
8. Repeat steps 4 through 7 for each addition line item.
9. When done with list items, press “Enter”.
10. Type: </ol>

ADD A BULLETED (UNORDERED) LIST:

1. At the point where you want to add an unordered list, type: <ul style=“list-style-type:
2. Type the value for the type of ordered list you want, with closing quotes (disc”, circle”, or square”)
3. Close the tag by typing: >
4. Press “Enter”.
5. Type: <li>
6. Type the text for the first item in your list.
7. Type: </li>
8. Repeat steps 4 through 7 for each additional list item.
9. When done with list items, press “Enter”.
10. Type: </ul>

ADD A NESTED LIST:

1. To add a nested list, complete steps 1-10 in either of the examples above, and place all of the information within an existing list element.

ADD A DEFINITION LIST:

1. At the point where you want to add a definition list, type: <dl>
2. Press “Enter”.
3. Type: <dt>
4. Type the text from the first term in your list.
5. Type: </dt>
6. Press “Enter”.
7. Type: <dd>
8. Type the text for your definition.
9. Type: </dd>
10. Repeat steps 2 through 9 for each additional list item.
11. When done with list items, press “Enter”.
12. Type: </dl>

EXERCISES-
CREATING LISTS

Purpose:

1. To add an ordered list to the HTML document created in the previous chapters.

Exercises:

1. Open your HTML page created in the previous chapters.
2. On the line after the last text in the body, type: <p>
3. Press “Enter”.
4. Type: This is an ordered list:
5. Press “Enter”.
6. Type: <ol type=“A”>
7. Press “Enter”.
8. Type: <li>First Item</li>
9. Press “Enter”.
10. Type: <li>Second Item</li>
11. Press “Enter”.
12. Type: <li>Third Item</li>
13. Press “Enter”.
14. Type: </ol><p>
15. Press “Enter”.
16. Save Document

Chapter 7- Links

7.1- What are Links?:

One of the most dynamic features of HTML is its ability to display selectable *hyperlinks* (or *links* for short) which allow the user to navigate from one topic or page to the next. By clicking on a link, the user “jumps” to the desired location. Links most often take the form of underlined text, but you can also turn images (see “Images” chapter) into links. When a link is created and a mouse pointer is held over the image or text that is a link, the pointer turns into a pointing hand that indicates a link. You can link users to other pages on your site, other pages on the Web, to an e-mail address and even to other types of documents (such as Microsoft Word, or a PDF (Portable Document Format)).

Before you begin creating links, it is important to understand URLs. As mentioned earlier, each page on the Internet has a unique address, called a URL (Uniform Resource Locator). When you know a Web address, you generally use HTTP (Hypertext Transmission Protocol) to reach a specific address (http://www.teachucomp.com). HTTPS (Hypertext Transmission Protocol, Secure) is used when handling secure transactions that require information to be encrypted such as credit card information in a commerce transaction. FTP (File Transfer Protocol) is used to physically transfer files to a different computer. When you upload your webpages to a Web server, you use FTP.

The first part of a Web address is the HTTP prefix followed by a colon and two forward slashes (http://). Most of the time when you are surfing the Web, your browser will insert the HTTP prefix for you. However, when inserting Web addresses into your HTML code, you must include it as part of the full Web address. The second part is the host name, preceded by www. (for World Wide Web). The host name is also called a domain. For example, in the Web address http://www.yahoo.com, yahoo.com is the host (domain) name. If the page you are linking to is not the main page of the site, you need to specify its exact location and file name. For example, to send someone to TeahUcomp, Inc.’s course selection page, the URL would be http://www.teachucomp.com/courses/. Be careful when typing these Web addresses, as one wrong character will cause a “broken” (non-functioning) link. Always test your links before you publish your HTML page so that users have a positive experience.

There are two types of links you can create in HTML – absolute and relative. An absolute link generally refers to and is used to create a link that takes the user to a specific page on the Web outside of the current website. An absolute link contains a complete URL. For example: http://www.teachucomp.com. A relative link is commonly referred to as a “shorthand Web address”. You use relative links to refer to pages on the same website. For example, once someone is on the TeachUcomp, Inc. website at http://www.teachucomp.com, a relative link to take the visitor to our course section would be “/courses/”. When using a relative link it is generally a good idea to put a trailing slash (/) at the end of the Web address. If left off, it causes your browser to submit a request to the server twice, the first time requesting the address without the trailing slash and then again with the trailing slash automatically added by your browser.

7.2- Text Links:

The tag used to define a link is actually an anchor tag. It takes the format of <a href=“…”>, where the “a” defines it as an anchor tag and the HREF attribute defines the “Hyperlink Reference” or action that will occur when the user selects the link. The <a href=“…”> tag is followed by the text that will be displayed to the user for their selection. The display text, and the anchor tag, are then terminated with the </a> end tag. The following is an example of a typical ink tag: <a href=“teachucomp.com”>TeachUcomp, Inc. Homepage</a>

In this example, the user sees “TeachUcomp, Inc. Homepage” in blue underlined lettering as selectable text. Once the user selects “TeachUcomp, Inc. Homepage” with their mouse, they will be sent to the website www.teachucomp.com.

Start Tag:	<a>	
End Tag:	</a>	
Attributes:	href=	
Example:	<a href=“http://www.teachucomp.com”>Home</a>	This tag will create a text link to the specified webpage.
Result:	Home	The link appears as underlined text, including a clickable link.

7.3- Image Links:

You can also turn images (see next chapter “Images”) into links. This is accomplished in the same manner as text links, with the image coding replacing the text that gets turned into a link.

Start Tag:	<a>	
End Tag:	</a>	
Attributes:	href=	
Example:	<a href=“http://www.teachucomp.com”><img src="http://www.teachucomp.com/wp-content/uploads/hd-video-lessons.png”></a>	This tag will create an image link to the specified webpage.
Result:		The link appears as the specified picture. When a mouse pointer is held over the image, a pointing hand appears indicating a link.

7.4- Opening a Page in a New Window or Tab:

By default, when you link to another page, the new page opens in the existing browser window. There may be times when you wish to open the page in a new browser window or tab. A tab is considered a "new window" in the same browser. This is considered good practice when linking to pages outside of your own website so that your users have access to the information but do not actually leave your site. To set an individual link to open in a new browser window, use the TARGET attribute with a value of "_blank" and place it within the <a> start tag.

Start Tag:	<a>	
End Tag:	</a>	
Attributes:	href=	
Example:	<a href="http://www.teachucomp.com" target="_blank">Home</a>	This tag will create a text link to the specified webpage and open it in a new browser window or tab.
Result:	Home	The link activates a new browser window or tab.

7.5- Setting All Links on a Page to Open in a New Window:

You can also set all of the links on an HTML page to open in a new window or tab each time the link is clicked. To accomplish this, you must place <base target="_blank"> between the <head>…</head> tags.

Start Tag:	<base>	
End Tag:	None	
Attributes:	target=	
Example:	<head> <base target="_blank"> … </head>	
Result:	This tag will cause all links on the page to open in a new window or tab.	

7.6- Linking to an Area on the Same Page (Bookmarks):

Sometimes, you may want to create a link that takes the user to another area on the same page. Also called bookmarks, these links are helpful for especially long pages, or when you want to make navigation easy on the page and organize information for the user. For example, you may want to use bookmarks on a FAQ page, so that a user can easily get to their specific questions, and return to the top of the page quickly and easily.

Creating a bookmark is a two-step process. First, you must identify and name the spot on your page that you want to link to. Place the anchor tags <a>…</a> around the area you want to link to. Within the <a> start tag, place the ID attribute with a value that is the name you choose. Second, create a link to your newly named area, with a slight variation. For bookmarks, place a pound sign (#) immediately before the name of the anchor.

Start and End tags:	<a>…</a> *(to create the anchor)* <a href=“#?”>…</a> *(to create the link)*	
Attributes:	Id=“?” *(within the <a> start tag of the anchor)* href=“#?” *(within the <a> start tag of the link)*	
Example:	<p><a href=“#colors”>Learn More</a></p> <p><a id=“colors”></a>Available Colors:</p> <p>Red Blue Green</p>	
Result:	Learn More Available Colors: Red Blue Green	The link takes the user to the specified anchor. In this example, the user would jump to the line “Available Colors”.

7.7- Linking to an E-mail Address:

You can create links on your page to e-mail addresses. When the link is clicked, the user's default e-mail editor will open with the e-mail address you specified in the "To" address area. To create an e-mail link, you use mailto:? as the value, replacing ? with the e-mail address you want the message sent to.

You can also pre-populate the "Subject:" line of the user's e-mail message to you by inserting a question mark (?) and "subject=?" following the email address. This can be useful by helping you quickly identify or organize your e-mails. If you use a multi-word subject line it is good practice to place %20 between each word to ensure proper spacing of the words in the subject line.

Start Tag:	<a>	
End Tag:	</a>	
Attributes:	href=	
Example:	<p><a href=“mailto:you@yourdomain.com?subject=Comments%20and%20Questions”>e-mail us</a></p>	
Result:	To: you@yourdomain.com Cc: Subject: Comments and Questions	The user's e-mail editor addresses the e-mail to the specified address, with the pre-populated subject.

7.8- Linking to Other Types of Files:

You can create links in your page to all types of files other than HTML, such as a Microsoft Word file, a PDF document and so on. One important thing to keep in mind is that the user must have the correct program in order to properly open the file. If you provide a link to a Microsoft Excel file and the user does not have Microsoft Excel on their computer, they will encounter problems. For this reason, it is a good idea to tell your users what type of file it is. Depending upon the file type, the user will probably be prompted to download the file with a "File Download" dialog box, at which point they can open or save the file.

To create a link to other types of files, use the file name as the HREF value. ***Important:*** Make sure the file you are pointing to is uploaded to your server, so the user actually has access to it.

Start Tag:	<a>	
End Tag:	</a>	
Attributes:	href=	
Example:	<a href=“samples.pdf”>See a Sample</a>	
Result:	See a Sample	The link prompts the user to open the specified PDF file.

ACTIONS- LINKS

ADD A TEXT LINK:

1. At the point where you want to add a text link, type the tag: <a href="?">
   (where "?" is the path and file name of the destination)
2. Type the text that you want to display as the link (ex: Home Page)
3. Type the end tag: </a>

ADD AN IMAGE LINK:

1. At the point where you want to add an image link, type the tag: <a href="?">
   (where "?" is the path and file name of the destination)
2. Type the tag: <img src="?">
   (where "?" is the path and file name of the image you want to turn into a link)
3. Type:</a>

OPEN A PAGE IN A NEW WINDOW:

1. At the point where you want to add a link to open in a new window or tab, type the tag: <a href="?"
   (where "?" is the path and file name of the destination)
2. Type a space and then: target="_blank">
3. Type the text that you want to display as the link (or the image path) as in the previous exercise.
4. Type the end tag </a>

SET ALL LINKS ON A PAGE TO OPEN IN A NEW WINDOW:

1. Between the <head>…</head> tags of your page, type: <base target="_blank">

LINK TO AN AREA ON THE SAME PAGE (BOOKMARKS):

1. At the location on the page that you want to link to, type: <a id="?"></a>
   (where the "?" is the name you choose for your anchor)
2. Create the link in the location of your choice by typing: <a href="#?"></a>
   (where "#?" is the name of your anchor that you created in Step 1)

LINK TO AN E-MAIL ADDRESS:

1. At the point where you want to add the e-mail link, type: <a href="mailto:?">
   (where "?" is the e-mail address you want to receive the message)

LINK TO OTHER TYPES OF FILES:

1. At the point where you want to add the link, type the tag: <a href="?">
   (where "?" is the path and file name of file)

EXERCISES- LINKS

Purpose:

1. To add and change the properties of several links in the HTML page created in previous chapters.

Exercises:

1. Open your HTML page created in previous chapters.
2. On the line after the last text in the body, type:
   <a href=“http://www.google.com”>Google</a>
3. Press “Enter”.
4. Type: <p>Click on the image below to see our site.</p>
5. Press “Enter”.
6. Type: <a href=“http://www.teachucomp.com” target=“_blank”><img src="http://www.teachucomp.com/wp-content/uploads/2013/03/logo.png”></a>
7. Press “Enter”.
8. Type: <p><a href=“#top”>Top of page</a></p>
9. Press “Enter”.
10. Type: <p> <a href=“mailto:test@teachucomp.com”>Email Us</a></p>
11. At the top of the page, replace “Welcome to my Webpage”, with: <a id=“top”>Welcome to my Webpage</a>
12. Save Document.

CHAPTER 8- Images

8.1- Introduction to Images for Webpages:

Images are a terrific way to add interest to your webpage and come in different forms such as photographs and graphics. There are many different file types used for images in webpages, the most common being JPEG, GIF and PNG. It is important to understand the different file types and how browsers display them.

GIF (Graphics Interchange Format) is most commonly used for simple, less-detailed images, such as graphics, logos, etc. GIF supports 256 colors and is primarily used when your image has only a few colors and basic lines and shapes. GIF is also used to display short, simple animations. JPEG or JPG for short (Joint Photographic Experts Group) supports literally millions of colors (called 24 bit) and is the format used when the image is complex, such as photographs and other detail-rich images. PNG (Portable Network Graphics) combines the 24 bit support of JPG with advanced compression capabilities.

After quality, probably the most important consideration when inserting images into your webpage is download time. When a user visits your webpage, their browser actually downloads the images so they can be viewed. The more images there are and the larger the file sizes, the slower the page downloads. You will need to consider both the number of images in your webpage as well as the individual image sizes. It is generally considered best practice to keep individual image sizes at 100K and below. Your pages load time is also dependant on the user's connection speed. There are many different image and graphics editing programs on the market today to help edit your images. Corel PaintShop Pro and Adobe Photoshop Elements are two of the most popular and strongly recommended. They will allow you to optimize your images before you insert them into your HTML code. With the image editing program of your choice, you should crop the image to remove any extraneous parts, which helps reduce the file size. These programs will allow you to adjust the resolution of the image (the higher the resolution, the higher the file size and download time) and compression of the file size while you monitor quality.

8.2- Adding Images to Webpages:

To add an image to your webpage, use the <img> tag and specify the file name with the SRC (source) attribute. The value is the name of the image, including the file extension. If the image is hosted on a different server, you must list the full path of the image. For example, "http://www.teachucomp.com/images/sunset.jpg".

Start Tag:	<img>	
End Tag:	None	
Attributes:	src=	
Example:	<img src="sunset.jpg">	
Result:	Places the image "sunset.jpg" in the desired location on the webpage.	

8.3- Re-Sizing an Image:

You can adjust the size of your image in your HTML page by using the WIDTH and HEIGHT attributes. Use caution when re-sizing your image using HTML coding. If you don't adjust the dimensions appropriately, you can distort your image. When you apply the size attributes, you can assign a value in pixels, or as a percentage of the browser window size.

Start Tag:	<img>	
End Tag:	None	
Attributes:	src=	
	width=	Measured in pixels or as a percentage of the width of the window.
	height=	Measured in pixels or as a percentage of the height of the window.
Examples:	<img src="logo.gif" width="150" height="150">	Using pixels.
	<img src="logo.gif" width="40%" height="25%">	Using percentages
Result:	The image, "logo.gif" will appear in the dimensions indicated.	

8.4- Alternative Text:

Alternative text (also known as placeholder text) serves two purposes. First, it provides a description of the image. Some users turn images off in the browsers to increase download speeds. When this is done, any ALT text you have inserted appears in the image's place and gives the user an idea of what the image is. Users who have images turned on in their browser do not see the ALT text. Second, some search engines use ALT text when indexing pages and assigning rankings to search results. If part of your goal is to optimize your webpage for search engines, adding keywords into ALT text is a common technique. To add alternative text to your image, insert the ALT attribute into the <img> tag.

Start Tag:	<img>	
End Tag:	None	
Attributes:	alt=	
Example:	<img src="logo.gif" width="150" height="150" alt="TeachUcomp, Inc. Logo">	
Result:	The image, "logo.gif" will appear in the dimensions indicated. Users who have images turned off will see the ALT text.	

8.5- Image Labels:

You can add a label (or title) to your image so that when a user moves their mouse pointer over the image, the label will appear in a small box next to the pointer. This is achieved by placing the TITLE attribute within the <img> tag.

Start Tag:	<img>	
End Tag:	None	
Attributes:	title=	
Example:	<img src=“sunset.jpg” title=“Sunset on the Water”>	
Result:	When a user places their mouse pointer over the image “sunset.jpg”, the title will appear in a small box next to the pointer.	

ACTIONS- IMAGES

ADD AN IMAGE:

1. At the point where you want to add the image, type the tag: <img src="?">
   (where "?" is the path and file name of the image you want to insert)

RE-SIZE AN IMAGE:

1. At the point where you want to add the image, type the tag: <img src="?">
   (where "?" is the path and file name of the image you want to insert)
2. Type a space and then: width="X" height="Y">
   (where "X" and "Y" are the dimensions you want, in pixels or percentages)

ADD ALTERNATIVE (ALT) TEXT:

1. Within the <img> tag, type the attribute: alt="?"
   (where "?" is the alternative text you want to enter)

ADD AN IMAGE LABEL (TITLE):

1. Within the <img> tag, type the attribute: title="?"
   (where "?" is the title you want to enter)

EXERCISES- IMAGES

Purpose:

1. To add and change the properties of images in the HTML page created in previous chapters.

NOTE: Use the file location and path of your images, replace “sample.jpg” with your image, replace “Your Description” and “Your Title” with the description and title of your choice.

Exercises:

1. Open your HTML page created in previous chapters.
2. On the line after the last text in the body, type: <img src=“sample.jpg” alt=“Your Description” title=“Your Title”>
3. Press “Enter”.
4. Save Document.

CHAPTER 9- Basic Tables

Basic Tables

9.1- Inserting a Table:

A table is a container that allows you to present data in an orderly arrangement. A table starts with the basic structure of a square with four borders. Inside the table are intersecting *columns* (vertical) and *rows* (horizontal) which create *cells*. Each cell has four borders and can store text, images and other data. Table and cell borders can be made to be visible or invisible. It is now considered bad form to use tables to set the structure of your webpage and tables should therefore be used to list data only. The basic components of HTML tables are <table>…</table> (defines the table), <tr>…</tr> (defines a table row), and <td>…</td> (defines table data, or cell content).

Before you begin to add coding to create your first table, you should have an idea of what you want your end result to look like. Take out a piece of paper and a writing instrument and draw a rough sketch of your table. Lay out where you want to place your elements and data. This will save a great deal of time when you actually begin to enter the code.

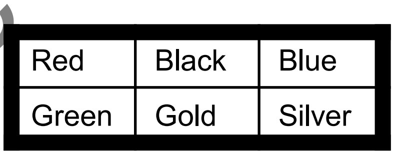
To add a table to your webpage, start by placing the <table> start tag where you want the table to appear. Next, you will need to define your table rows and the data contained in the cells. Keep your row and data tags on their own unique lines. This will make it easier to read your HTML code and to make future changes. Add a row by typing the <tr> start tag. On the next line, create your first cell by using the <td>…</td> tags, surrounding the data to be contained in the cell. Repeat the data tags on a new line for each cell in that row. When you are done with the row, insert the </tr> end tag. Repeat the process to add additional rows. When you are done with your table, finish with the </table> end tag.

Start Tag:	<table>	
End Tag:	</table>	
Associated Tags:	<tr>…</tr>	Defines rows.
	<td>…</td>	Defines data in cells.
Attributes:	None.	
Example:	<table> <tr> <td>Red</td> <td>Black</td> <td>Blue</td> </tr> <tr> <td>Green</td> <td>Gold</td> <td>Silver</td> </tr> </table>	
Result:	Red Black Blue Green Gold Silver	Creates a table with no visible borders.

9.2- Table Borders:

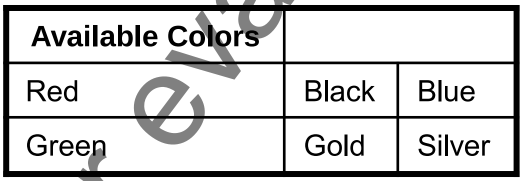
By default, tables that you create do not have visible borders (a line around the table that visually defines the table). Borders can help to make your table stand out more and adds visual interest. To assign a border, insert the BORDER attribute into the <table> start tag and assign a thickness to the table border, measured in pixels. The thickness will only be applied to the table border and not the individual cells. If you wish to change the color, add the BORDERCOLOR attribute to the <table> start tag and assign the color you want as the value. The border color will be applied to the table border, as well as the individual cell borders.

While you can style your table in your HTML coding, it’s a good idea to do most of your style selection in your cascading style sheet, which we will cover in a later chapter.

Start Tag:	<table>	
End Tag:	</table>	
Associated Tags:	<tr>…</tr>	
	<td>…</td>	
Attributes:	border=	Measured in pixels.
	bordercolor=	
Example:	<table border=“6px” bordercolor=“#000000”> <tr> <td>Red</td> <td>Black</td> <td>Blue</td> </tr> <tr> <td>Green</td> <td>Gold</td> <td>Silver</td> </tr> </table>	
Result:	 Red Black Blue Green Gold Silver	Creates a table with a 6 pixel thick border.

9.3- Table Headers:

You can add a header to the top of the table to help identify or label the contents of the table. Table headers appear at the top of the table, within their own cell and with text that is both bold and centered. To add a table header, use the <th>…</th> tags and place them after (not within) the <table> start tag.

Start Tag:	<th>	
End Tag:	</th>	
Attributes:	None.	
Example:	<table border="2" bordercolor="#000000"> <th>Available Colors</th> <tr> <td>Red</td> <td>Black</td> <td>Blue</td> </tr> <tr> <td>Green</td> <td>Gold</td> <td>Silver</td> </tr> </table>	
Result:	 **Available Colors** Red Black Blue Green Gold Silver	

ACTIONS- BASIC TABLES

INSERT A TABLE:

1. At the point where you want to add the table, type the start tag: <table>
2. On a new line, type the start tag: <tr>
3. On a new line, type the start tag: <td>
4. Type the information you want to contain in the cell.
5. Type the end tag: </td>
6. Repeat steps 3 through 5 for each cell until done with the row.
7. On a new line, type the end tag: </tr>
8. Repeat steps 2 through 7 for each row in your table until done.
9. On a new line, type the end tag: </table>

ADD A BORDER TO A TABLE:

1. Within the <table> start tag, type the attribute: border="X" bordercolor="Y"
   (where "X" is the thickness of the table measured in pixels and "Y" is the color you choose)

ADD A TABLE HEADER:

1. On the line after the start <table> tag, type the start tag: <th>
2. Enter the information you want as your header.
3. Type the end tag: </th>

EXERCISES- BASIC TABLES

Purpose:

1. To add a table with a border and a header in the HTML page created in earlier chapters.

Exercises:

1. Open your HTML page created in previous chapters.
2. On the line after the last text in the body, add a few lines by typing three lines of: <p>&nbsp;</p>
3. On a new line, type: <table border="5" bordercolor="#000000">
4. Press "Enter".
5. Type: <tr>
6. Press "Enter".
7. Type: <th>This is the Header</th>
8. Press "Enter".
9. Type: </tr>
10. Press "Enter".
11. Type: <tr>
12. Press "Enter".
13. Type: <td>Red</td>
14. Press "Enter".
15. Type: <td>Black</td>
16. Press "Enter".
17. Type: <td>Blue</td>
18. Press "Enter".
19. Type: </tr>
20. Press "Enter".
21. Type: <td>Green</td>
22. Press "Enter".
23. Type: <td>Gold</td>
24. Press "Enter".
25. Type: <td>Silver</td>
26. Press "Enter".
27. Type: </tr>
28. Press "Enter".
29. Type: </table>
30. Press "Enter".
31. Save Document.

CHAPTER 10
IFRAMES

10.1- What is an Iframe?:

An Iframe (Inline Frame) is a separate HTML document that is embedded into your HTML document. The most common usage of the Iframe element is to insert content from another webpage, such as a video or an advertisement. Iframes can be configured to act as a "mini" page on your page. For example, it can have an independent scrollbar that will not affect the base page it is embedded in.

10.2- Inserting an Iframe:

Inserting an Iframe into your webpage requires both the start and end tags <iframe>…</iframe> as well as the target source you are placing in the Iframe. To do so you use the attribute src=. Similar to inserting an image in your webpage.

Start Tag:	<iframe>	
End Tag:	</iframe>	
Attributes:	src=	
Example:	<iframe src="http://www.teachucomp.com"></iframe>	
Result:	Places the webpage www.teachucomp.com into your webpage.	

10.3- Setting Height and Width:

You can also set a specific height and width to your Iframe by placing the height= and width= attributes in the start tag. Values are defined either in pixels or as a percentage of the browser window.

Start Tag:	<iframe>	
End Tag:	</iframe>	
Attributes:	height=	Specifies the height of your Iframe in pixels or percentages.
	width=	Specifies the width of your Iframe in pixels or percentages
Example:	<iframe src="http://www.teachucomp.com" height="300" width="100%"></iframe>	
Result:	Displays the webpage in an Iframe 300 pixels high and as wide as the browser window.	

10.4- Using an Iframe for a Link Target:

You can set a link on your page to open inside an Iframe by using the TARGET attribute. To do so you must name your Iframe using the NAME attribute. The target and name must have the same value.

Start Tag:	<iframe>	
End Tag:	</iframe>	
Attributes:	target=	Sets the link to open in the specified Iframe.
	name=	Defines the name of the Iframe.
Example:	<iframe name="iframe_1"></iframe> <p><a href="http://www.teachucomp.com" target="iframe_1">teachUcomp, Inc.</a></p>	
Result:	Causes the link teachUcomp, Inc. to open inside your Iframe.	

ACTIONS- IFRAMES

INSERTING AN IFRAME:

1. At the point where you want to add the Iframe, type the start tag: <iframe src="?">
   (where "?" is the source page you'd like to display in the Iframe)
2. Type the end tag </iframe.>

SETTING HEIGHT AND WIDTH:

1. Within the <iframe> start tag, type the attributes: height="X" width="Y"
   (where "X" and "Y" are expressed in pixels or as a percentage of the browser window)

USING AN IFRAME FOR A LINK TARGET:

1. Within the <iframe> start tag, type the attribute: name="?"
   (where "?" is the name of your iframe)
2. Type the end tag: </iframe>
3. Type <p>
4. Type <a href="?" target="*iframe_name*">*Link Name*</a></p>
   (where "?" is the link, "*iframe_name"* is the name of the Iframe from step 1 (these values must match) and "*Link Name*" is the title of your link.

EXERCISES- IFRAMES

Purpose:

1. To place an Iframe into your webpage and set the width and height.
2. Set an Iframe to be the target for a link to open into.

Exercises:

1. Open your HTML page created in previous chapters.
2. On the line after the last text in the body, add a few lines by typing three lines of: <p>&nbsp;</p>
3. On a new line, type: <iframe src="http://www.teachucomp.com" width="300" height="300">
4. Press "Enter".
5. Type: </iframe>
6. Press "Enter".
7. Add a few lines by typing three lines of: <p>&nbsp;</p>
8. On a new line, type: <iframe name="iframe_1">
9. Press "Enter".
10. Type: <a href="http://www.teachucomp.com" target="iframe_1">teachUcomp, Inc. Home Page</a>
11. Press "Enter".
12. Save Document.

CHAPTER 11-
Forms

11.1- About Forms:

Forms are used to collect information from people who visit your website. For example, you can use forms to find out details about your visitors through surveys and feedback, or engage in e-commerce by selling your goods and services to people.

Forms are defined by the <form>…</form> tags and are made up of different *elements* to collect data, such as text boxes and radio buttons. Once the user inputs all of the information, they submit the form, using the "submit" button that you create. What happens with the data is a decision you will need to make. You can use a script to manage the data, sent the data to a database, or even receive the data via e-mail.

Most forms are processed using CGI scripts, CGI (Common Gateway Interface) is a script written in a language such as Java or Perl and runs on a Web server. Most Web servers accommodate the processing of CGI scripts, but you should check first with your Web host to make sure. In addition, you will want to find out the location of the server's CGI-bin (a directory where CGI scripting is stored), as this is where you will need to store the CGI script you create or use. If you know a language such as Perl, you can write your own script. There are also hundreds of free scripts available online that you can use, such as the ones at sites like The CGI Resource Index (http://www.cgi.resourceindex.com) and JavaScript Kit (http://www.javascriptkit.com). You will need to make the necessary changes to the CGI script that you use (script variables, path names, etc.) and upload the CGI script to your Web host server.

To begin creating a form using CGI script, start with the <form> tag, containing the command (method="post") and the ACTION attribute, with a value equal to the path and name of your CGI script.

Start Tag:	<form>	
End Tag:	</form>	
Attributes:	method="post" action="?"	Where "?" is the path and name of your CGI script for the form.
Example:	<form method="post" action="/cgi-bin/contact.pl">	
Result:	Creates the structure for the form.	

11.2- Sending to E-mail:

If your Web server does not support CGI scripting, or if you prefer to avoid it altogether, you can send the form data directly to an e-mail address. Keep in mind that this is not a good solution if you are capturing sensitive data such as credit card numbers, as it is not a secure form of transmission. Sending your form data to an e-mail address is a good solution if your form is simple.

To send the information to an e-mail address, you use the following coding:

Start Tag:	<form>	
End Tag:	</form>	
Attributes:	method="post" enctype="text/plain" action="mailto:?"	Where "?" is the e-mail address you with to send the form to.
Example:	<form method="post" enctype="text/plain" action="mailto:info@yourcompany.com"> </form>	
Result:	Creates a form that tells the Web browser to send the form data to the selected e-mail address.	

11.3- Text Boxes:

Text boxes are the most basic elements that forms use in the collection of data. Text boxes are typically used when the input requires a single line of text. To create a text box, you use the <input> element tag and TYPE attribute with a value of “text” and place it between the <form>…</form> tags. You must also specify a unique name for the text box using the NAME attribute.

By default, text boxes are 20 characters wide. You can change the width of the field that is displayed by using the SIZE attribute. You can also limit the number of characters the user can type into the text box by using the MAXLENGTH attribute.

Start Tag:	<input>	No end tag.
Attributes:	type=“text”	Identifies the element as a text box.
	name=	Required.
	size=	
	maxlength=	
Example:	<form method=“post” action=“cgi-bin/contact.pl”> Your Full Name: <input type=“text” name=“fullname” size=“50” maxlength=“45”> </form>	
Result:	Creates a text box called “fullname” that is 50 characters in length and holds a maximum of 45 characters of input.	

11.4- Text Areas:

Sometimes, you will want to collect text from individuals that requires a larger box. This is common in situations where you ask for feedback that may require multiple sentences. In this case, you can insert a large text area using the <textarea>…</textarea> element tags and assigning a unique (and required) name using the NAME attribute.

You control the dimensions of the text area using the COLS and ROWS attributes, measured in relation to the character height. You can also control how text wraps within the text area using the WRAP attribute. You have three choices when assigning a value to the WRAP attribute: “soft”, “hard”, and “off”. A value of “soft” will wrap the text in your area, but will not wrap text in the form results (meaning it will be in a single field in a database or a single line in an e-mail). The value “hard” wraps text in both the text area and form results. The value “off” turns text wrapping completely off and forces users to create new lines using the “Enter” key on their keyboard.

11.4- Text Areas- (cont.):

If the user types more characters than can be seen in the text area you created, scroll bars will appear to enable viewing of the text. Text areas hold up to 32,700 characters.

Start Tag:	<textarea>	
End Tag:	</textarea>	
Attributes:	type="text"	Identifies the element as a text box.
	name=	Required.
	rows= and cols=	
	wrap=	"soft", "hard", or "off"
Example:	<form method="post" action="/cgi-bin/contact.pl">
Your Comments: <textarea name="comments" rows="15" cols="75" wrap="hard"> </textarea> </form>	
Result:	Creates a text are called "comments" that is 15 rows high and 75 columns wide and forces a "hard" wrap, where text in the text area and form results are wrapped.	

11.5- Check Boxes:

Check boxes are used when you want visitors to select from one or more options that you present. To create check boxes, you use the <input> element tag and TYPE attribute with a value of “checkbox” and place it between the <form>…</form> tags. You can group the check boxes using the same NAME attribute for each element. Remember that the value you assign is a description of the checkbox. If you want your check boxes to all appear on a separate line, use the <p> or <br> tags before each element.

Start Tag:	<input>	No end tag.
Attributes:	type=“checkbox”	Identifies the check box.
	name=	Required.
	value=	
Example:	<form method=“post” action=“/cgi-bin/contact.pl”><br><br>Have you purchased from us before?<br><br><input type=“checkbox” name=“purchase” value=“yes”>Yes<br><br><input type=“checkbox” name=“purchase” value=“no”>No </form>	
Result:	Have you purchased from us before? □ Yes □ No	

11.6- Menu Lists:

Menu lists are typically used when you have a long list of choices to give users. The menu list appears as a drop-down list and allows the user to select their choice. Between the <form>…</form> tags, use the <select> element tag, along with the NAME attribute and define a size for the box with the SIZE attribute (measured in character lines). The <option> element tag is then used to define each of the choices in the menu.

Start Tag:	<select>	
End Tag:	</select>	
Attributes:	name=	Required.
	size=	Measured in character lines.
Supporting Start tag:	<option>	No end tag.
Attributes	value=	Name of selection.
Example:	<form method="post" action="/cgi-bin/contact.pl"> Which best describes your status? <select name="status" size="1"> <option value="Business">Business <option value="Government">Government <option value="Individual">Individual </select> </form>	
Result:	Which best describes your status? **Business** **V**	Creates a drop=down with the list of choices specified.

11.7- Radio Buttons:

Radio buttons are the small circles (O) that appear in forms. Radio buttons allow you to present a series of choices (grouped under the same NAME attribute) but only allows the user to select a single choice. To add radio buttons, use the <input> element tag between the <form>…</form> tags with a TYPE value of "radio". The <p> or <br> tags allow you to position each choice on its own line.

Start Tag:	<input>	No end tag.
Attributes:	type="radio"	
	name=	Required.
	value=	
Example:	<form method="post" action="/cgi-bin/contact.pl"><br><br>How did you hear about us?<br><br><input type="radio" name="source" value="tvradio">TV, Radio<br><br><input type="radio" name="source" value="print">Print Ad<br><br><input type="radio" name="source" value="internet">Internet Search </form>	
Result:	How did you hear about us? o TV, Radio o Print Ad o Internet Search	

11.8- Submit Button:

The submit button is a required piece of your form since when it is clicked, it actually sends the data to be processed. You **must** have a submit button in your form in order to receive the data. Using the <input> element tag and a value of "submit" for the TYPE attribute, assign a VALUE that is the text you want to appear in the button (typically "Submit").

Start Tag:	<input>	No end tag.
Attributes:	type="submit"	
	value=	Text in button.
Example:	<form method="post" action="/cgi-bin/contact.pl"<br><br><input type="submit" value="Submit"> </form>	
Result:	Submit	

11.9- Reset Button:

The reset button is used to allow a user to clear **all** of the information they have entered into a form. To insert a reset button, use the <input> element tag with a TYPE attribute and a value of "reset", and the VALUE attribute with a value that is the text you want to appear in the button (typically "Reset").

Start Tag:	<input>	No end tag.
Attributes:	type="reset"	
	value=	Text in button
Example:	<form method="post" action="/cgi-bin/contact.pl"> <input type="reset" value="Reset"> </form>	
Result:	Reset	Clears all form data.

11.10- Changing the Tab Order:

Users can navigate through the elements of your form by using the Tab key on their keyboard, jumping from one element to the next. By default, the order the Tab key follows is the order in which you entered the form elements in your HTML page. To change the tab order, add the TABINDEX attribute to your element tags and assign a numeric value for the position you want ("1" for 1st, "2" for 2nd and so on.)

Start Tag:	Any element tag in your form.	
Attributes:	tabindex=	
Example:	<form method="post" action="/cgi-bin/contact.pl"> How did you hear about us? <input type="radio" name="source" value="tvradio" tabindex="1">TV, Radio <input type="radio" name="source" value="print" tabindex="2">Print Ad <input type="radio" name="source" value="internet" tabindex="3">Internet Search </form>	
Result:	How did you hear about us? o TV, Radio o Print Ad o Internet Search	Tab order follows your specifications.

ACTIONS-
FORMS

CREATE A FORM THAT USES A CGI SCRIPT:

1. At the point where you want to insert the form, type the first part of the start tag: <form
2. Type a space and then the attributes: method="post" action="?"
   (where "?" is the path and name of you CGI script for the form.)
3. Close the tag by typing: >
4. On the next line, type the end tag: </form>

CREATE A FORM THAT SENDS INFORMATION TO AN E-MAIL ADDRESS:

1. At the point where you want to insert the form, type the first part of the start tag: <form
2. Type a space and then the attributes: method="post" enctype="text/plain" action=mailto:"?"
   (where "?" is the e-mail address to receive the form data)
3. Close the tag by typing: >
4. On the next line, type the end tag: </form>

ADD A TEXT BOX TO A FORM:

1. Between the <form>…</form> tags, type: <input type="text" name="?">
   (where "?" is the required unique name for the text box)

ADD A TEXT AREA TO A FORM:

1. At the point in the form where you want to add a text area, type the first part of the start tag: <textarea name="?"
   (where "?" is the require unique name for the text area.
2. Close the tag by typing: >
3. On the next line, type the end tag: </textarea>

ADD CHECK BOXES TO A FORM:

1. At the point in the form where you want to add a check box, type: <input type="checkbox" name="X">Y
   (where "X" is the required name for the checkbox and "Y" is the text to appear next to the check box.

ADD A MENU LIST TO A FORM:

1. At the point in the form where you want to add a menu list, type the start tag: <select name="?" size="X">
   (where "?" is the required unique name for the menu list and "X" is the size of your menu list box, in character lines)
2. On the next line, identify the first choice by typing the element tag: <option value="X">Y
   (where "X" is the value of the option and "Y" is the text to appear as the choice)
3. Repeat steps 1-2 for each choice until finished.
4. On the last line, type the end tag: </select>

ACTIONS- FORMS

ADD RADIO BUTTONS TO A FORM:

1. At the point in the form that you want to add radio buttons, type: <input type="radio" name="X">Y (where "X" is the required name for the radio button and "Y" is the text to appear next to the radio button)

ADD A (REQUIRED) SUBMIT BUTTON TO A FORM:

1. At the point in the form where you want to add the submit button, type <input type="submit" value ="?"> (where "?" is the text to appear on the submit button)

ADD A RESET BUTTON TO A FORM:

1. At the point in the form where you want to add the reset button, type: <input type="reset" value="?"> (where the "?" is the text to appear on the reset button)

CHANGE THE TAB ORDER:

1. Within each element tag, type the attribute: tabindex="?"
   (where "?" is the numerical value of the element's placement in the tab order)

EXERCISES- FORMS

Purpose:

1. To create a basic form with different elements and send the form to an e-mail address.

Exercises:

1. At the point in your HTML page where you want to enter a form, type: <form name="practice" method="post" action="mailto:youremail@yourdomain.com">
2. Press "Enter".
3. Type: <h2>Please fill in the following information:</h2>
4. Press "Enter".
5. Type: <p>Your Name<input type="text" name="Name"></p>
6. Press "Enter".
7. Type:<p>Tell us about yourself: <textarea name="About"></textarea></p>
8. Press "Enter".
9. Type: <p>Have you purchased from us before?</p>
10. Press "Enter".
11. Type: <p><input type="checkbox" name="yes" value="yes">Yes</p>
12. Press "Enter".
13. Type: <input type="checkbox" name="no" value="no">No</p>
14. Press "Enter".
15. Type: <p>Which best describes your status?
16. Press "Enter".
17. Type: <select name="select">
18. Press "Enter".
19. Type: <option value="Business">Business</option>
20. Press "Enter".
21. Type: <option value="Government">Government</option>
22. Press "Enter".
23. Type: <option value="Individual">Individual</option>
24. Press "Enter".
25. Type: </select></p>
26. Press "Enter".
27. Type: <p>How did you hear about us?</p>
28. Press "Enter".
29. Type: <p><input type="radio" name="TV Radio" value="TV Radio">TV, Radio
30. Press "Enter".
31. Type: <input type="radio" name="Print" value="Print">Print Ad
32. Press "Enter".
33. Type: <input type="radio" name="Internet" value="Internet">Internet Search</p>
34. Press "Enter".
35. Type: <p><input type="submit" name="Submit" value="Submit">
36. Press "Enter".
37. Type: <input type="reset" name="reset" value="Reset"></p>
38. Press "Enter".
39. Type: </form>
40. Save Document.

CHAPTER 12-
Video and Audio

Video and Audio

12.1- About Video and Audio Files:

You can add video and audio files to your webpage by either providing links to the files or embedding them directly into your HTML pages. Before the adoption of HTML5, there was no standard for showing video across multiple browsers. Videos could only be played with a “plug-in” like Adobe Flash, however different browsers used different plug-ins. HTML5 offers a new element, the <video> element, that does not require a plug-in to play video files. The <video> element is supported by most of the standard browsers used today.

The same can be said for audio files with the new <audio> element. Most standard browsers now support the <audio> element. Negating the need for a plug-in with most browsers.

Even though modern browsers support the <video> and <audio> elements it is a good idea to have fallback measures when users are trying to view your webpage from older browsers. The standard fallback for audio and video files is using the <embed> element in conjunction with the <audio> and <video> elements.

The easiest way to display video, without having to worry about browser support, is to use YouTube to view your video. You can upload your videos to YouTube and use that as your source to display your videos. The benefit to this is twofold, you save space on your server and YouTube uses a non-browser specific standard to display your videos.

12.2- Linking to Video and Audio Files:

The most basic way to provide audio and video files to your visitors is to link directly to the files. When clicked, the file will open and play in a separate window. Just as with other links, be sure to upload the video or audio file to your Web server to avoid broken links. The tag used to define the link is the anchor tag and takes the format of <a href=…>, where the “a” defines it as an anchor tag and the HREF attribute defines the “Hyperlink Reference” or action that will occur when the user selects the link. The <a href=…> tag is followed by the text that will be displayed to the user for their selection. The display text, and the anchor tag, are then terminated with the </a> end tag. If you choose to use this method, your visitor must be using a browser that supports the linked file type.

Start Tag:	<a>	
End Tag:	</a>	
Attributes:	href=	
Example:	<a href=“http://www.teachucomp.com/commercial.mp3” >Our Commercial </a>	This tag will create a text link to the specified .mp3 audio file.
Result:	Our Commercial	This link appears as underlined text, indicating a clickable link.

12.3- Adding Video:

There are a few steps necessary when adding or embedding video into your webpage. In addition to the <video> element, you will need to supply a <source> element. The <source> element in conjunction with the SRC attribute, tells your webpage what video file to display. You must also tell your browser the type of file you are linking with the TYPE attribute. There are currently three major video file types supported. They are: MP4, WebM and Ogg. The MP4 format is the most widely supported format with the major browsers (Internet Explorer, Chrome, Firefox and Safari). It may be a good idea to add multiple file types to your HTML coding to ensure compatibility across many browser platforms. You will be able to tell what type of file you are working with by checking either the extension at the end of your file, or looking at your file properties. You can use multiple <source> elements within your <video>…</video> element to link different video files and types. The browser your user is implementing will use the first recognized format of your video file.

It is important to include the HEIGHT and WIDTH attributes in the start tag <video>. When you include these attributes, the space for the video is reserved when your page is loading. If you don't include the height and width of your video, your page layout will change while the video is loading.

The CONTROL attribute is used to add controls (ex. play, pause, volume) to your video. There is no value attached to this attribute as it will just add basic controls to your video file.

To increase the likelihood of your video playing in your users' browser, you can combine the <video> element with the <embed> element. The <embed> element creates a container for any video file you choose to link to your webpage.

It is a good idea to include some text between the <video>…</video> tags to describe your video file or to let the user know their browser does not support the <video> tag. Any text you enter will only be displayed if the <video> element is not supported by their browser.

12.3- Adding Video- (cont.):

Start Tag:	<video>	
End Tag:	</video>	
Attributes:	height=	Sets height of video.
	width=	Sets width of video.
	controls	Adds video controls play, pause, volume, etc. to your file.
Supporting Start Tag:	<source>	No end tag.
Attributes:	src=	Defines link to video file.
	type=	Tells browser what type of file you are using.
Supporting Start Tag:	<embed>	No end tag.
Example:	<video width=“350" height=“260" controls> <source src=“samplevideo.mp4" type="video/mp4"> <source src=“samplevideo.ogg" type="video/ogg"> <embed src=“samplevideo.avi" width="320" height="240"> Sample Video. Your browser does not support this file type. </video>	
Result:	Displays a video, with play, pause and volume controls, in a space 240 pixels high by 320 pixels wide. If the browser does not support the video the text you entered will be displayed: Sample Video. Your browser does not support this file type.	

12.4- Adding Audio:

There are a few steps necessary when adding or embedding audio into your webpage. In addition to the <audio> element, you will need to supply a <source> element. The <source> element in conjunction with the SRC attribute, tells your webpage what audio file to play. You must also tell your browser the type of file you are linking with the TYPE attribute. There are currently three major audio file types supported. They are: MP3, Wav and Ogg. It is important to add at least two versions of your file, to increase the likelihood of your users browser being able to play your audio. You will be able to tell what type of file you are working with by checking either the extension at the end of your file, or looking at your file properties. You can use multiple <source> elements within your <audio>…</audio> element to link different files and types. The browser your user is implementing will use the first recognized format of your audio file.

The CONTROLS attribute is used to add controls to your audio like volume, play and pause. There is no value attached to this attribute and it will add the basic controls to your audio, allowing the user to play your file.

Just as with videos, you can combine the <audio> element with the <embed> element. The <embed> element creates a container for any audio file you choose to link to your webpage.

It is a good idea to include some text between the <audio>…</audio> tags to describe your audio file in case the browser doesn't support that file type. Any text you enter will only be displayed if the file type is not supported by the browser.

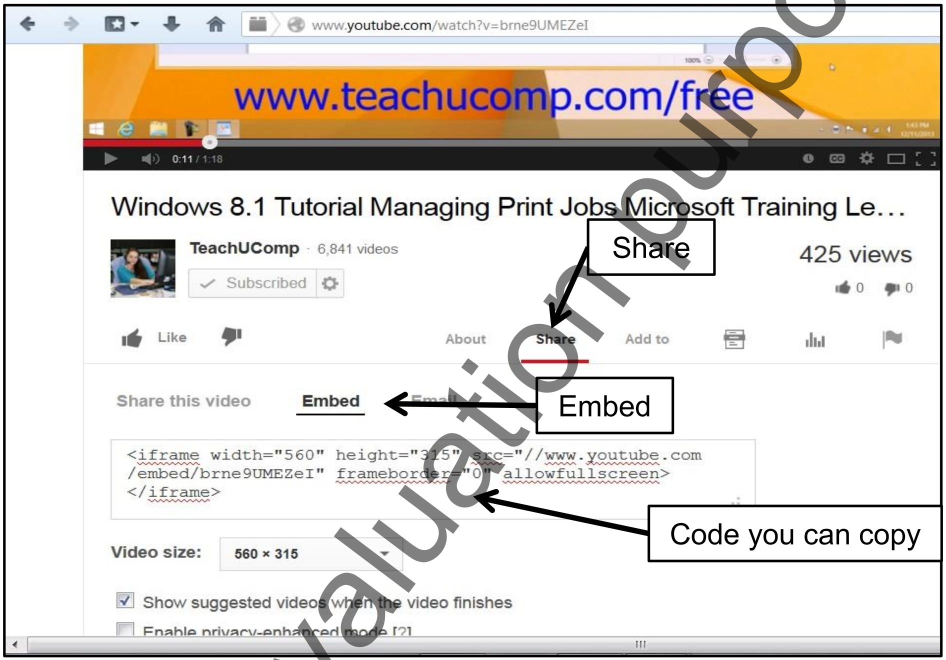
Start Tag:/	<audio>	
End Tag:	</audio>	
Attributes:	controls	Adds controls to your file.
Supporting Start Tag:	<source>	No end tag.
Attributes:	src=	Defines link to the audio file.
	type=	Tells browser what type of file you are using.
Supporting Start Tag:	<embed>	No end tag.
Example:	<audio controls> <source src="sample.mp3" type="audio/mpeg"> <source src="sample.ogg" type="audio/ogg"> <embed height="50" width="100" src="sample.mp3"> File Not Supported. </audio>	
Result:	Displays your audio file with controls for your user to play the file. Displays any entered text if file is not supported.	

12.5- Using YouTube to Display Video:

The easiest way to display videos on your webpage is using a link to YouTube. You can upload your own videos to a YouTube account or use any other video from YouTube. This method is easy because you don't have to worry about file types and browser support.

To use a YouTube video, you just create an Iframe using <iframe>…</iframe> (Chapter 10) and linking your YouTube video as your SRC attribute.

When linking a YouTube video you need to make sure under "Share" you choose the "Embed" option. This will actually give you the proper coding you need, without having to write it yourself.



www.youtube.com/watch?v=brne9UMEZeI

www.teachucomp.com/free

0:11 / 1:18

Windows 8.1 Tutorial Managing Print Jobs Microsoft Training Le…

TeachUComp · 6,841 videos

Subscribed

425 views

0 0

Like About Share Add to

Share this video Embed Email

```
<iframe width="560" height="315" src="//www.youtube.com
/embed/brne9UMEZeI" frameborder="0" allowfullscreen>
</iframe>
```

Video size: 560 × 315

☑ Show suggested videos when the video finishes

☐ Enable privacy-enhanced mode [?]

Start Tag:	<iframe>	
End Tag:	</iframe>	
Attributes:	src=	
Example:	<iframe width="560" height="315" src="//www.youtube.com/embed/brne9UMEZeI" frameborder="0" allowfullscreen></iframe>	
Result:	Displays the video in an Iframe 560 pixels wide by 315 pixels high.	

ACTIONS- Video and Audio

LINK TO A VIDEO OR AUDIO FILE:

1. At the point where you want to add the link, type the tag: <a href="?">Y</a>
   (where "?" is the path and file name of the file and "Y" is the text for your link)

ADDING VIDEO:

1. At the point where you want your video placed, type the start tag: <video
2. Add the width and height attributes by typing: width="X" height="Y"
   (where "X" and "Y" are the values in pixels)
3. Add controls to your video by typing: controls>
4. To add you source file, type: <source src="?" type="video/mp4"> **OR** type="video/webm"> **OR** type="video/ogg">
   (where "?" is the path and file name of the video file)
5. To add an embedded video for older browsers, type: <embed src="?">
   (where "?" is the path and file name of the file)
6. To add text for unsupported files, after the last line, type your chosen text.
7. On the last line, type the end tag: </video>

ADDING AUDIO:

1. At the point where you want your audio placed, type the start tag: <audio
2. Add controls to your video by typing: controls>
3. To add you source file, type: <source src="?" type="audio/mp3"> **OR** type="audio/wav"> **OR** type="audio/ogg">
   (where "?" is the path and file name of the audio file)
5. To add an embedded audio for older browsers, type: <embed src="?" width="X" height="Y">
   (where "?" is the path and file name of the file and "X", "Y" are the values in pixels)
6. To add text for unsupported files, after the last line, type your chosen text.
7. On the last line, type the end tag: </audio>

USING YOUTUBE TO DISPLAY VIDEO:

1. At the point where you want to add a YouTube video, select the code from the "Share"|"Embed" section of the YouTube video page and paste it into your HTML document.

EXERCISES-
VIDEO AND AUDIO

Purpose:

1. To be add video and audio to your webpage.

NOTE: All video or audio files should be chosen from an existing file on your computer for this exercise with the exception of the YouTube video file.

Exercises:

1. Open your HTML page created in previous chapters.
2. On the line after the last text in the body, type: <a href=“http://www.yoursamplevideo.com/video.mp4”>Sample Video</a>
3. Press “Enter”.
4. Type: <p>
5. Press “Enter”.
6. Type: <video width=“350" height=“260" controls>
7. Press “Enter”.
8. Type: <source src=“samplevideo.mp4" type="video/mp4">
9. Press “Enter”.
10. Type: <embed src=“samplevideo.avi" width="320" height="240">
11. Type: Sample Video. Your browser does not support this file type.
12. Press “Enter”.
13. Type: </video></p>
14. Type: <p>
15. Type: <audio controls>
16. Press “Enter”.
17. Type: <source src=“sample.mp3” type=“audio/mpeg”>
18. Press “Enter”.
19. Type: <embed height=“50” width=“100” src=sample.mp3”>
20. Type: File Not Supported.
21. Press “Enter”.
22. Type: </audio></p>
23. Press “Enter”.
24. Type: <p>
25. Press “Enter”.
26. Paste your selected code from the chosen YouTube page Share|Embed section.
27. Press “Enter”.
28. Type: </p>
29. Press “Enter”.
30. Save Document.

CHAPTER 13- Troubleshooting

13.1- Troubleshooting

13.1- Troubleshooting:

If you have ever visited a website that does no function properly, you can relate to the frustration that a user experiences when they run into problems. It is critical that you test your website to avoid such problems. By taking the time to double-check your work, you can avoid the pitfalls associated with common mistakes such as:

1. **Typos.** Like any other programming language, HTML must be exact in order to work correctly. A single mistyped or missing character can reek havoc in your webpages. Always proofread your documents and check for missing end tags or other elements.
2. **Broken Links.** If a user clicks on a link and it doesn't take them where you intend, it is considered a "broken link". When you test your webpages, be sure to click on every link to make sure it works. If it doesn't, verify that the path and file name are correct and that the file the link points to is actually uploaded to your Web server.
3. **Missing Images.** The notorious red "x" that shows in place of an image is also a common error. Check that the path and file name of the image is exactly correct, including the file extension (.jpg, .gif, etc.). You should also check that the image has been uploaded to your Web server.

If all else fails, and you cannot determine the problem, ask another experienced Web developer to take a look at your code. Sometimes, a fresh pair of eyes will catch a mistake you may have missed. And remember, as discussed at the beginning of this manual, there are many great resources available to you on the Internet. Examining other developers' HTML pages can provide you with fresh ideas and inspiration in your own design endeavors.

CHAPTER 14- Cascading Style Sheets

14.1- What are Cascading Style Sheets (CSS)?:

Cascading Style Sheets (CSS) are separate text files you create that contain rules and definitions which you can apply to other parts of your website. Once the style sheet is created, you link it to your HTML documents, allowing you to control the look and feel of several pages by changing a single source. Managing the formatting of your website, in a single place, maintains a cohesive design and allows for more control over individual elements in your website. Cascading Style Sheets can be external (meaning a text file separate from your HTML documents), internal (style code contained within an HTML page) or inline (style code contained in a line of HTML code).

It is important to remember when you are using multiple style guidelines that they will cascade in a certain order, with each maintaining any additional formatting. In the order of recognized importance:

1. Inline Style
2. Internal Style
3. External Style Sheet
4. Browser Default

14.2- CSS Syntax:

The syntax of CSS looks very different from HTML. Each rule within CSS has two components: Selectors and Declarations. The Selector is the element to which the rule applies and is followed by a space. Then, the declaration (surrounded by curly brackets) identifies the formatting to apply and consists of a property and a value. CSS Declarations **always** end with a semicolon. You can assign multiple declarations to a single selector by separating the declarations with a semicolon. The rule **h1 {font-weight: bold; color: maroon;}** sets all level 1 Headings to appear with maroon bold text. It is important to note that declarations apply not only to the specified element, but also any element that you place within it. For example, if you create a rule that applies to the <body> element, any elements you place within it inherit the same formatting properties.

14.3- Creating an External CSS:

To create an external Cascading Style Sheet, start a new document in your text editor and type your rules, keeping one rule to each line for future editing. Once you have listed your rules, save the document as a text file and assign the extension .css. We will discuss how to write specific rules in the next chapter. Once you create the CSS, you will need to link it to your HTML pages.

Start:	Your text editor.	
Type:	Your CSS rules.	
Save:	The document as a text file with a .css extension	

Example:	h1 {color: maroon; font-style: italic;}
	h2 {color: yellow;}
		Text file, saved as teachstyles.css

Result:	Creates a CSS with the rule defining all level 1 headings as maroon and italicized and all level 2 headings as yellow.	

14.4- Linking to a Cascading Style Sheet:

Once the CSS is created, you will need to link it to each HTML page to which you wish to apply the formatting. Between the <head>…</head> tags, you will need to add the following code: **<link rel="stylesheet" type="text/css" href="?">**, with the ? Replaced by the name of your style sheet.

Start tag:	<head>	
End tag:	</head>	
Attributes:	href=	
Example:	<head> <link rel="stylesheet" type="text/css" href="teachstyles.css"> </head>	
Result:	Links the style sheet to the HTML page.	

14.5- Adding Comments and Notes to a CSS:

Just as you can in your HTML pages, you can also insert comments into your Cascading Style Sheets as a form of documentation or as notes to yourself or other designers viewing your code. Comments only appear within the CSS code and are not displayed in the Web browser.

To add comments to your CSS, start your notes with a forward slash and an asterisk (/*) and end the notes with the reverse, an asterisk followed by a forward slash (*/).

Start:	/*	
End:	*/	
Example:	h1 {color: #660033; font-style: italic;} /*logo color*/	
Result:	The notes identifying the code as the color of the logo will not appear.	

14.6- Creating an Internal Style Sheet:

Style sheets can also be internal, meaning the rules are contained within the HTML page instead of a separate text file. This can be useful if your website consists of a single, long page with many elements you wish to control from a single location.

Internal style sheets are stored between the <head>…</head> tags. Because the coding is placed within your HTML page, you must use the <style>…</style> tags around the rules of the CSS. No attributes are required. Simply type the rules in the same manner as external style sheets.

Start tag:	<style>	
End tag:	</style>	
Attributes:	None.	
Example:	<head> <style> h1 {color: maroon; font-style: italic;} h2 {color: yellow;} </style> </head>	
Result:	Creates an *internal* CSS with the rule defining all level 1 headings displayed as maroon and italicized and all level 2 headings displayed in yellow.	

14.7- ID and Class:

There are two different ways to change the formatting on different elements in your HTML page. They are ID and Class. The ID selector is used when you want to change the style for a single, unique element. The Class selector is used when you want to change a group of elements within your HTML page.

The ID selector uses the ID attribute of the HTML element. It is defined with the pound symbol (#). For example, if you have a heading with the ID attribute of “heading1” in your HTML coding, the style coding would be labeled “#heading1”. This will style only the tag labeled with “heading1” and will leave all other headings as previously formatted. This is useful when you want to draw extra attention to a specific heading on your webpage.

You can create classes in both internal and external style sheets when you want to apply different style rules to specific sets of tags. For example, you want certain paragraphs to appear in a specific font and color and other paragraphs to have a different set of traits. You could create two classes within your rules that apply to both types of paragraphs.

To create a class in your CSS, first define the class by typing the tag you wish to control, followed by a period and a name you choose for the class. Follow the name with the declarations surrounded by curly brackets. Once the class is created, assign the CLASS attribute to the appropriate start tags in your HTML document, with the values as the class name.

With both ID and Class, it is very important that the labels match. For example, your HTML document lists an ID as “id=heading1” your CSS document must be “heading1”. If you list it as “Heading1”, your code will not validate as they are different. In this instance capitalization is very important.

ID:	#heading1 {font-family: “Times New Roman”; color: blue;}	Created in CSS file.
Attribute:	Id=	Assigned in HTML page.
Example:	<h3 id=“heading1”>This is my first Heading.> </h3> <h6>This is my second Heading.> </h6>	
Result:	This is my First Heading. This is my second Heading.	

Class:	p.Introductions {font-family: georgia; color: red;}
	p.Speech {font-family: verdana; color: blue:}
		Created in CSS file.

Attribute:	class=	Assigned in HTML page.
Example:	<p class=“Introductions”>Introduction text here.</p> <p class=“Speech”>Speech text here.</p>	
Result:	Introduction text here. Speech text here.	

14.8- Inline Styling:

Inline styling takes your CSS coding and inserts it into the tag of your HTML code. To do this you use the STYLE attribute inside the tag you want to work with. It is a good idea to keep inline styling to a minimum as you lose much of the advantage of style sheets.

Start Tag:	Whichever tag you decide to style. <p>	Created in your HTML document.
End Tag:	Whichever tag you decide to style. </p>	
Attribute:	style=	In quotations with CSS coding.
Example:	<p style="font-weight: bold; color: green;"> </p>	
Result:	The styled paragraph will have bold text and be green.	

ACTIONS-
CASCADING STYLE SHEETS

CREATE AN EXTERNAL CSS:

1. In your text editor, start a new document and type each rule on a unique line. Ex: h1 {color: maroon;}
2. Save the file as a text file with a .css extension.

LINK TO A CSS:

1. In the HTML page you want to link your CSS to, between the <head>…</head> tags, type: <link rel="stylesheet" type="text/css" href="?"
   (where "?" is the name of the style sheet you created)

ADD COMMENTS OR NOTES TO A CSS:

1. In your text editor, within your CSS, type: /*
2. Type your comments or notes.
3. Type: */

CREATE AN INTERNAL STYLE SHEET:

1. In your HTML document, between the <head>…</head> tags, type the start tag: <style>
2. On each new line, type the rule in the same format as an external style sheet.
3. On the line after the last rule, type the end tag: </style>

USING ID AND CLASSES:

1. In your text editor, within your HTML document label the tag you want to assign using the id= attribute: <h3 id="heading1">
2. In your text editor, within your CSS, type: #heading1 {font-family: arial; color: blue;}
3. In your text editor, within your HTML document, label the tag you want to assign using the class= attribute: <p class="introductions">
4. In your text editor, within your CSS, type: p.introductions {font-family: georgia; color: red;}

USING INLINE STYLING:

1. In your HTML document, type: <p style="font-weight: bold; color: green;"> </p>

EXERCISES-
CASCADING STYLE SHEETS

Purpose:

1. Covered at the end of the next chapter.

Exercises:

1. Covered at the end of the next chapter.

CHAPTER 15- Working with Text in CSS

15.1- Emphasizing Text (Bold and Italic):

To bold text in a CSS style rule, use the FONT-WEIGHT property. You can use the default value of “bold” or you can assign a degree of boldness with a numerical value in multiples of 100, with 100 being the lightest and 900 being the darkest bold.

To italicize text in a CSS style rule, use the FONT-STYLE property. There are three values that can be attributed to the FONT-STYLE property: italic, oblique and normal. “Italic” defines the italic version of the assigned font. If the font you are using does not have an italic version, you can use “oblique” to tell the browser to make an attempt to slant the font to mimic italics. If the element has inherited italics from a previous element and you want to remove the italics, use the value “normal”.

Property:	font-weight	Used to bold text
Value:	bold or 100, 200…900	
Example:	p {font-weight: 300;}	
Result:	Sets the CSS rule to apply boldness of 300 to the text of paragraphs.	
Property:	font-style	Used to italicize text
Value:	Italic, oblique *or* normal	
Example:	p {font-style: italic;}	
Result:	Sets the CSS rule to italicize the text of paragraphs.	

15.2- Decoration:

The TEXT-DECORATION property is used to add strike-through marks, underline and overstrike marks to your text. Generally this property is used to remove underlines from links. For example, a link in the middle of a sentence that is already underlined, will stand out more if is not underlined. It is generally good practice to not underline text that isn’t a link as it can be confusing to some users.

Property:	text-decoration	
Value:	overline *or* line-through *or* underline *or* none	

Example:	p {text-decoration: underline;}
	a {text-decoration: none;}	

Result:	Sets the CSS rule to underline all text of your paragraphs and sets the rule to NOT underline any links.	

15.3- Indenting:

To indent the first line of a paragraph use the TEXT-INDENT property. The value you assign can be in the form of pixels (px), inches (in), millimeters (mm), centimeters (cm), points (pt), picas (pc) or x-height (ex). The most commonly used is pixels (px).

Property:	text-indent	
Value:	px, in, mm, cm, pt, pc *or* ex	
Example:	p {text-indent: 35px;}	
Result:	Sets the CSS rule to indent the first line of paragraphs by 35 pixels.	

15.4- Transformation:

The TEXT-TRANSFORM property is used to define the case your text is displayed in. There are three values available: uppercase, lowercase and capitalize. Uppercase turns all letters into uppercase (similar to using the caps lock key), lowercase turns all letters into lowercase, and capitalize sets the rule to capitalize the first letter of each word.

Property:	text-transform	
Value:	uppercase *or* lowercase *or* capitalize	
Example:	p {text-transform: uppercase;}	
Result:	Sets the CSS rule to capitalize every letter in the paragraphs.	

15.5- Alignment:

You can use the TEXT-ALIGN property to control the horizontal alignment of any block-level text. Block-level text are paragraphs, tables and other elements that have a blank line before and after them. The value you assign to the alignment is either “left”, “right”, “center” or “justify”.

Property:	text-align	
Value:	left, right, center *or* justify	
Example:	h1 {text-align: center;}	
Result:	Sets the CSS rule to center all level 1 headings.	

15.6- Fonts:

To control the fonts in a CSS style rule, use the FONT-FAMILY property. When using the FONT-FAMILY property you want to list several fonts as not all fonts are supported by all browsers. Your user’s browser will attempt each of the fonts, in the order they are listed. When listing multiple fonts they are separated by a comma. For example: listing Arial, Helvetica, “Times New Roman” will display Arial, if available, then try Helvetica, then Times New Roman, and so on. When adding font values to the FONT-FAMILY property, you only need to use quotes with multiple word fonts, like “Times New Roman”.

Property:	font-family	
Value:	font name	
Example:	p {font-family: verdana, georgia;}	
Result:	Sets the CSS rule to apply the verdana font first, then georgia to the text of paragraphs.	

15.7- Font Sizes:

The FONT-SIZE property allows you to create a CSS rule that controls the size of the font. The value you assign can be a numerical value in the form of pixels (px), inches (in), millimeters (mm), centimeters (cm), points (pt), picas (pc), x-height (ex), em (the height of the current font, 1 em is equal to 16px) or in a percentage. The default font size for browsers is 16px. It is generally a good idea to use the em value, as this allows the browser to correctly display your text when the zoom tool is used to resize your webpage.

Property:	font-size	
Value:	px, in, mm, cm, pt, pc, ex, em *or* percentage	
Example:	p {font-size: 1.5em;}	
Result:	Sets the CSS rule to display all paragraph text in 1.5 em or 24px.	

15.8- Letter Spacing (Kerning):

Kerning is a term that refers to controlling the horizontal spacing between characters of text. You can control the kerning of text by using the LETTER-SPACING property. The value you assign can be in the form of points (pt), pixels (px), inches (in), millimeters (mm), centimeters (cm), picas (pc), x-height (ex), em.

Property:	letter-spacing	
Value:	pt, px, cm, mm, in, pc, ex *or* em	
Example:	h1 {letter-spacing: 5pt;}	
Result:	Welcome to TeachUcomp	Sets the CSS rule to kern level 1 headings by 5 points.

15.9- Line Spacing (Leading):

Leading (pronounced *ledding*) is a term that refers to controlling the vertical spacing between lines of text. You can control the leading of text by using the LINE-HEIGHT property. The value you assign is typically expressed as a multiple of the height of the font. However, you can also assign spacing as a percentage of the font, or an absolute value measured in the form of point (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in, picas (pc), x-height (ex), or em.

Property:	line-height	
Value:	x.x	Where “x.x” is expressed as a multiple of the font
	x%	Where “x” is a percentage of the font.
	pt, px, cm, mm, in, pc, ex, *or* em	
Example:	p {line-height: 2.0;}	Expressed as two times the height of the font.
Result:	Sets the CSS rule to apply a level of two times the height of the font as the leading.	

15.10- Text Color:

You can use the COLOR property to control the color of text. You can assign either a hexadecimal value or use the name of a color. The COLOR property can also be applied to other elements such as tables, borders and lines.

Property:	color	
Value:	color name or hexadecimal color	
Example:	h1 {color: maroon;}	
Result:	Sets the CSS rule to apply a maroon font color to all level 1 headings.	

15.11- Margins:

If you want to control the margins of elements within your HTML pages, you can use the MARGIN property in your style sheets. You can change all four margins of an element with the MARGIN property, so you must also add a dash (-) and the name of the margin you want to change (left, right, top or bottom). The value you assign can be in points (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex), or em.

Property:	margin-x	Where “x” is the margin you want to change (left, right, top or bottom).
Value:	pt, px, cm, mm, in, pc, ex *or* em	
Example:	h2 {margin-left: 30px;}	
Result:	Sets the CSS rule to apply a left margin of 30 pixels to all level 2 headings.	

15.12- Padding:

The PADDING property is used when you want to add padding (blank spaces) around the content of an element. The value you assign can be in points (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex), or em.

Property:	padding	
Value:	pt, px, cm, mm, in, pc, ex *or* em	
Example:	h1 {padding: 30px;}	
Result:	Sets the CSS rule to apply padding of 30 pixels around all level 1 headings.	

15.13- Borders:

Adding a border to a webpage element can add emphasis and make the element stand out on your page. The property you use in your CSS rule is BORDER. You can assign a thickness to the border by using a value of “thin”, “medium” or “thick”. In addition, in order for Web browsers to display your border you must also specify a border style; “solid”, “double”, “groove”, “ridge”, “inset”, “outset”, “dotted” or “dashed”. After you have indicated which border style you want, you can also add a space and color value to change the border’s color. Additionally, it is generally a good idea to add some padding to elements surrounded by a border so that the text is easy to read. Remember to separate the BORDER property and its assigned values and the PADDING property with a semicolon.

You can give your borders rounded corners by using the BORDER-RADIUS property. The value you assign can be in points (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex), or em.

Property:	border	
Value:	thin, medium, thick	Changes thickness of the border.
	Solid, double, groove, ridge, inset, outset, dotted *or* dashed	**Required value.** Identifies the border type.
Example:	h1 {border: solid navy; padding: 10px;}	
Result:	Sets the CSS rule to apply a solid navy border around all level 1 headings with a 10px padding.	
Property:	border-radius	
Value:	pt, px, cm, mm, in, pc, ex *or* em	
Example:	h1 {border: solid navy; border-radius: 25px;}	
Result:	Sets the CSS rule to apply a solid navy border, with a 25px rounded corner to all level 1 headings.	

15.14- Styling Links:

You can use CSS to assign different colors and decorations to links in their various states in your HTML pages. By default, a link's color changes after a visitor has clicked on it. This helps visitors know where they have been already. If you want to change the link properties, use a:? (where ? Is "link" for links not yet visited, "visited" for visited links, "hover" for when your mouse pointer is over the link and "active" for when you are currently clicking on a link) followed by a color value. In addition, if you assign a value of "text-decoration: none;", you can remove the default underlining that appears beneath a link.

Property:	a:? {color:}	
Value:	link, visited, hover *or* active *(to identify which links)* color value *(to assign the color)*	
Example:	a:link {color: navy; text-decoration: none;}	
Result:	Sets the CSS rule to make all unselected links navy with no underline.	

15.15- Number and Bullet Styles:

If your goal is to change the look of your ordered or unordered lists, you can use the LIST-STYLE-TYPE property. Value choices for ordered lists are: "decimal" (the default), "lower-alpha", "upper-alpha", "lower-roman" and "upper-roman". Value choices for unordered lists are: "disc" (the default), "circle" or "square".

You can also use an image as a list marker using the LIST-STYLE-IMAGE property. The value choice for this property is the file name of the image you want to use.

Property:	list-style-type	
Value:	decimal, lower-alpha, upper-alpha, lower-roman *or* upper-roman	For ordered lists.
	disc, circle *or* square	For unordered lists.
Example:	ul {list-style-type: circle;}	
Result:	Sets the CSS rule to apply circles as the bullets for all unordered lists.	
Property:	list-style-image	
Value:	url(sample.jpg)	
Example:	ul {list-style-image: url(sample.jpg);}	
Result:	Sets all unordered lists to use the image as bullet.	

15.16- Sizing Elements:

If you want to control the sizing of certain elements in your pages, you can use the WIDTH and HEIGHT properties in your style sheets. The value you assign can be in points (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex), em or as a percentage of the page size.

Property:	width, height	
Value:	pt, px, cm, mm, in, pc, ex, *or* em or a percentage of the page	
Example:	img {width: 167px; height: 145px;}	
Result:	All image tags are resized to 167 x 145 pixels.	

15.17- Text Wrapping:

The WHITE-SPACE property is used when you want to ensure text wraps in the element you are using. The default for any element is to automatically wrap text. The values for the WHITE-SPACE property are “normal” (the default) and “nowrap” which disables wrapping inside the element. The only time you need to use the WHITE-SPACE property is when you want to turn off text wrapping, as the default is to always wrap inside HTML elements.

Property:	white-space	
Value:	normal, nowrap	
Example:	div {white-space: nowrap;}	
Result:	Sets the CSS rule to turn off text-wrapping for all div elements.	

15.18- Shadowing:

If you are looking to add greater impact to any of your text you can use the TEXT-SHADOW property. The value you assign can be in points (pt), pixels (px), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex) or em. You also need to set a value for the color of the shadow, using either a color name or hexadecimal value. When defining text shadows you need to set a value for the horizontal shadow, the vertical shadow, the blur distance, and the color of the shadow.

Property:	text-shadow	
Value:	pt, px, cm, mm, in, pc, ex, *or* em and a color name or hexadecimal	
Example:	h1 {text-shadow: 5pt 5pt 5pt blue;}	
Result:	Sets the CSS rule to show all level 1 headings with a 5pt shadow in blue.	

ACTIONS-
WORKING WITH TEXT IN CSS

EMPHASIZING TEXT (BOLD AND ITALIC):

1. Type: X {font-weight: Y;}
   (where "X" is the element you wish to control and "Y" is the value either as "bold" **OR** a numerical value in multiples of 100)
2. Type: X {font-style: Y;}
   (where "X" is the element you wish to control and "Y" is the value: italic, oblique or none)

DECORATION:

1. Type: X {text-decoration: Y;}
   (where "X" is the element you wish to control and "Y" is the value: overline, line-through, underline, or none)

INDENTING:

1. Type: X {text-indent: Y;}
   (where "X" is the element you wish to control and "Y" is the value in px, in, mm, cm, pt, pc or ex.)

TRANSFORMATION:

1. Type: X {text-transform: Y;}
   (where "X" is the element you wish to control and "Y" is the value capitalize, uppercase, lowercase, or none)

TEXT ALIGNMENT:

1. Type: X {text-align: Y;}
   (where "X" is the element you wish to control and "Y" is the value left, right, center, or justify)

FONTS:

1. Type: X {font-family: "Y", "Z";}
   (where "X" is the element you wish to control and "Y" and "Z" are your font choices, in quotes if a multi-word font).

FONT SIZES:

1. Type: X {font-size: Y;}
   (where "X" is the element you wish to control and "Y" is the value in px, in, mm, cm, pt, pc, ex, or em **OR** percentage)

ACTIONS-
WORKING WITH TEXT IN CSS

LETTER SPACING (KERNING):

1. Type: X {letter-spacing: Y;}
   (where “X” is the element you wish to control and “Y” is the value in px, in, mm, cm, pt, pc, ex, or em)

LINE SPACING (LEADING):

1. Type: X {letter-height: Y;}
   (where “X” is the element you wish to control and “Y” is the value in px, in, mm, cm, pt, pc, ex, or em)

TEXT COLOR:

1. Type: X {color: Y;}
   (where “X” is the element you wish to control and “Y” is the color name or hexadecimal color value)

MARGIN:

1. Type: X {margin-Y: Z;}
   (where “X” is the element you wish to control, “Y” is the margin you want to control with left, right, top or bottom and “Z” is the size of the margin in px, in, mm, cm, pt, pc, ex, or em)

PADDING:

1. Type: X {padding: Y;}
   (where “X” is the element you wish to control and “Y” is the size of the padding in px, in, mm, cm, pt, pc, ex, or em)

BORDERS:

1. Type: X {border: Y Z; padding: ?;}
   (where “X” is the element you wish to control and “Y” is the required value border type in solid, double, groove, ridge, inset, outset, dotted, or dashed and “Z” is the border thickness in thin, medium or thick and “?” is the value of the padding in px, in, mm, cm, pt, pc, ex, or em)

STYLING LINKS:

1. Type: a:X {color:Y;}
   (where “X” is the type of link you want to change – as link, visited, hover, or active and “Y” is the color value you wish to assign)

ACTIONS-
WORKING WITH TEXT IN CSS

NUMBER AND BULLET STYLES:

1. Type: X {list-style-type: Y;}
   (where "X" is the type of list you wish to control and "Y" is the value as decimal, lower-alpha, upper-alpha, lower-roman, upper-roman, disc, circle, or square)
2. Type: X {list-style-image: url(Y);}
   (where "X" is the type of list you wish to control and "Y" is the file name of the image you want to use as your list marker)

SIZING ELEMENTS:

1. Type: X {width: Y; height: Z;}
   (where "X" is the element you wish to control and "Y" and "Z" are the size values in px, in, mm, cm, pt, pc, ex, or em)

TEXT WRAPPING:

1. Type: X {white-space: Y;}
   (where "X" is the element you wish to control and "Y" is the value normal or no-wrap)

SHADOWING:

1. Type: V {text-shadow: W X Y Z;}
   (where "V" is the element you wish to control, "W", "X" and "Y" are the sizes of the shadow in px, in, mm, cm, pt, pc, ex, or em and "Z" is the hexadecimal color value or color name)

EXERCISES-
WORKING WITH TEXT IN CSS

Purpose:

1. To create a simple *internal* CSS and apply it to the HTML page created in earlier lessons.

Exercises:

1. Open your HTML page created in previous chapters.
2. On the line before the </head> end tag, type: <style>
3. Press "Enter".
4. Type: h1 {color: maroon; font-style: italic;}
5. Press "Enter".
6. Type: table {background: yellow;}
7. Press "Enter".
8. Type: a:link {color: green;}
9. Press "Enter".
10. Type: </style>
11. Press "Enter".
12. Save Document.

Purpose:

1. To create a simple *external* CSS and apply it to the HTML page created in earlier lessons.

Exercises:

1. Open your text editor and start a new document.
2. Type: p {color: #0000FF; font-family: arial, georgia;}
3. Press "Enter".
4. Type: div {background: #D0D0D0;}
5. Press "Enter".
6. Type: a:hover {color: red;}
7. Save the document as a text file, name it "my-test-stylesheet" with a .css extension.
8. Open your HTML page created in earlier lessons.
9. Between the <head>…</head> tags, type: <link rel="stylesheet" type="text/css" href="my-test-stylesheet.css">
10. Press "Enter".
11. Save Document.

CHAPTER 16-
Creating Backgrounds in CSS

16.1- Colors:

To set the background color in your webpage you want to use the BACKGROUND-COLOR property. Like other color selection properties you can use either a hexadecimal value or color name. The background of your whole page is defined using the body selector.

Property:	background-color	
Value:	color name or hexadecimal value	
Example:	body {background-color: #ADD8E6;}	
Result:	Sets the CSS rule to make the background color of your webpage light blue.	

16.2- Images:

To set an image as the background of your webpage you want to use the BACKGROUND-IMAGE property. You can use this property to set the background of your whole page using the body selector or you can use it for any element you'd like to have an image as the background, like a table or paragraph.

When using an image as a background, the default setting is for the image to repeat itself to fill the available space. You can specify how you would like the image to repeat or to not repeat at all using the BACKGROUND-REPEAT property with the values: repeat-x (repeats only horizontally), repeat-y (repeats only vertically) or no-repeat.

It is important to select an image that does not compete with the text or other elements of your webpage. Using a complex image may result in difficulties when users are trying to view your webpage.

Property:	background-image	
Value:	url("X")	The link to your image file.
Supporting Property:	background-repeat	To define image repetition
Value:	repeat-x, repeat-y *or* no-repeat.	

Example:	body {background-image: url("background.jpg");
	background-repeat: repeat-y;}	

Result:	Applies the CSS rule to make your background the selected image and to have the image repeat vertically only.	

16.3- Fixed Images:

When using an image as a background you can also control whether or not it scrolls with the other elements of your webpage. Using the BACKGROUND-ATTACHMENT property and the values: scroll or fixed. Scroll is the default setting and allows the background image to move when your user is navigating your webpage and fixed sets the background image to not move. For example, when the image is set to fixed, the position of the image will always be in the background, no matter what part of the page your user is currently viewing.

Property:	background-attachment	
Value:	scroll or fixed	
Example:	body {background-image: url(“background.jpg”); background-attachment: fixed;}	
Result:	Sets the CSS rule to make the background the image you selected and to have a fixed position on the page.	

ACTIONS-
Creating Backgrounds in CSS

COLORS:

1. Type: X {background-color: Y;}
   (where “X” is the element you wish to control and “Y” is the color name or hexadecimal value)

IMAGES:

1. Type: X {background-image: url(“Y”);}
   (where “X” is the element you wish to control and “Y” is the link to the image you want to use)

FIXED IMAGES:

1. Type: X {background-image: url(“Y”); background-attachment: Z;}
   (where “X” is the element you wish to control and “Y” is the link to the image you want to use and “Z” is the value scroll or fixed)

EXERCISES-
Creating Backgrounds in CSS

Purpose:

1. To add a background to the external style sheet created in the previous chapter.

NOTE: Image files should be selected from your computer, for this exercise replace “sample.jpg” with your name and file path of your chosen image.

Exercises:

1. Open your .css document
2. After the last line, type: body {background-image: url(“sample.jpg”); background-attachment: fixed;}
3. Save your document

CHAPTER 17-
Images in CSS

17.1- Opacity:

You can adjust an image's opacity using CSS by implementing the OPACITY property and assigning a value from 0.0 – 1.0. The lower the value the more transparent your image or element.

To adjust the opacity for specific images you need to name your images using the ID selector, as discussed in Chapter 14.

Property:	opacity	
Value:	0.0 to 1.0	The lower the value the more transparent the image.
Example:	img {opacity: 0.5;}	
Result:	Sets the CSS rule to display all images at 50% opacity.	

17.2- Floating Images:

When you want an image to be placed to the left or right of another element you use the FLOAT property. The FLOAT property uses the values: left and right. The floated image will move to the farthest point left or right that it can, allowing any elements that come after it to "flow" around the image. For example, an image floated to the right will allow the text of the following element to wrap around the left side of the image. Any elements that come before the floated image will not be affected.

Property:	float	
Value:	left or right	
Example:	img {float: left;}	
Result:	Sets the CSS rule for all images to be on the left-hand side of your page with any following elements wrapped on the right side of the image.	

17.3- Image Galleries:

Image galleries can be created in CSS using the DIV element. You will be using the CLASS property and values that correspond with your HTML code. In order for the values to work properly the images and descriptions you are using must correspond with the images and image descriptions in your HTML code. For instance, in your HTML code, the images you want to display in your gallery need to be in the <div class="img"> (where "img" is the name you choose for your div element) and your CSS will use the DIV element and a value of img (div.img). In the same way your image descriptions need to be labeled with <div class="desc"> (where "desc" is the name you assign to the div element containing your image descriptions). It is important to note that spelling and case are important when labeling classes in HTML and CSS as they must match exactly.

Only the images and descriptions that are in the labeled <div> tags on your HTML code will be displayed in a gallery. Any other images you have will be displayed in the order or position they appear in your HTML code. You can set as many different div tags as you would like, if your goal is to display all your images in galleries.

You will also need to assign height, width, margin, padding, border, display, float and text-align properties to assure proper positioning of your images and descriptions. You can also assign an a:hover property and class to your images, so that when you mouse over the image a different border surrounds just the image allowing it to stand out.

Property:	div	
Value:	class, desc	
Example:	div.img {margin: 5px; padding: 5px; border: 1px solid #C0C0C0; height: 250px; width: 250px; float: left; text-align: left;} ***(this code sets the size, position and border for the image and text as a whole)*** div.img img {display: inline; margin: 3px; border: 2px solid blue;} ***(this code sets the images to display one after the other in a row and sets the border around each image)*** div.img a:hover img {border: 2px dotted red;} ***(this code set the hover border around the image)*** div.desc {text-align: left; font-weight: bold; width: 150px; margin: 2px;} ***(this code sets the weight and size of the text)***	
Result:	Sets the CSS rules to display images in a row with each image and description being contained in a silver border and showing a dotted red border when you mouse over each image.	

17.4- Image Sprites:

Using CSS3 you can create an image sprite, which is simply a set of graphics that is read as a single image by the Web browser. Since an image heavy webpage can sometimes take a long time to load, this shortens load times by reducing the number of server requests. This is useful when you have several images you want to display in different sections of your page.

To make image sprites work you need to know the sizes and positions of your combined images. For example, you have two 50 pixel by 50 pixel images, set top to bottom, making one 50 pixel by 100 pixel image. You have to set your CSS language to separate the images and only display the one you want in the position you want it. You must specify the size of each image and then set the position to read it from. For example, with a 50px by 100px image, set top to bottom, the second image would be read at 0 -50px. Telling the browser to read horizontally starting at 0 and vertically at -50px (halfway down the full image).

The easiest way to do this is using the <div> tag as you can set each section of the image you want to display in a <div> tag and place it wherever on the page you would like.

Property:	div	
Value:	width, height, background:url	
Example:	div.img {width: 50px; height: 50px; background:url (sample.jpg) 0 0;} ***(sets the size of the first image, where 0 0 is the start point)*** div.secondimg {width: 50px; height: 50px; background:url (sample.jpg) 0 -50px;} ***(sets the size of the second image, where 0 -50px tells the browser the position of the second image)***	
Result:	Sets the CSS rule to show the first half of the image in a <div> tag and the second image in a separate <div> tag.	

ACTIONS-
IMAGES IN CSS

OPACITY:

1. Type: X {opacity: Y;}
(where "X" is the element you wish to control and "Y" is the degree of transparency from 0.0 to 1.0)

FLOATING IMAGES:

1. Type: X {float: Y;}
(where "X" is the element you wish to control and "Y" is direction you want to float, left or right)

IMAGE GALLERIES:

1. Type: div.A {margin: B; padding: C; border: D E F; height: G; width: H; float: I; text-align: J;}
(where "A" the img tag in the <div> element you want to control, "B" is the size in of your margin, "C" is the amount of padding, "D" is size of your border: px, in, mm, cm, pt, pc, ex, or em, "E" is the type of border: solid, double, groove, ridge, inset, outset, dotted, or dashed , "F" is the color: by name or hexadecimal, "G" is the height of your image and description: px, in, mm, cm, pt, pc, ex, or em, "H" is the width of your image and description: px, in, mm, cm, pt, pc, ex, or em, "I" is the position of the starting image: left or right, and "J" sets the justification for your text: left, right, center or justify)
2. Type: div.A A {display: B; margin: C; border: D E F;}
(where "A" is the img tag in your <div> element, "B" is the type of display: block or inline, "C" is the size of your margin: px, in, mm, cm, pt, pc, ex, or em, "D" is the size of the border: px, in, mm, cm, pt, pc, ex, or em, "E" is the type of border: solid, double, groove, ridge, inset, outset, dotted, or dashed, "F" is the color of the border: color name or hexadecimal)
3. Type: div.A a:hover A {border: B C D;}
(where "A" is the img tag in your <div> element, "B" is the size of your border: px, in, mm, cm, pt, pc, ex, or em, "C" is the type of border: solid, double, groove, ridge, inset, outset, dotted, or dashed, and "D" is the color: color name or hexadecimal)
4. Type: div.A {text-align: B; font-weight: C; width: D; margin: E;}
(where "A" is the defined text in your <div> tag, "B" is the alignment of your text: left, right, center, justify, "C" is the weight of your font: bold or 100…900, "D" is the width of your text: px, in, mm, cm, pt, pc, ex, or em, and "E" is the margin size: px, in, mm, cm, pt, pc, ex, or em)

IMAGE SPRITES:

1. Type: div.A {width: B; height: C; background:url (E) F G;}
(where "A" is the img tag in your <div> element, "B" is the width of your image: px, in, mm, cm, pt, pc, ex, or em, "C" is the height of your image: px, in, mm, cm, pt, pc, ex, or em, "E" is the link to your image file, "F" is the horizontal starting point of your image: px, in, mm, cm, pt, pc, ex, or em, and "G" is the vertical starting point of your image : px, in, mm, cm, pt, pc, ex, or em)
2. Repeat for any further images in your sprite, where "A" will change to indicate the second, third, etc. images in your sprite as classified in your <div> tag.

EXERCISES-
IMAGES IN CSS

Purpose:

1. To add an image gallery to your webpage

NOTE: Image files should be selected from your computer, for this exercise replace “sample.jpg” with your picture and “sample.htm” with the file path of your chosen image.

Exercises:

1. Open your HTML document created in previous exercises.
2. On the last line before the </body> tag, type: <div class=“img”>
3. Press “Enter”.
4. Type: <a href=“sample.htm”>
5. Press “Enter”.
6. Type: <img src=“sample.jpg”>
7. Press “Enter”.
8. Type: </a>
9. Press “Enter”.
10. Type: <div class=“desc”>Add a description of your image here</div>
11. Press “Enter”.
12. Type: </div>
13. Repeat steps 2 through 12 for each image you want to include in your gallery
14. Save your document.

Exercises:

1. Open your .css document created in previous exercises.
2. On the last line type: div.img {margin: 5px; padding: 5px; border: 1px solid blue; height: 250px; width: 250px; float: left; text-align: left;}
3. Press “Enter”.
4. Type: div.img img {display: inline; margin: 2px; border: 2px solid #000000;}
5. Press “Enter”.
6. Type: div.img a:hover img {border: 2px; dashed red;}
7. Press “Enter”.
8. Type: div.desc {text-align: left; font-weight: 200; width: 200px; margin: 2px;}
9. Press “Enter”.
10. Save your document.

CHAPTER 18- Box Model in CSS

Box Model in CSS

18.1- What is a Box Model?:

When working in CSS, the box model is used for design and layout. Every element in your HTML coding is considered a box. For example, a paragraph, image and heading, are all separate boxes. You want to use the box model to control their placement and orientation to one another, by controlling the margins, borders, padding and the actual content of each “box”.

When using box models it is important to properly calculate the total size of the elements you are working with. When you set the width and height of an element in CSS you are only setting the size of the content. For the total size of each element you have to consider the content, padding, borders and margins. This is the formula you should use when thinking of the size of your elements: Total width= width of your content + left padding + right padding + left margin + right margin + left border + right border. The same concept is used for height replacing left and right with top and bottom.

You’ll note that each side needs to be calculated separately as each part can be said to be a separate piece.

Example:	p.one {width: 220px; padding; 10px; border 5px solid maroon; margin: 0px;}	
Result:	Sets your paragraph element with the ID “one” to have a total width of 250 pixels, a 5 pixel solid border in the color maroon.	

18.2- Margin:

The MARGIN property in CSS controls the amount of space around your element **outside** the assigned border You can use one value to control the whole margin or you can control each side independently. To do this you use the properties: MARGIN-LEFT, MARGIN-RIGHT, MARGIN-TOP, MARGIN-BOTTOM or MARGIN (to control all elements with one value). You can express the values for your margins by using: auto (which allows the browser to define the size), a percentage amount (%) or pixels (px), points (pt), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex) or em.

The margin of your HTML elements is always transparent and is not affected by any background color your element may be set with.

If you decide to use the individual control properties, you must assign a value or it will be set to the default value of 0px.

Property:	margin, margin-left, margin-right, margin-top, margin-bottom	
Value:	auto, a percentage (%) *or* px, pt, cm, mm, in, pc, ex or em	
Example:	p.one {margin: 50px;}	
Result:	Sets the CSS rule to have paragraph “one” with a margin of 50 pixels on all sides.	

18.3- Padding:

The PADDING property controls the amount of space **inside** the border and between the content of your HTML element. The PADDING element is affected by the background color inside your element and will appear as the same color.

The values for the PADDING element can be expressed in two ways: by a percentage amount (%) or using pixels (px), points (pt), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex) or em.

Just like the MARGIN property, the PADDING property can also control the individual sides of an element using: PADDING-LEFT, PADDING-RIGHT, PADDING-TOP, PADDING-BOTTOM or PADDING (to affect all sides equally).

If you decide to use the individual control properties, you must assign a value to each property, or any unassigned side will be set at the default of 0px.

Property:	padding, padding-left, padding-right, padding-top, padding-bottom	
Value:	a percentage (%) *or* px, pt, cm, mm, in, pc, ex or em	
Example:	p.one {padding-left: 5px; padding-right: 5px; padding-top: 10px; padding-bottom: 10px;}	
Result:	Sets the CSS rule to have paragraph “one” with a padding of 5 pixels on both the left and right side, and 10 pixels on both the top and bottom.	

18.4- Border:

To set the borders of an element you need to use the BORDER-STYLE property. You must use the BORDER-STYLE property before trying to control the size and color of your borders. The values for BORDER-STYLE are: solid, double, groove, ridge, inset, outset, dotted or dashed.

You can also control the size of your border using the BORDER-WIDTH property. You can assign the BORDER-WIDTH properties the values of: thin, medium or thick, *or* pixels (px), points (pt), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex) or em.

You can control the color of your borders using the BORDER-COLOR property. The BORDER-COLOR values are expressed using the color name, a hexadecimal value or transparent (creates an invisible border).

As with the margins and padding properties you can control the style of border on individual sides using the properties: BORDER-LEFT-STYLE, BORDER-RIGHT-STYLE, BORDER-TOP-STYLE and BORDER-BOTTOM-STYLE.

Property:	border-style	
Value:	solid, double, groove, ridge, inset, outset, dotted *or* dashed	**Required value.** Identifies the border type.
Property:	border-width	
Value:	thin, medium, thick *or* px, pt, cm, mm, in, pc, ex or em	
Property:	border-color	
Value:	color name, hexadecimal value or transparent	
Property:	border-left-style, border-right-style, border-top-style, border-bottom-style	
Value:	Solid, double, groove, ridge, inset, outset, dotted *or* dashed	
Example:	p.one {border-style: dashed; border-width: medium; border-color: red;}	
Result:	Sets the CSS rule, for paragraph “one”, to have a medium sized, dashed border in the color red.	

18.5- Outline:

The outline property is used to add extra emphasis to any element on your HTML page. It is drawn outside the border of your element. It is important to note that the outline property does not affect the total size of your element. It is not included when calculating the dimensions of the element you want to effect.

You need to define the style of an outline using the OUTLINE-STYLE property. The values for this property are: solid, double, groove, ridge, inset, outset, dotted, dashed or none.

You can also control the width of your outline using the OUTLINE-WIDTH property. The values for OUTLINE-WIDTH can be set using: thin, medium, thick *or* using pixels (px), points (pt), centimeters (cm), millimeters (mm), inches (in), picas (pc), x-height (ex) or em.

The color of your outline can be controlled using the OUTLINE-COLOR property. You can set the values for this property using either the color name or a hexadecimal value.

Property:	outline-style	
Value:	solid, double, groove, ridge, inset, outset, dotted, dashed *or* none	Identifies the outline type.
Property:	outline-width	
Value:	thin, medium, thick *or* px, pt, cm, mm, in, pc, ex or em	Sets the size of the outline
Property:	outline-color	
Value:	color name, hexadecimal value	Defines the color of the outline
Example:	p.one {outline-style: double: outline-width: thin: outline-color: #FF0000;}	
Result:	Sets the CSS rule, for paragraph “one”, to have a thin, double outline in red.	

ACTIONS-
Box Model in CSS

MARGIN:

1. Type: X {margin: Y;}
   (where "X" is the element you wish to control and "Y" is the size you want your margin: auto, a percentage (%) *or* px, pt, cm, mm, in, pc, ex or em)

PADDING:

1. Type: X {padding: Y;}
   (where "X" is the element you wish to control and "Y" is the size of your padding: a percentage (%) *or* px, pt, cm, mm, in, pc, ex or em)

BORDER:

1. Type: X {border-style: Y; border-width: Z; border-color: A;}
   (where "X" is the element you wish to control and "Y" is the style of your border: solid, double, groove, ridge, inset, outset, dotted *or* dashed, "Z" is the size of your border: thin, medium, thick *or* px, pt, cm, mm, in, pc, ex or em and "A" is the color: color name or a hexadecimal number)

OUTLINE:

1. Type: X {outline-style: Y; outline-width: Z; outline-color: A;}
   (where "X" is the element you wish to control and "Y" is the style of your outline: none, solid, double, groove, ridge, inset, outset, dotted *or* dashed, "Z" is the size of your outline: thin, medium, thick *or* px, pt, cm, mm, in, pc, ex or em and "A" is the color: color name or a hexadecimal number)

EXERCISES- Box Model in CSS

Purpose:

1. To define the padding, border, margin and outline of an element in your HTML document.

Exercises:

1. Open your .css document
2. After the last line, type: h1{padding: 5px; border-style: double; border-width: thin; border-color: blue; margin: 5px; outline-style: dotted; outline-width: thick; outline-color: #C0C0C0;}
3. Press “Enter”.
4. Save your document.

CHAPTER 19-
Working with Elements in CSS

19.1- Display and Visibility:

There are two ways to hide or display an element on your webpage. They are the DISPLAY property and the VISIBILITY property. While they are similar in function, the results are different when it comes to the layout of your page.

The DISPLAY property can be used to hide your element or to set it as a block or an inline element. The basic values for the DISPLAY property are: none, block *or* inline. The "none" value hides the element from the page and will not take up any space on your layout. The "block" value will display your element with line breaks before and after the content. The "inline" value will display the element without any line breaks. A few examples of block elements are: <p>, <div> and <h1>. Examples of inline elements include <a> and <span>.

When using the VISIBILITY property to hide an element on your page, the layout will still read the element as if it was there, meaning there will be a blank space where that element should be. The values for the VISIBILITY property are: visible (the default), hidden and collapse.

The collapse value is a useful tool to use when you want to quickly remove columns or rows from a table without having to recalculate the height and width of the table as a whole. The values originally assigned the table will stay the same, but the specific rows or columns will collapse and be hidden.

Property:	display	
Value:	none, block, inline	
Property:	visibility	
Value:	visible, hidden, collapse	

Example:	h1.hidden {display: none;}
	h3.hide {visibility: hidden;}	

Result:	All h1 headers with class="hidden" are not displayed and no space will be seen and all h3 headers with class="hide" are not displayed but the space they are assigned is left blank on your page.	

19.2- Grouping and Nesting:

Grouping a set of selectors is helpful when you want to change the look of a whole set of elements. For example if you want to change the color of all headers and paragraphs, you can group your elements together, with each different element separated by a comma: h1, h2, p. You can then set the CSS rule to change whatever attribute you would like them all to have.

You can also change a style for a selector that is within a selector. For example you could have one style set for all paragraph elements (p), a second style set for all elements with the class of class="definitions", and a third style that is set for only paragraph (p) elements that lie within the element with the class: class="definitions".

Grouping Example:	h1, p {color: purple;}	
Result:	Sets the CSS rule that all level 1 headings and all paragraphs to the color purple.	

Nesting Example:	p {color: black; text-align: left;}
	.definitions {background-color: blue;}
	.definitions p {color: #FFFFFF;}	

Result:	All paragraphs have black text and are left-aligned, all elements with class="definitions" have a background color of blue and all paragraphs within an element with class="definitions" have white text.	

19.3- Dimensions:

You can control the dimensions of your elements using the height and width properties. You can also ascribe a maximum or minimum height and width using the max-height, max-width, min-height, min-width attributes. Any elements not assigned a specific dimension will span the browser window horizontally and only take up the actual size of the contents vertically. The values for these attributes can be expressed in pixels, a percentage of the webpage or as auto. The attribute value of auto allows the browser to calculate the size of the element.

Attributes:	height, width, min-height, min-width, max-height, max-width	Values in pixels, percentage or auto
Example:	p.one {width: 100px; max-height: 100px; background-color: blue;}	
Result:	Sets paragraph with class="one" to a width of 100 pixels and a max height of 100 pixels with a background color of blue.	

19.4- Positioning:

You can control the way elements position themselves on your webpage using the four different position properties. They are STATIC (the default), FIXED, RELATIVE and ABSOLUTE. They can be adjusted using the top, bottom, left and right attributes and are assigned a numerical value using pixels (px). It is important to remember that positioning attributes will not work without first assigning a position property and they will work differently depending on which property you use.

As mentioned the STATIC property is the default and is not affected by the positioning attributes. All elements are considered to be static unless given a specific positioning property.

The FIXED property forces an element into a fixed position relative to the browser window. The fixed element will not move, even when the page is scrolled. Fixed position elements are not part of the normal “flow” of the webpage and any other elements on your page will ignore the fixed element. Fixed elements can and will overlap any other elements on your webpage.

The RELATIVE property will move an element relative to it’s normal position. Relative positioned elements reserve the space of the normal position in the flow of your page after they are repositioned and will overlap other elements that occupy the space.

Like a fixed position element, the ABSOLUTE property sets an element in a specific location and it is not affected by the flow of the page. All other elements and your webpage as a whole, act as if the absolutely positioned element is not there. An ABSOLUTE positioned element is placed relative to the first element that has a position set other than static. If you don’t have any other specified elements the position defaults to the <html> tag (the first position on your page).

The Z-INDEX property controls the way elements are stacked if they overlap each other. It’s value is described numerically and can have a negative value. The Z-INDEX properties are counted with the lowest value always being behind any higher valued elements. If no Z-INDEX is assigned to positioned elements, the element positioned last in your HTML code will be on top.

When setting positions of elements you need to make sure you have a “class” defined for the element you are positioning. For example <h2 class=“top”> will allow you to change a single level 2 heading using h2.top in your CSS code so not all level 2 heading are affected by the positioning.

19.4- Positioning (cont.):

Properties:	fixed, relative, absolute	
Values:	top, bottom, right, left	Expressed numerically in pixels

Examples:	p.first {position: fixed; top: 5px; right: 100px;}
	h1.first {position: relative; top: -10px; right: -10px;}
	h2 {position: absolute; left:100px; top:150px;}	

Result:	Paragraph with class="first" is set in a fixed position 5 pixels from the top of the browser window and 100px from the right of the browser window. Level 1 headings with class="first" have a relative position 10 pixels above and 10 pixels to the right of it's original position. All level 2 headings will be positioned 100 pixels from the left of the browser window and 150 pixels from the top of the browser window.	

19.5- Floating:

When you want an element to be placed to the left or right of another element you use the FLOAT property. The FLOAT property uses the values: left and right. The floated element will move to the farthest point left or right that it can, allowing any elements that come after it to "flow" around it. For example you can use the FLOAT property to create a horizontal list of links, as in the example that follows. Any elements that come before the floated element will not be affected.

Property:	float	
Value:	left or right	

Example:	ul {float: left; width:100%; padding: 1; margin: 1; list-style-type: none;}
	a {float: left; width: 6em; text-decoration: none; color: black; background-color: yellow; padding: 4px; border-right: 2px solid blue;}
	a:hover {background-color: #C0C0C0;}
	li {display: inline;}	

Result:	Creates a horizontal list of links with a yellow background, black text, no underline, a border of blue on the right side of each link and a hover color of silver.	

19.6- Pseudo-Classes/Pseudo-Elements:

CSS pseudo-classes/pseudo-elements can be used to change specific selectors within elements of your webpage. For example, you can have the first line of a paragraph be a different font, color, have different emphasis, etc. from the rest of the paragraph by using the :first-line pseudo-class/pseudo-element.

The syntax for CSS pseudo-classes/pseudo-elements are as follows: selector:pseudo-class {property: value;}. You can also use a CSS class in conjunction with a pseudo class by adding the class to the selector: selector.class:pseudo-class {property: value;} There are currently eleven different CSS pseudo-classes that can be used to add special effects to your webpage.

CSS pseudo-classes/pseudo-elements are not case sensitive. When using pseudo-classes/pseudo-elements it is important to remember there **cannot** be a space between the selector, colon and pseudo-class/pseudo-element.

Selector	**Example**	**Description of Action**
:link	a:link	Sets rule to select all unvisited links
:visited	a:visited	Sets rule to select all visited links
:hover	a:hover	Sets rule to select links on mouse over **(a:hover MUST come after a:link and a:visited to be effective)**
:active	a:active	Sets rule to select the currently active link **(a:active MUST come after a:hover to be effective)**
:focus	input:focus	Sets rule to select the input element that is currently being used
:first-letter	p:first-letter	Sets the rule to select the first letter of every paragraph element
:first-line	p:first-line	Sets the rule to select the first line of every paragraph element
:first-child	p:first-child	Sets the rule to select every paragraph element that is the first child of its parent element
:before	p:before	Sets the rule to insert content before every paragraph element
:after	p:after	Sets the rule to insert content after every paragraph element
:lang	p:lang(was)	Sets the rule to select every paragraph element with a lang="was" attribute

19.6- Pseudo-Classes/Pseudo-Elements (cont.):

Example:	p:first-child i {color: blue;}
	a:visited {color: #FF00FF;}
	p.one:first-line {font-weight: 500;}	

Result:	Sets the CSS rules to have the first italicized word in a paragraph to the color blue, sets all visited links to fuchsia and sets the first line of paragraph class="one" to be bolder than the rest of the paragraph.	

ACTIONS- WORKING WITH ELEMENTS IN CSS

DISPLAY AND VISIBILITY:

1. Type: X {display: Y;}
   (where “X” is the element you wish to control and “Y” is the state you want that element: none, block or inline)
2. Type: X {visibility: Y;}
   (where “X” is the element you wish to control and “Y” is the state you want that element: visible, hidden, collapse)

GROUPING AND NESTING:

1. Type: A,B {C: D;}
   (where “A” and “B” are the elements you are grouping, “C” is the property you’d like to affect and “D” is the value of that property)
2. Type: A {B: C;}
   Type: .D {E: F;}
   Type: .D A {G: H;}
   (where “A” is the first element you would like to control, “B” is the property and “C” is the value, “D” is the labeled class of elements you’d like to control, “E” is the property and “F” is the value, the final line “D” is the labeled class of elements, “A” is an element contained within the labeled class, “G” is the property and “H” is the value)

DIMENSIONS OF ELEMENTS:

1. Type: X {Y: Z;}
   (where “X” is the element you wish to control and “Y” is the property: height, width, min-height, min-width, max-height or max-width and “Z” is the value expressed as pixels, a percentage of the webpage or as auto)

POSITIONING:

1. Type: A {B: C: D;}
   (where “A” is the element you wish to control and “B” is the property of: fixed, relative or absolute, “C” is the attribute value: top, bottom, left, right and “D” is the numerical value of “C” expressed in pixels.

FLOATING:

1. Type: X {float: Y;}
   (where “X” is the element you wish to control and “Y” is the value of left or right)

PSEUDO-CLASSES/PSEUDO-ELEMENTS:

1. Type: A:B {C: D;}
   (where “A” is the element you wish to control and “B” is the selector: link, visited, hover, active, focus, first-letter, first-line, first-child, before, after, lang, “C” is the property and “D” is the value)

EXERCISES-
WORKING WITH ELEMENTS IN CSS

Purpose:

1. To change the properties for all the links on your webpage.

Exercises:

1. Open your .css document
2. After the last line, type: a:link {color: green;}
3. Press “Enter”.
4. Type: a:visited {color: black;}
5. Press “Enter”.
6. Type: a:hover {color: #800080;}
7. Press “Enter”.
8. Type: a:active {color: #00FFFF;}
9. Save your document

CHAPTER 20- Adding a Navigation Bar in CSS

20.1- Vertical Navigation Bar

20.2- Horizontal Navigation Bar - Inline

20.3- Horizontal Navigation Bar - Floating

Adding a Navigation Bar in CSS

20.1- Vertical Navigation Bar:

A navigation bar is essentially a list of links. To start building your navigation bar you need to start with a basic list of links. You'll want to have your list built in your HTML code and then you can style it how you choose using CSS.

Your basic vertical navigation bar is built by removing the bullets, margins and padding from your list of links using the LIST-STYLE-TYPE property and the value of none. Set the margins and padding to zero, set the DISPLAY property with the block value and set a width. You now have a vertical navigation bar. You can now style your block property how you choose. When you set your DISPLAY property to block, it makes the whole area clickable, not just the text of the link. You can click next to your link and it will activate the link on that line.

Remember when using a vertical navigation bar to set a width, if you don't the list will take up the whole width available in the browser window.

Example:	ul.vertnav {list-style-type: none; margin: 0; padding:0;}
	a {display: block; width: 80px; background-color: #FFFFFF;}	

Result:	A vertical navigation bar that is 80 pixels wide with an aqua background color using the list with class="vertnav"	

20.2- Horizontal Navigation Bar - Inline:

There are two ways to create a horizontal navigation bar. The first and simplest is displaying your list of links with a DISPLAY property of inline for your list items (li). This sets your list to display horizontally next to each other. You set the LIST-STYLE-TYPE property the same way as you do for a vertical navigation bar, removing the list markers and setting the margin and padding to zero.

Example:	ul.horiznav {list-style-type: none; margin: 0; padding: 0;}
	li {display: inline;}	

Result:	A horizontal navigation bar using the list with class="horiznav"	

20.3- Horizontal Navigation Bar - Floating:

The second way to create a horizontal navigation bar is to use the FLOAT property to float the links in a horizontal row. When using the FLOAT property you assign it to float the individual list items (li) and not the unnumbered list (ul) as a whole. This is important to remember as you will need to specify a CLASS to each list item.

Example:	ul {list-style-type: none; margin: 0; padding: 0;}
	li.horiznav {float: left;}
	a {display: block; width: 60px;}	

Result:	A horizontal navigation bar where the list of links with class="horiznav", are floated from left to right and displayed as a block element with a width of 60 pixels	

ACTIONS-
Adding a Navigation Bar in CSS

VERITCAL NAVIGATION BAR:

1. Type: X.Y {list-style-type: none; margin: 0; padding: 0;}
   (where “X” is the list of your links and “Y” is the class id of your list)
2. Type: a {display: block; width: X;}
   (where “X” is the width of your block element. You can add any other styling to your list of links inside the block property within the curly brackets)

HORIZONTAL NAVIGATION BAR - INLINE:

1. Type: A.B {list-style-type: none; margin: 0; padding: 0;}
   (where “A” is the list of links and “B” is the class id of your list)
2. Type: li {display: inline;}

HORIZONTAL NAVIGATION BAR - FLOATING:

1. Type: X {list-style-type: none; margin: 0; padding: 0;}
   (where “X” is the list you want to use)
2. Type: li.X {float: Y;}
   (where “X” is the class id assigned to your list items and “Y” is the float value left or right)
3. Type: a {display: block; width: X;}
   (where “X” is the width you would like your link properties set)

EXERCISES-
ADDING A NAVIGATION BAR IN CSS

Purpose:

1. To add a horizontal inline navigation bar to your webpage.

NOTE: The following exercise will use #home, #contact, etc. as link references, replace with your actual page descriptors link references.

Exercises:

1. Open your HTML document from previous exercises.
2. On the line of your <body> that you want your navigation bar to appear, type: <ul class="horiznav">
3. Press "Enter".
4. Type: <li><a href="#home">Home</a></li>
5. Press "Enter".
6. Type: <li><a href="#contact">Contact</a></li>
7. Press "Enter".
8. Type: <li><a href="#faq">FAQ</a></li>
9. Press "Enter".
10. Type: <li><a href="#products">Products</a></li>
11. Press "Enter".
12. Type: </ul>
13. Save your document,

Exercises:

1. Open your .css document from previous exercises.
2. On the last line, type: ul.horiznav {list-style-type: none; margin: 0; padding: 0;}
3. Press "Enter".
4. Type: li {display: inline;}
5. Press: "Enter".
6. Save your document.

CHAPTER 21- CSS TABLES

CSS Tables

21.1- Borders:

It’s easy to control the borders of your tables using CSS. You only have to specify your styling using the BORDER property and the attributes of size in pixels, border style: solid, double, groove, ridge, inset, outset, dotted *or* dashed and a color value as a color name or hexadecimal number. You can assign a single border style to the table, table header (th) and table data (td) elements or you can have a different size, style and color for each element.

When styling borders this way your table will have a “double” border as each element has its own border. To have a single table border you use the border-collapse property, covered in the next lesson.

Example:	table, th, td {border: 2px solid black;}	
Result:	Sets all tables, th and td elements to have a 2 pixel, solid border in black.	

21.2- Collapsed Borders:

The BORDER-COLLAPSE property allows you to have a single border around your table, by ignoring the exterior border properties of the table header (th) and table data (td) elements., while still allowing interior borders for each header and data cell.

Example:	table.one {border-collapse: collapse;}
	table.one, th, td {border: 2px solid blue;}	

Result:	Sets the table with class=“one” to have a single, solid exterior border, 2 pixels wide in blue and an interior th and td border 2 pixels wide in blue.	

21.3- Table Width and Cell Height:

You can control the total table width using the WIDTH property and the height of your th and td elements by applying the HEIGHT property. These values can be set using: pixels (px), inches (in), centimeters (cm), millimeters (mm), points (pt) or em. The table WIDTH property can also be expressed in a percentage of the browser window.

Example:	table {width; 100%;}
	th {height: 50px;}
	td {height: 25px;}	

Result:	Sets the table width to span the browser window, the headers have a height of 50 pixels and the data cells have a height of 25 pixels.	

21.4- Table Background Color:

You can control the background color of your th and td elements separately by defining a BACKGROUND-COLOR property for each element. You can also set the background color of the whole table using the same property.

Example:	table, th, td {border: 1px solid black;}
	th {background-color: green;}
	td {background-color: yellow;}	

Result:	Sets the table, th and td elements to have a solid, 1 pixel black border and all th elements to have a green background and all td elements to have a yellow background.	
Example:	table, th, td {border: 1px solid black; background-color: yellow;}	
Result:	Sets all borders to a solid, 1 pixel black line with all backgrounds in yellow.	

21.5- Table Text Alignment:

You can control the horizontal and vertical alignment of the text in your table by using the TEXT-ALIGN property to control the horizontal alignment and the VERTICAL-ALIGN property to control the vertical alignment. TEXT-ALIGN has the values of left, right and center. The VERTICAL-ALIGN property has the values of top, bottom and center.

You can set the whole table to have the same alignment, by setting the alignment values under the table element styling. You can also control the th and td elements individually for more visual interest or readability.

Example:	table {text-align: left; vertical-align: center;}	
Result:	Sets all elements in the table with left horizontal alignment and centered vertically.	

Example:	th {text-align: center; vertical-align: top;}
	td {text-align: left; vertical-align: bottom;}	

Result:	The th element text is centered horizontally and at the top of the cell and the td element text is set horizontally to the left and aligned with the bottom of the cell.	

21.6- Table Padding:

You can control the space between the content and border of your th and td elements by applying the PADDING property to your style code. This allows for greater readability in your table.

Example:	th {padding: 10px;}
	td {padding: 20px;}	

Result:	Sets the th element to have a blank space of 10 pixels around the content inside the border and sets the td element to have a blank space of 20 pixels around the content inside the border.	

ACTIONS- CSS TABLES

BORDERS:

1. Type: table, th, td {border: X Y Z;}
   (where "X" is the size of your border in pixels, "Y" is the style of your border: solid, double, groove, ridge, inset, outset, dotted *or* dashed and "Z" is the color using color name or a hexadecimal value)

COLLAPSED BORDERS:

1. Type: table {border-collapse: collapse;}
2. Type: table, th, td {border: X Y Z;}
   (where "X" is the size of your border in pixels, "Y" is the style of your border: solid, double, groove, ridge, inset, outset, dotted *or* dashed and "Z" is the color using color name or a hexadecimal value)

TABLE WIDTH AND CELL HEIGHT:

1. Type: table {width: X;}
   (where "X" is the width of the table expressed in pixels (px), inches (in), centimeters (cm), millimeters (mm), points (pt) or em or a percentage)
2. Type: th {height: X;}
   (where "X" is the height of the th elements expressed in pixels (px), inches (in), centimeters (cm), millimeters (mm), points (pt) or em)
3. Type: td {height: X;}
   (where "X" is the height of the td elements expressed in pixels (px), inches (in), centimeters (cm), millimeters (mm), points (pt) or em)

TABLE BACKGROUND COLOR:

1. Type: table {background-color: X;}
   (where "X" is the color value by color name or hexadecimal number)
2. Type: th {background-color: X;}
   (where "X" is the color value by color name or hexadecimal number)
3. Type: td {background-color: X;}
   (where "X" is the color value by color name or hexadecimal number)

TABLE TEXT ALIGNMENT:

1. Type: table {text-align: X; vertical-align: Y;}
   (where "X" is the horizontal alignment: left, right, center and "Y" is the vertical alignment: top, bottom, center)
2. Type: th {text-align: X; vertical-align: Y;}
   (where "X" is the horizontal alignment: left, right, center and "Y" is the vertical alignment: top, bottom, center)
3. Type: td {text-align: X; vertical-align: Y;}
   (where "X" is the horizontal alignment: left, right, center and "Y" is the vertical alignment: top, bottom, center)

ACTIONS-
CSS TABLES

PADDING:

1. Type: th {padding: X;}
   (where "X" is the padding in pixels)
2. Type: td {padding: X;}
   (where "X" is the padding in pixels)

EXERCISES- CSS TABLES

Purpose:

1. To change the border styling and background color of the table you created in a previous exercise.

Exercises:

1. Open you .css document.
2. On the last line type: table {border-collapse: collapse;}
3. Press “Enter”.
4. Type: table, th, td {border: 2px solid blue;}
5. Press “Enter”.
6. Type: th {background-color: yellow;}
7. Press “Enter”.
8. Type: td {background-color: #808080;}
9. Press “Enter”.
10. Save your document.

CHAPTER 22- Working with Transforms in CSS

22.1- What are Transforms?:

The CSS TRANSFORM property allows you to turn, spin, stretch, scale and move your elements in two or three dimensions. This will give your webpage more visual interest and allows you to position elements where you want them to be.

When using the TRANSFORM property, you need to use a different syntax for the Web browsers Chrome, Safari and Opera, as they do not support the basic syntax. To allow your code to work with these browsers you need to supply a prefix to the TRANSFORM property. This prefix is: -webkit-.

You do not need to add an additional line of code. The separate coding can be contained in the same set of curly brackets as your basic TRANSFORM coding. Your code will look like the following example (using the translate method as an example): {transform:translate(50px,100px); -webkit-transform:translate(50px,100px);}.

22.2- 2D Transforms:

When using the 2D TRANSFORM property there are many different values you can ascribe to make changes to your elements. The syntax to achieve a 2D transform is as follows: {transform: *method*(X-axis value,Y-axis value);} (**NOTE:** There cannot be a space between any of the different sections of the coding when using transform).

There are eleven different methods you can use with the 2D TRANSFORM property, they are listed in the table below.

Method	**Description**
translate(x,y)	Sets the rule to translate (move) the element along the X-axis and the Y-axis (x,y defined in pixels)
translateX(n)	Sets the rule to translate the element along the X-axis only (n defined in pixels)
translateY(n)	Sets the rule to translate the element along the Y-axis only (n defined in pixels)
scale(x,y)	Sets the rule to scale the element by changing its width and height (x defines the multiples of width, and y defines the multiples of height)
scaleX(n)	Sets the rule to change the elements width only (n describes the multiples of width)
scaleY(n)	Sets the rule to change the elements height only (n describes the multiples of height)
rotate(angle)	Sets the rule to rotate the element to the specific angle defined (angle is defined in degrees (deg))
skew(x-angle,y-angle)	Sets the rule to skew the element along the X-axis and Y-axis (x-angle, y-angle defined in degrees (deg))
skewX(angle)	Sets the rule to skew the element along the X-axis by the defined angle (angle defined in degrees (deg))
skewY(angle)	Sets the rule to skew the element along the Y-axis by the defined angle (angle defined in degrees (deg))
matrix(n,n,n,n,n,n)	The matrix method combines all 2D transform methods into one, using six parameters, containing mathematic functions, to allow you to rotate, scale, translate and skew elements.

22.2- 2D Transforms- (cont.):

Method:	translate	Moves element relative to its original position
	rotate	Rotates element to the defined angle
Example	div {transform:translate(50px,100px); -webkit-transform:translate(50px,100px);}	
Result:	Translates (moves) div element 50 pixels from the left and 100 pixels from the top from it's original position	
Example:	div.one {transform:rotate(30deg); -webkit-transform:rotate(30deg);}	
Result:	Rotates div.one element 30 degrees clockwise	

22.3- 3D Transforms:

Where 2D transforms interact with only the x- and y-axes, the 3D transforms also act along the z-axis of your webpage. For example, the rotate method in 3D, set at 180 deg, will show the "back-side" or a reverse of your element, as opposed to your element appearing upside down.

Along with the basic TRANSFORM property, there are several more properties available to use with 3D transforms. They are listed in the following table.

Property	**Description**
transform-origin	Sets the rule to allow you to change the position of transformed elements
transform-style	Sets the rule to specify how nested elements are rendered in 3D space
perspective	Sets the rule that specifies the perspective of 3D elements
perspective-origin	Sets the rule that specifies the bottom position of 3D elements
backface-visibility	Sets the rule on whether an element should be visible when it's not facing the screen

22.3- 3D Transforms- (cont.):

While some of the methods used are the same in 2D and 3D transforms, many of them are different or require more information. The following table includes the 3D transform methods.

Method	Description
translate3d(x,y,z)	Sets the rule to define a 3D translation (x,y,z defined in px)
translateX(x)	Sets the rule to define a 3D translation for the x-axis only (x defined in px)
translateY(y)	Sets the rule to define a 3D translation for the y-axis only (y defined in px)
translateZ(z)	Sets the rule to define a 3D translation for the z-axis only (z defined in px)
scale3d(x,y,z)	Sets the rule to define a 3D scale transformation (x.y.z defined by multiples of width (x), height (y) and depth (z))
scaleX(x)	Sets the rule to define a 3D scale transformation on the x-axis only
scaleY(y)	Sets the rule to define a 3D scale transformation on the y-axis only
scaleZ(z)	Sets the rule to define a 3D scale transformation on the z-axis only
rotate3d(x,y,z angle)	Sets the rule to define a 3D rotation (x,y,z.angle defined by deg)
rotateX(angle)	Sets the rule to define a 3D rotation along the x-axis only
rotateY(angle)	Sets the rule to define a 3D rotation along the y-axis only
rotateZ(angle)	Sets the rule to define a 3D rotation along the z-axis only
perspective(n)	Sets the rule to define a perspective view for a 3D transformed element (n is expressed in px)
matrix3d(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)	Sets the rule to define a 3D transformation using a 4x4 matrix of 16 values, containing mathematical functions.

Method:	rotatex	Rotates element along x-axis
Example	div {transform:rotatex(180deg); -webkit-transform:rotatex(180deg);}	
Result:	Displays the div element rotated 180 degrees along the x-axis, so the element shows “upside-down”	

ACTIONS-
WORKING WITH TRANSFORMS IN CSS

2D TRANSFORMS:

1. Type: {transform:X(Y,Z);}
   (where "X" is the method you want to apply (see table), "Y" and "Z" are the values attributed to the method)
2. Type: {-webkit-transform:X(Y,Z);}
   (where "X" is the method you want to apply (see table), "Y" and "Z" are the values attributed to the method)

3D TRANSFORMS:

1. Type: {transform:X(Y,Z);}
   (where "X" is the method you want to apply (see table), "Y" and "Z" are the values attributed to the method)
2. Type: {-webkit-transform:X(Y,Z);}
   (where "X" is the method you want to apply (see table), "Y" and "Z" are the values attributed to the method)

EXERCISES- WORKING WITH TRANSFORMS IN CSS

Purpose:

1. To rotate an image, using 2D transform, in the HTML document created in previous exercises.
2. To rotate an image, on the Y-axis, using 3D transform.

Exercises:

1. Open your HTML document and give an image of your choice the class=“rotate”
2. Give a different image the class=”rotate3D”
3. Save your document.

Exercises:

1. Open your .css document created in previous exercises.
2. On the last line of your document type: img.rotate {transform:rotate(180deg); -webkit-transform:rotate(180deg);}
3. Press “Enter”.
4. Type: img.rotate3D {transform:rotatey(180deg); -webkit-transform:rotatey(180deg);}
5. Press “Enter”.
6. Save your document.

CHAPTER 23-
Transitions and Animations in CSS

23.1- Transitions

23.2- Animations

23.1- Transitions:

Transitions allow you to change the state of an element, over a specific time, without having to use any other programs like Flash or JavaScript. For example, you can change the width of an element when a user hovers over it. Transitions are heavily dependent on the timing of the change. The default for TRANSITION-DURATION property is zero, so it is important to set a time, otherwise your transitions will not work. There are several transition properties, listed in the table that follows.

You can apply transitions to any property in your HTML coding that can be translated into a mathematical reference. For example, you can transition the font-weight of your text but not the font-face.

The syntax for transitions is as follows: {transition: property_name duration timing_function delay;}. This example is for setting all the properties in one line (shorthand), where each property is listed separately using the full property name, separated by semicolons in the same curly brackets, as with any other CSS coding.

Property	**Description**
transition	Sets the rule to change the setting for the four transition properties into a single property.
transition-property	Sets the rule to name the property the transition is applied to. (for example; width, font-weight, height, border, etc.)
transition-duration	Sets the rule to define the duration of the transition. (in seconds (s) or milliseconds (ms))
transition-timing-function	Sets the rule to define how the speed of transition is calculated. (ease (default), linear, ease-in, ease-out, ease-in-out, cubic-bezier())
transition-delay	Sets the rule to define when the transition will start. (default is 0, in seconds (s) or milliseconds (ms))

Example	div {width:100px; height:100px; background: blue;
	transition:width height; transition-duration:2s;}
	div:hover {width: 200px; height:200px;}	

Result:	Sets the div element to a size of 100px by 100px with a background color in blue and sets the transition of width and height to 200 px by 200 px to be changed over two seconds, when user hovers over element.	

23.2- Animations:

Animations allow you to add visual interest to your webpage by creating elements and images that move around the page, spin, change colors, etc. automatically. Animations allow you to gradually change an element from one style to another. You can change as many styles as you would like and as many times as you want in an element. When using animations you have options for specifying when the change will happen: either in percentages 0% to 100% or “from” and “to”, which is the same as the percentages.

To use animations in CSS you have to use the @keyframes rule. This is where you create the animation by specifying the CSS style properties inside the @keyframes rule. This allows the style to change from the current style to the new style.

Like transitions, you must define durations for your animations otherwise they will not run. You also must “bind” an animation to an element or it will not work. You bind an animation to an element by defining the animation name and duration inside the styling rules of your element.

Property	Description
@keyframes	Sets the rule to specify the animation.
animation	Sets the rule to define the property for all animation properties except the animation-play-state property.
animation-name	Sets the rule to specify the name of the @keyframes animation
animation-duration	Sets the rule for how long an animation takes to complete one cycle (value in seconds (s) or milliseconds (ms) default is 0)
animation-timing-function	Sets the rule to describe how the animation will progress over one cycle (ease (default), linear, ease-in, ease-out, ease-in-out, cubic-bezier())
animation-delay	Sets the rule to define when the animation will start (in seconds (s) or milliseconds (ms), default is 0)
animation-iteration-count	Sets the rule to define the number of times an animation is played. Default setting is 1, a number can be given or the term infinite
animation-direction	Sets the rule to define how the animation should play (normal, reverse, alternate, alternate-reverse, default is normal)
animation-play-state	Sets the rule to define whether the animation is running or paused, the default setting is running.

23.2- Animations- (cont.):

Example	div {width:150px; height:150px; background:red;
	position:relative; animation:myfirst 5s infinite;}
	@keyframes myfirst
	{
	0% {background:red; left:0px; top:0px;}
	25% {background:green; left:200px; top:0px;}
	50% {background:#c0c0c0; left:200px; top:200px;}
	75% {background:black; left:0px; top:200px;}
	100% {background:red; left:0px; top:0px;}
	}	

Result:	Sets the div element with a size of 150px by 150px, a starting background color of red and an animation time of five seconds. The @keyframes rule defines the animation to start at the original position, then move 200 px right and gradually change to green, move 200px down and gradually change to silver, move 200 px to the left and gradually change to black, then back to the start point and gradually change back to red and then continue infinitely.	

ACTIONS- Transitions and Animations in CSS

TRANSITIONS:

1. Type: A {B:C; D:E; transition: F G; transition-duration:H;}
   (where "A" is the element you want to transition, "B" and "D" are the element properties, "C" and "E" are the values of the element properties, "F" and "G" are the properties you wish to transition (ie: width, height, etc.) and "H" is the duration of the transition in seconds (s) or milliseconds (ms))
2. Type: A:B {C:D; E:F;}
   (where "A" is the element and "B" is the action (ie. :hover), "C" and "E" are the element properties that are transitioning and "D" and "F" are the element property values)

ANIMATIONS:

```
Type: @keyframes A
        {0%   {B:C;}
        25%  {B:C;}
        50%  {B:C;}
        75%  {B:C;}
        100% {B:C;}}
```

(where "A" is the name of your @keyframes rule, "B" is the property you want to change and "C" is the value) **NOTE:** you can have as many properties and values as you would like to change listed here separated by a semicolon)

2. Type: A {B:C; D:E; animation-name:F; animation-duration:G;}
   (where "A" is the element you are animating, "B" and "D" are the properties of your element, "C" and "E" are the values (these are the base properties of your element and can be as many or as few as you would like), "F" is the name of your @keyframes rule and "G" is the duration of your animation.

EXERCISES-
Transitions and Animations in CSS

Purpose:

1. To transition a link on hover.
2. To animate the background color of a link.

Exercises:

1. Open your HTML document and give a link the class, class="trans".
2. Select another link and give it the class, class="animate".
3. Save your document.

Exercises:

1. Open your .css document created in previous exercises.
2. On the last line of your document type: a.trans {color:blue; transition:all 0.3s ease;}
3. Press "Enter".
4. Type: a.trans:hover {color:green;}
5. Press "Enter".
6. Type: @keyframes linkani
7. Press "Enter".
8. Type: {0% {background:blue;} 25% {background:red;} 50% {background:silver;} 100% {background:fuchsia;}
9. Press "Enter".
10. Type: a.animate {color:black; animation:linkani; 5s infinite;}
11. Press "Enter".
12. Save document.

CHAPTER 24-
CSS SHORTHAND

24.1- SHORTHAND PROPERTIES

24.1- Shorthand Properties:

CSS provides a few Shorthand Properties that allow you to group selectors within a single declaration instead of a separate one for each property, thus reducing the time and effort in coding, as well as reducing the size of your CSS file itself. To create the Shorthand code, you separate the selectors with a space within the declaration. It's important to note that you should always declare a value for each of the selectors in Shorthand code. If you do not define each selector in the correct order, web browsers will either ignore the declaration altogether, or use the default setting. So, be cautious when using Shorthand Properties, otherwise you may not achieve your desired outcome. The syntax for Shorthand Properties is as follows:

{Shorthand Property Name: Selector Selector;}

For Example, you can use:
{margin: 2px 1px 3px 4px;}

Instead of:
{margin-top:2px; margin-right:1px; margin-bottom:3px; margin-left:4px;}

The following table lists available Shorthand Properties and their selectors in the order they should appear:

Shorthand Property	**Selectors (Do not omit any and keep in order!)**
font	font-size; line-height; font-weight; font-style; font-family
background	background-color; background-image; background-repeat; background-position
list-style	list-style; list-style-type; list-style-position; list-style-image
border	border-width; border-color; border-style
margin	margin-top; margin-right; margin-bottom; margin-left
padding	padding-top; padding-right; padding-bottom; padding-left

The margin and padding Shorthand properties work a little differently due to the nature of the properties (four sides) that can result in different combinations of values. For example, you can have a different value for each of the four margins (top, right, bottom and left) as in the example above. Or, you might have just two different values for the four total sides. This type of shorthand notation can take one, two, three, or four values. If you specify four values, they are assigned to the appropriate sides in order (clockwise – top, right, bottom, left). If you only specify two or three values, the "missing" side is assigned the same value as the one opposite it. If you only specify a single value, it will be applied to all four sides. For example, the declaration "margin: 3px 1px" would assign a value of 3 pixels to the top and bottom margins and 1 pixel to the left and right margins (since only two values are listed in the declaration).

Appendix- HTML Basic Tags

Basic Tags & Attributes	
Tag/Attribute	***Description***
<!-->	Insert a comment
<body>	Identifies content of a webpage
<br>	Line break
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>	Heading levels
<head>	Contains information about a webpage title, style sheets, and search engine keywords.
<!DOCTYPE html>	Allows browsers to know what version of HTML your code is written in.
<html>	Identifies an HTML document
<meta>	Contains extra information about a page
CONTENT	Sets custom information about a page
NAME	Adds a description or copyright information to a webpage.
<p>	Starts a new paragraph
<title>	Creates a title for a webpage
<span>	Used to group inline elements and also to change specific items within an element. i.e. color a specific word in a sentence or paragraph

Appendix- HTML Text Formatting Tags

Basic Tags & Attributes	
Tag/Attribute	***Description***
<b>	Bold
<blockquote>	Separates text form the main text
<i>	Italics
<nobr>	Keeps all text on a single line, without line breaks
<pre>	Keeps preformatted spacing in text
<q>	Quotes short passages of text
<small>	Makes text smaller than surrounding text
<u>	Underlines text

Appendix- HTML List Tags

Basic Tags & Attributes	
Tag/Attribute	***Description***
<dd>	Identifies Definition List
<dl>	Creates a definition list of terms
<dt>	Identifies a term in a definition list
<li>	Identifies an item in an ordered or unordered list
<ol>	Creates an ordered list
START	Start number for an ordered list
TYPE	Seta a number style for an ordered list
<ul>	Creates an unordered list
TYPE	Sets a bullet style for an unordered list

Appendix- HTML Link Tags

Basic Tags & Attributes	
Tag/Attribute	***Description***
<a>	Creates a link
HREF	Specifies the location of a webpage or other document or file
ID	Identifies or names a webpage area displayed by selecting a link
TARGET	Specifies where linked information appears
<ol>	Creates an ordered list
START	Start number for an ordered list
TYPE	Seta a number style for an ordered list
<ul>	Creates an unordered list
TYPE	Sets a bullet style for an unordered list

Appendix-
HTML Image Tags

Basic Tags & Attributes	
Tag/Attribute	***Description***
<img>	Inserts an image on a page
ALT	Displays alternative text when an image does not load
BORDER	Adds a border to an image
HEIGHT	Controls the height of an image
SRC	Specifies the image location or path
WIDTH	Controls the width of an image
TITLE	Displays a text title when mouse is hovering over image

Appendix-
HTML Table Tags

Basic Tags & Attributes	
Tag/Attribute	***Description***
<table>	Creates a table
BORDER	Adds a border to a table
BORDERCOLOR	Defines the color of the border
<tr>	Defines rows in a table
<th>	Adds a header to a table

Appendix-
HTML Iframe Tags

Basic Tags & Attributes	
Tag/Attribute	***Description***
<iframe>	Creates an Iframe
HEIGHT	Defines the height of the Iframe
WIDTH	Defines the width of the Iframe
NAME	Defines the name of the Iframe
TARGET	Sets link to open in the named Iframe

Appendix- HTML Form Tags

Basic Tags & Attributes	
Tag/Attribute	***Description***
<form>	Creates a form
ACTION	Identifies the location of a CGI script for a form
METHOD	Identifies how form information transfers to a Web server
<input>	Creates an input item on a form
CHECKED	Automatically selects a radio button or a check box
ENCTYPE	Specifies file transfer for form data over the Internet
MAXLENGTH	Identifies the maximum number of characteristics for a form entry
NAME	Identifies a form item for a server
SIZE	Identifies the form item size
TYPE	Identifies the form item type
VALUE	Identifies a form item
<label>	Used to label form elements
FOR	Identifies which form element the label is associated with
<option>	Creates a menu option for a form
SELECTED	Automatically selects a menu item
VALUE	Identifies a form item for a server
<select>	Creates a menu item on a form
NAME	Identifies a form menu for a server
SIZE	Specifies the number of menu options
<textarea>	Creates a large text area on a form
COLS	Specifies a width for a large text area
NAME	Identifies a form text area for a server
ROWS	Specifies a height for a large text area
WRAP	Wraps text within a large text area

Appendix-
HTML Video and Audio Tags

Basic Tags & Attributes	
Tag/Attribute	***Description***
<video>	Adds video to a webpage
<audio>	Adds audio to a webpage
HEIGHT	Defines height of video file
WIDTH	Defines width of video file
<source>	Defines video or audio source
SRC	Path of file name of video or audio file
TYPE	Defines type of video or audio file
CONTROL	Adds play, pause, etc. controls to an audio or video file
<embed>	Used for older browsers in conjunction with <audio> or <video> to add files to webpage
<a>	Creates a clickable link to audio or video files
HREF	Specifies the location of the linked audio or video files

Appendix- Named Colors

Named Colors	
Color	***Name***
Aqua	#00FFFF
Black	#000000
Blue	#0000FF
Fuchsia	#FF00FF
Gray	#808080
Green	#008000
Lime	#00FF00
Maroon	#800000
Navy	#000080
Olive	#808000
Purple	#800080
Red	#FF0000
Silver	#C0C0C0
Teal	#008080
White	#FFFFFF
Yellow	#FFF00

Appendix- Style Sheet Characteristics

User Interface Properties	
Characteristic	***Description***
box-sizing	Sets the rule to define elements to fit in an area in a certain way
content	Used with the pseudo-elements :before and :after, to insert generated content
cursor	Sets the rule to define type of cursor to be used
icon	Set the rule to allow author to style an element with an equivalent icon
nav-down	Sets rule to define where to navigate when using down arrow key
nav-index	Sets the rule to define tabbing order of elements
nav-left	Sets the rule to define where to navigate when using left arrow key
nav-right	Sets the rule to define where to navigate when using right arrow key
nav-up	Sets the rule to define where to navigate when using up arrow key
outline	Sets rule to define all outline properties
outline-color	Sets the rule to define outline color
outline-offset	Sets rule to offset and outline and draw it beyond a border edge
outline-style	Sets the rule to define the style of outline
outline-width	Sets the rule to define width of outline
resize	Sets the rule to define whether an element can be resized by user
text-overflow	Sets the rule to define what happens when text overflows containing element

Appendix-
Style Sheet Characteristics

Background and Border Properties	
Characteristic	***Description***
background	Sets the rule to define all background properties in one set
background-attachment	Sets the rule for a fixed or scrolling background
background-color	Sets the rule to define a background color for an element
background-image	Sets the rule to assign an image as a background for an element
background-position	Sets the rule for the starting position of a background image
background-repeat	Sets the rule for how a background image will repeat
background-clip	Sets the rule to specify the painting area of a background image
background-origin	Sets the rule for the positioning area of a background image
background-size	Sets the rule for the size of the background image
border	Sets the rule to define all border properties in one set
border-bottom	Sets the rule to define all bottom border properties
border-bottom-color	Sets the rule to define bottom border color
border-bottom-left-radius	Sets the rule to define the shape of bottom left corner of border
border-bottom-right-radius	Sets the rule to define the shape of bottom right corner of border
border-bottom-style	Sets the rule to define the style of bottom border
border-bottom-width	Sets the rule to define the width of the bottom border
border-color	Sets the rule to define the color of all four borders
border-image	Sets the rule to define all border-image- properties in one set
border-image-outset	Sets the rule to define the amount the border image extends outside the border box
border-image-repeat	Sets the rule to define image-border as repeated rounded or stretched
border-image-slice	Sets the rule to define the inward offset of the image-border
border-image-source	Sets the rule to define the image used as a border

Background and Border Properties Cont.	
Characteristic	***Description***
border-image-width	Sets the rule to define the image width used as border
border-left	Sets the rule to define all properties for left border in one set
border-left-color	Sets the rule to define the color of left border
border-left-style	Sets the rule to define style of left border
border-left-width	Sets the rule to define the width of left border
border-radius	Sets the rule to define the shape of all corners of border in one set
border-right	Sets the rule to define all right border properties in one set
border-right-color	Sets the rule to define color of right border
border-right-style	Sets the rule to define style of right border
border-right-width	Sets the rule to define width of right border
border-style	Sets the rule to define style of all borders in one set
border-top	Sets the rule to define all to border properties in one set
border-top-color	Sets the rule to define color of top border
border-top-left-radius	Sets the rule to define the shape of top left corner of border
border-top-right-radius	Sets the rule to define the shape of top right corner of border
border-top-style	Sets the rule to define the style of top border
border-top-width	Sets the rule to define width of top border
border-width	Sets the rule to define width of all four borders
box-decoration-break	Sets the rule to define behavior of the background and border of an element at page-break, or, for in-line elements, at line-break
box-shadow	Sets the rule to attach box shadows to a box

Appendix-
Style Sheet Characteristics

Font Properties	
Characteristic	***Description***
font	Defines all font properties in one set
font-family	Defines the font-family for text
font-size	Defines the size of the font for text
font-style	Defines the style of font for text
font-variant	Defines whether font should be displayed in small-cap format
font-weight	Defines the weight of the font
@font-face	Sets the rule to allow webpages to download fonts that aren’t “web-safe”
font-size-adjust	Sets the rule to allow font to preserve size regardless of font family
font-stretch	Sets the rule to define normal, condensed or expanded face from a font family

Appendix- Style Sheet Characteristics

Text Properties	
Characteristic	***Description***
direction	Sets the rule to specify text direction
hanging-punctuation	Sets the rule to allow punctuation character to be placed outside a line box
hyphens	Sets the rule to define how to split words in paragraphs
letter-spacing	Sets the rule to define space between character in text (kerning)
line-height	Sets the rule to define line spacing (leading)
tab-size	Sets the rule to define length of tab-character
text-align	Defines the horizontal alignment of text
text-align-last	Defines how a line before a forced line break is aligned when text-align is set to "justify"
text-indent	Defines the indentation of first line of paragraph
text-justify	Sets the rule to define justification of text
text-transform	Sets the rule to define capitalization of text
unicode-bidi	Used in conjunction with direction property to set rule to allow text to be overridden to support multiple languages in same document
white-space	Sets the rule to define text wrapping in an element
word-break	Sets the rule for line breaking of non-CJK scripts (CJK scripts are Chinese, Japanese Korean)
word-spacing	Sets the rule to increase or decrease white space between words
word-wrap	Sets rule to allow long, un-breakable words to break and wrap

Text Decoration Properties	
Characteristic	***Description***
text-decoration	Sets the rule to define the decoration of text
text-decoration-color	Sets the rule to define color of text decoration
text-decoration-line	Sets the rule to define type of line in text decoration
text-decoration-style	Sets the rule to define style of line in text decoration
text-shadow	Sets the rule to add shadow to text

Appendix- Style Sheet Characteristics

Basic Box Layout Properties	
Characteristic	***Description***
Bottom	Sets the rule to define bottom position of positioned element
clear	Sets the rule to specify which sides of an element will not allow other floating elements
clip	Sets rule to clip an absolutely positioned element
display	Sets rule to define how a specific HTML element is displayed
float	Sets rule to allow an element to float left or right
height	Define the height of an element
left	Sets the rule for left position of an element
overflow	Sets rule to define what happens if content overflows element's box
overflow-x	Defines whether or not to clip left or right edges of content that overflows an element's box
overflow-y	Defines whether or not to clip top or bottom edges of content that overflows and element's box
padding	Sets rule to define all padding properties in one set
padding-bottom	Sets rule to define bottom padding on an element
padding-left	Sets rule to define left padding of an element
padding-right	Sets rule to define right padding of an element
padding-top	Sets rule to define top padding of an element
position	Sets rule to define type of positioning method for an element
Right	Defines right position of a positioned element
top	Defines top position of a positioned element
visibility	Sets rule to define whether an element is visible
width	Sets rule to define width of an element
vertical-align	Sets rule to define vertical alignment of an element
z-index	Sets rule to define stack order of a positioned element

Flexible Box Layout Properties	
Characteristic	***Description***
align-content	Sets rule to define alignment between lines in a flexible container when all space is not used
align-items	Sets rule to define alignment inside a flexible container
align-self	Sets rule to define alignment for selected items inside a flexible container
display	Sets rule to define how an HTML element is displayed
flex	Sets rule to define length of an item relative to the rest
flex-basis	Sets rule to define initial length of a flexible item
flex-direction	Sets rule to define direction of a flexible item
flex-flow	Defines values for flex-direction and flex-wrap in one set
flex-grow	Sets rule to define how much item will grow relative to the rest
flex-shrink	Sets rule to define how much item will shrink relative to the rest
flex-wrap	Sets rule to define whether flexible items wrap or not
justify-content	Sets rule to define alignment between items in a flexible container when all space is not used
margin	Sets rule to define all margin properties
margin-bottom	Defines bottom margin of an element
margin-left	Defines left margin of an element
margin-right	Defines right margin of an element
margin-top	Defines top margin of an element
max-height	Sets rule to define maximum height of an element
max-width	Sets rule to define maximum width of an element
min-height	Sets rule to define minimum height of an element
min-width	Sets rule to define minimum width of an element
order	Sets rule to define order of the flexible item, relative to the rest

Table Properties	
Characteristic	***Description***
border-collapse	Defines whether table borders should be collapsed
border-spacing	Sets rule to define distance between border or adjacent cells
caption-side	Sets rule to define placement of table caption
empty-cells	Sets rule to define whether border and background of empty cells are displayed
table-layout	Sets the rule to define layout calculation used for a table

List Properties	
Characteristic	***Description***
list-style	Sets rule to define all list properties
list-style-image	Sets the rule to define image as list item marker
list-style-position	Sets rule to define position of list markers, inside or outside content flow
list-style-type	Sets the rule to define type of list item marker

2D and 3D Transform Properties	
Characteristic	***Description***
backface-visibility	Sets rule to define whether an element should be visible when it is not facing the screen
perspective	Sets the rule to define the perspective of 3D elements
perspective-origin	Sets the rule to define the bottom position of 3D elements
transform	Sets the rule to apply a 2D or 3D transform to elements
transform-origin	Sets the rule to define the position of transformed elements
transform-style	Sets the rule to define how nested elements are rendered in 3D space

Transition Properties	
Characteristic	***Description***
transition	Sets the rule to define all four transition properties
transition-property	Sets the rule to define the name of the CSS property to apply transition effect to
transition-duration	Sets the rule to define how long the transition effect is to take
transition-timing-function	Defines the speed curve of the transition effect
transition-delay	Define when the transition effect is to start

Appendix- Style Sheet Characteristics

Animation Properties	
Characteristic	***Description***
@keyframes	Sets the rule to specify the animation
animation	Sets the rule to define all animation properties except animation-play-state
animation-delay	Sets the rule to define when animation is to start
animation-direction	Sets the rule to define whether the animation should play in reverse or alternate cycles
animation-duration	Sets the rule to define the length of time to complete one animation cycle
animation-fill-mode	Defines what values are applied by the animation outside the time it takes to execute
animation-iteration-count	Sets the rule to define the number of times an animation repeats
animation-name	Sets the rule to define a name for the @keyframes animation
animation-timing-function	Defines the speed curve of the animation
animation-play-state	Sets the rule to define whether the animation is to run or not

Appendix- Style Sheet Characteristics

Pseudo-Class/Pseudo-Element Properties	
Characteristic	***Description***
:link	Sets rule to select all unvisited links
:visited	Sets rule to select all visited links
:hover	Sets rule to select links on mouse over **(a:hover MUST come after a:link and a:visited to be effective)**
:active	Sets rule to select the currently active link **(a:active MUST come after a:hover to be effective)**
:focus	Sets rule to select the input element that is currently being used
:first-letter	Sets the rule to select the first letter of every paragraph element
:first-line	Sets the rule to select the first line of every paragraph element
:first-child	Sets the rule to select every paragraph element that is the first child of its parent element
:before	Sets the rule to insert content before every paragraph element
:after	Sets the rule to insert content after every paragraph element
:lang	Sets the rule to select every paragraph element with a lang="was" attribute