Introduction HTML

Introduction to HTML/XHTML
Handout Companion to
the Interactive Media Center's
Online Tutorial

HTML (HTML5) Language & Encoding

This document is the handout version of the Interactive Media Center's online tutorial Introduction to HTML/XHTML. It may be found at http://library.albany.edu/imc/html_tut/.

About HTML/XHTML

HTML (Hyper Text Markup Language) and its more recent incarnation, XHTML HTML (Hyper Text Markup Language) and its more recent incarnation, XHTML (Extensible Hyper Text Markup Language) is the central core on which all Web pages are built. Although there are new types of technology and coding options, including PHP, ASP, XML and others, traditional markup language is the place where "all things Web" starts.

At this time the latest version of HTML is called "HTML5" and it is coming into use, although it is not "official" yet. Each browser update seems to incorporate more HTML5 capabilities. In the tutorial we will discuss many aspects of HTML, XHTML and HTML5. In most instances there is no need to differentiate between the three. However, when the requirement arises the variations will be noted.

Some Web editing software publishers claim that it is not necessary to know HTML if you use their programs to build Web pages. However, this is rarely the case in actual practice. In almost every instance at least some knowledge of HTML is helpful.

In this tutorial we will guide you through the basics of writing HTML code. It is important to realize that there is no better way to learn and understand the code than to

actually write and use it. This tutorial will get you started on your first steps toward becoming knowledgeable about the code.

Technique

Many people try to write Web page code as if they were creating a term paper. They start at the beginning and continue on through the middle to the end in a linear fashion. HTML/XHTML code consists of two types of tags. There are the container tags and the single part tags, often called "non-container" tags.

Container tags have a opening part and a closing part with the "stuff" that the tag controls inside. An example is the tag that creates bold text, . is the opening part and is the closing part.

If we write a code snippet it might look like this, "This is bold" and you can see the bold text in between the tag parts.

An example of a single-part tag is the line break, <br /> which merely stops a line of text at a specific point. This is used where a line must end after a particular word.

Because Web pages will have lots of code, trying to remember which tag has been opened or closed can be extremely difficult. The best technique for writing code is to make both the opening and closing parts at the same time, and THEN insert the material between the tags. This prevents the problems that arise when a tag is accidentally left unclosed and is the technique that will be used in this tutorial.

Single-part tags can be inserted as required anytime.

Tools

Writing HTML and XHTML code can be done simple word processor. But it is important to note the average code write should never use a processor such as Word, WordPerfect, OpenOffice, or similar. These "high end" word processors are too smart and will try to insert their own formatting codes.

The best software to use is also the most basic. NotePad is a perfectly suitable tool for writing HTML. Some programs that are specifically designed to create Web pages, such as Dreamweaver, have special HTML writing functions that allow you to create straight code. There are also programs such as HTMLKit that function as "code insertion" programs where the user doesn't necessarily write the code, but rather instructs the program to insert blocks of HTML or XHTML. Even though these programs make writing code easier, it is still necessary to know the code.

In this tutorial we suggest that you use NotePad or a similar program.

Web vs. Print

We know that trying to do the tasks and follow the examples online at the same time may be difficult. We have a printed version of this tutorial that you may download here. You may then read printed version and perform the steps on your computer.
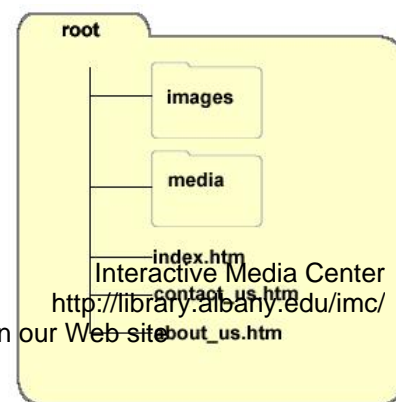
Getting Started

To get started with you Web project you need to set up your site with the proper structure. Site structure is very simple. Your project has a "root folder" that contains all of the files associated with the site. Inside the root folder (which, by the way, can have any name you wish, it doesn't have to be called "root") will be sub folders for

root

images

media

index.htm
contact_us.htm
about_us.htm

Interactive Media Center
http://library.albany.edu/imc/

This and other handouts are available online on our Web site

specific types of files. While the HTML Web pages will be "loose" in the root folder, all the images will be in an image folder, PDF files in a PDF folder, and so on. The diagram on the right illustrates a standard Web site structure with three HTML pages and two sub folders.

One you have your site set up on your computer with a root folder and the necessary sub folders you can start to build your first Web page. This will probably be your home page. Unless instructed otherwise, your home page will have one of these names:

index.htm
index.html
default.htm
default.html

The exact name will be specified by your Web master maintaining the host server for your site. Most Web hosting servers have a hierarchy of names. When a site visitor types into his or her browser the URL of a Web site, the browser is directed to the host server. The host server automatically looks for the name of the home page based on the order of the list, and when it is found it sends the page and all of its associated files to the visitor's computer where the browser assembles it on the screen.

If the server doesn't find the preferred home page name in the collection of HTML files it automatically looks for the second file name in the hierarchy. If that page isn't found it looks for the third page name, and so on. It is helpful to giver your home page the preferred home page name on your host server so that the time lost to the search process is reduced.

HTML Horizontal Line || <hr> tag

Open a new document with NotePad (or whichever program you decide to use) and then save it in your root folder with one of the home page names noted above

Setting Up the Basic Code

A Web page consists of two parts, the head and the body. The head contains information about the page, such as meta tags, when it was made, and the specifications of the code itself. The body contains the material that is seen when the page is viewed. The data in the head is not visible, although it may specify how the "stuff" in the body is displayed. All the tags, including the head and body tags, are located inside what can be called the "ultimate container tags," <html></html>, which tell the browser that everything inside them is for a Web page.

Start by typing the opening and closing HTML tags <html></html> as shown in the illustration on the right.

```
<html>


</html>
```

Now add the opening and closing tags for the head of the HTML page so that the code looks like the example on the right.

```
<html>
<head>
</head>
</html>
```

So far, so good. Now add the body tag.

Save the file.

```
<html>
<head>
</head>
<body>
</body>
</html>
```

Now add some text into your Web page so that you have something to see when you test it. It must be typed in the body, between the opening and closing body tags.

```
<html>
<head>
</head>
<body>
This is my first Web page!
</body>
</html>
```

Now you can start your Web browser, Internet Explorer, Firefox, Mozilla, Safari, Opera, or whatever your favorite is. Open your new Web page and view the file. In most browsers you will click File> Open (or Open File)> then navigate to your saved Web page and click OK.

The illustration on the right shows what your page should look like.

Now you can add some data to the head part of the Web page that will be helpful to your visitors.

Inside the head tags, create a title tag and type in a title for your page. Titles are not names of the files. Rather, they appear in the top bar of a browser and help identify to the visitor which page is being viewed. They may also be used by search engines to determine what a page is about.

```
<html>
<head>
<title>My Web Page</title>
</head>
<body>
This is my first Web page!
</body>
</html>
```

Save and view your page again. You will see your page's title at the very top of the browser.

NOTE: It isn't necessary to follow the File> Open sequence every time you view a saved page. Just make sure your page is saved then click the Refresh or Reload button (depending on which browser you are using.

Adding and Formatting Text

HTML Text Formatting Tags

In this next series of steps you will discover how to add next to your Web page and format for appearance.

You will start by adding a paragraph tag to the body of your Web page. The code is illustrated on the right.

Paragraphs are one of the elements that HTML is designed to identify. Notice that paragraphs are specified with the p tag, <p></p>.

NOTE: What is that text? The text illustrated in the example on the right is often called "Lorem Ipsum" from the first two words in this often quoted sample. It is a piece of classical literature in Latin that is regularly used by printers, graphic artists, and Web designers when "dummy" text is needed to show what a design, such as a Web page or printed brochure, may look like when completed. Because the appearance of the text mimics English, and because most people cannot read Latin, it is a useful tool for showing how a design will look while at the same time avoiding the natural tenancy to read text rather than

```
<html>
<head>
<title>My Web Page</title>
</head>
<body>
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p> </body>

</html>
```

concentrate on overall appearance and usability. An excellent source for Lorem Ipsum is the Web site www.lipsum.com where you can learn all about it and copy examples. For this tutorial you do not have to use Lorem Ipsum. You can type anything you wish.

No doubt by now you have noticed a pattern to the way container tags are formed. The closing part of a container tag always has a forward leaning slash, /, to it. This slash is what tells the browser that the use of the tag has ended and is closed.

Let's format your text a little bit by adding some simple bold to the first few words of the paragraph. Insert the strong tag as shown on the right. How will this look when viewed in a browser? Save the file and test it with your browser.

Notice that only the words inside the strong tag show up as bold. This is how container tags work.

```
<html>
<head>
<title>My Web Page</title>
</head>
<body>
<p><strong>Lorem ipsum dolor</strong> sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
</body>
</html>
```

Another container tag is the emphasis tag that is used to display text as italics. That tag is . Try applying it to a section of text other than the part you bolded earlier. The example on the right will how you how this might be done. You can see a real Web page using the emphasis tag by clicking here.

```
<html>
<head>
<title>My Web Page</title>
</head>
<body>
<p><strong>Lorem ipsum dolor</strong> sit amet, consectetur <em>adipisicing elit</em>, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
</body>
</html>
```

Nesting Tags

Nesting tags refers to the order in which the opening and closing parts of tags appear when multiple tags are applied to a single segment of text or another object. For instance, a portion of text can be both bold and in italics. Proper nesting of the strong and emphasis tags would look like this:

This is bold and italics

The strong tag is on the "outside" while the emphasis tag is on the "inside" and the text is in the middle.

You could also make the tags nest thusly:

This is bold and italics

Improperly nested tags will not have the outside-inside-middle sequence. An example of an improperly nested set of tags is:

This is bold and italics

In the "old days" of the Web improperly nested tags usually did not make much difference. But, as the World Wide Web evolved it became increasingly important for browsers to adhere to common standards.

Today's XHTML has been described as a more strict version of HTML. In many ways that is correct. XHTML rules specify proper nesting, tag writing, and other important points. Modern browsers comply with these rules closely. If your code does not meet the rules of XHTML, it may not display correctly on a monitor.

To find out more about HTML and XHTML rules, visit the World Wide Web Consortium at www.w3x.org. This is the organization that sets the specification of all things associated with the Web.

HTML Div Align Center

Advanced Text Formatting

In addition to creating bold and italicized text, you can also use HTML to control the appearance in more advanced ways. Included is the ability to display text in color, control the font, and set the text size.

Font presentation is controlled using the font tag, . In the example on the right, font size is specified at 3. Font sizes are specified 1 through 7

```
<html>
<head>
<title>My Web Page</title>
</head>
<body>
<p><font size="3">This text is
font size 3.</font></p>
</body>
</html>
```

with 1 as the smallest and 7 the largest.

HTML Paragraphs tag

Size 3 is the default size. That is the size that is "normal" for most browsers under most circumstances.

You can experiment with the font sizes by changing the value using 1 through 7. The list below shows what these sizes look similar to what is shown below. Remember, actual font size will depend on many factors, including the size of the monitor screen.

Font size 1

Font size 2

Font size 3

Font size 4

Font size 5

Font size 6

Font size 7

It is also possible to specify font sizes relative to the settings in the browser. This is done using font size attributes that are placed with "+" or "-". An example is on the right where the size +2 is used. This means that the font should be displayed 2 sizes larger than the browser's default size.

According to the rules for HTML, these font sizes should override the settings that are set in a visitor's browser. In other words, if a visitor to your Web page has customized his or her font sizes in the browser, the use of the size attribute in the font tag will "take over" and the text should appear as you, the page builder, desire. The example on the right illustrates how a "font +size" attribute could be applied.

```
<html>
<head>
<title>My Web Page</title>
</head>
<body>
<p><font size="+2">This text is font size 6.</font></p>
</body>
</html>
```

So, what is a "+" or a "-" exactly? You can't tell! Because each browser could have any font size set for default, it is not possible to determine exactly what the text will look like because it can vary from machine to machine. The examples below will give you some idea, but to see how the appearance you may wish to visit this page later with another browser setting of another machine.

This is font size +1

This is font size +2

This is font size +3

This is font size +4

This is font size +5

This is font size +6

This is font size +7

This is font size -1

This is font size -2

This is font size -3

This is font size -4

This is font size -5

This is font size -6

This is font size -6

It is possible that several of the above examples look identical. This is because some browsers will limit just how large or small font can be displayed. There is really no point in creating text that is unreadable, which can happen if it is too big or too tiny.

Another way to format text is by using color. Color can be specified by using names, such as red, blue, or ForestGreen. They can also be defined using their RGB (**R**ed, **G**reen, **B**lue) values. The most common way for colors to be specified on the Web is by their hexadecimal values. Examples of each method are below.

This is red specified as a color name.

\This is red specified as an RGB value.\

\This is red specified in hexadecimal.\

HTML Text Formatting Tags

Color and the Web

Color plays an important part of Web page design. It can be used for decorative purposes, such as to make something look stylish or flashy, or it can be used to indicate an action or function. For instance, the links on these pages are blue, and that makes them easier to identify.

Color can be applied to text, backgrounds, and certain kinds of borders. It is important to understand both the benefits and pitfalls that color can present. While a Web page with a black background and gray letters may look sophisticated, it will be difficult to read.

There are several excellent tools on the Web to help you obtain a better understanding of color. Some links are:

- ☐ Color and the Web (http://library.albany.edu/imc/webcolor/index.htm), the Interactive Media Center's online tutorial.
- ☐ Browser Safe Color Chart (http://library.albany.edu/imc/webcolors.htm), online chart of Web safe colors from the Interactive Media Center.
- ☐ Color Names Chart (http://www.w3schools.com/html/html_colornames.asp), online chart from the W3 Schools.
- ☐ The Browser Safe Color Pallet (http://www.lynda.com/hex.asp#), online guide from Lynda Weinman.

☐

Download the Best Backgrounds for Websites
Headings
HTML Headings tag

Headings one of the objects on a Web page that HTML and XHTML are intended to identify. Headings tags that are used to specify importance and are defined by using the "h" tag. There are six heading tags ranging from 1 to 6 with 1 being the most important and 6 being the least important.

Imagine that you are making a traditional outline for a paper or article you are writing. It might look something like this:

I. Introduction

A. Welcome

II. Main Message

A. Part 1 - What I did on my summer vacation.

1. Getting to the airport

2. Flying on the airplane.

a. Visiting Alaska.

b. Seeing a bear.

3. Flying home.

B. Summing my vacation.

III. Conclusion

The most important parts of the outline are in Roman numerals, the second most important parts are noted with capitalized letters, and so on. Headings follow a similar pattern. The most important h1 would be comparable to the Roman numerals, the h2 would be second most important as is the capitalized letters. This continues on down through the h6 tag.

Heading tags are container tags, and their format is illustrated below.

<h1>This is heading H1</h1>

When the h1 tag is used in a Web page, the code will resemble the example on the right.

```
<html>
<head>
<title>My Web Page</title>
</head>
<body>
<h1>This is an example of the h1 tag.</h1>
</body>
</html>
```

It a simple Web page the h1 tag would resemble the illustration on the right.

Lists

Lists are also objects that HTML and XHTML are intended to define. The code allows you to specify two types of lists, ordered lists and unordered lists. The primary difference between the types is that the ordered lists indicate importance and unordered lists only indicate that the data is a list, no importance is intended.

The list tags are container tags, but they each have two parts. Part one indicates whether the list is ordered or unordered. Part two is the item in the list itself. Below is an example of an ordered list.

1. Dinner at 8:00 pm.
2. Lunch at noon.
3. Snacks at 11:00 am.

This list indicates that the most important item is dinner at 8:00 pm, the second most important is lunch at noon, and the least important item is snacks at 11:00 am. The code snippet for this list is shown below.

```
<ol>
<li>Dinner at 8:00 pm.</li>
<li>Lunch at noon.</li>
<li>Snacks at 11:00 am.</li>
</ol>
```

Notice that each item is inside the list item tag, and that the whole set of list items is within the ordered list tag .

HTML Lists Tags

An example of an ordered list is below.

- Apples
- Oranges
- grapes

This unordered list does not indicate which fruit is most important. The code snipped for this list shown below.

```
<ul>
<li>Apples</li>
<li>Oranges</li>
<li>grapes</li>
</ul>
```

The unordered list tag is , and as with the ordered list, each item is within the list item tag, .

This and other handouts are available online on our Web site

Many people will use the ordered and unordered list tags for presentational purposes. In other words, when they are not interested in actual lists, but want a group of items with bullets in front of them they will use the unordered list tag. When they want numbers instead of bullets they will use the ordered list. This is not good Web practice. List tags are specifically intended to specify lists and not *how something looks*. This may seem like a trivial point, but there are situations (such as with search engines) that using a list tag incorrectly can cause problems.

In the examples above, if the ordered list tag is used inappropriately, the code would be telling a search engine that oranges are the most important thing on the list when that is perhaps not the message that was desired. It may be that the page designer merely wanted a set of lines with numbers in front of each and had no intention of indicated the importance of oranges. This could cause the page to come up unsatisfactorily when a search is performed.

Links

Links are the most important part of the Web. In fact, the whole premise of the World Wide Web is the ability to link from one document to another. Now, of course, links lead from one site to another, and to specific points within a site.

There are two types of links, absolute and relative. All link tags are container tags.

The way to think of an absolute tag is that it tells exactly where a browser should go by following all the necessary paths. They are used when you link from one Web site to another. Relative links (sometimes called "local links") just tell the shortcuts.

Let's say that we want to make an absolute link to the University at Albany's Web site. The link is inside the A tag and would look like this:

Link to UAlbany

Roughly translated the link says that the browser should use hyper text transfer protocol to go to the World Wide Web and find the Web site of called "albany.edu."

We wanted to find the home page of a specific department on the University's Web site. An absolute link to the Communication Department's home page is:

Link to the Communication Department home page

This link says that the browser should use hypertext transfer protocol to go onto the World Wide Web and find the University's Web site, then find the Communication Department's site within the University's site, and finally present the page called "index.html."

Visit the Communication Department going to http://www.albany.edu/communication.

Relative links leave out all the "http stuff." They say that instead of going to the World Wide Web the browser should just stay "in the neighborhood" and get a page. A good example is a link to the previous page of this tutorial. This is page 5, and let's say that you want to create a link to page for. It would be:

Previous Page

This relative link tells the browser to go to "page4.html" without going "out" to the Web and coming back into the University's Web site.

You can also link internally to a Web page by using the anchor tag. The anchor tag allows you to jump to a point on a page. We have an anchor called "top" installed on this page. The code for the anchor point is:

The link to it is:

Go to the top of this page

The example on the right shows how this might be coded into a Web page.

Note: An anchor tag always has a "#" at the beginning.

```
<html>
<head>
<title>My Web Page</title>
</head>
<body>
<a name="top"></a>
<p>Welcome to my Web page.
You can go to the top of the
page by <a href="#top">by
clicking here</a></p>
</body>
</html>
```

You can add include an anchor point in an absolute or relative link. The relative link to this page's top anchor point is:

Top of Page 5

[Blank (Empty) Space in HTML (HTML5)](#)
Layout and Tables

The layout of a Web page is important. Placing components, such as text, images, and headings is critical to not only the appearance of a page, but also how well the visitor understands the information presented. A poor layout will look bad, and it will also make the information on the page hard to find and understand.

For information on how to lay out a Web page, visit out our Interactive Media Center Web Design online tutorial at http://library.albany.edu/imc/webdesign/.

Web pages "want" to be displayed with all components aligned vertically on the left side of the page. Center and align right attributes can be applied to many objects. Bet even these alignments do not allow for careful and specific placement of page components.

The illustration on the right shows a Web page with no formatting of any kind.

Using attributes to align object left, center, or right can help establish the position of objects on a Web page. An example is on the right. We will be learning about position attributes and the center tag in detail later in this tutorial.

Tables are the solution to the problems that caused by the default left alignment of standard HTML and XHTML code. Originally intended to present data in "tabular" form, that is columns and rows, tables quickly became the Web page designer's "best friend" by allowing accurate placement object
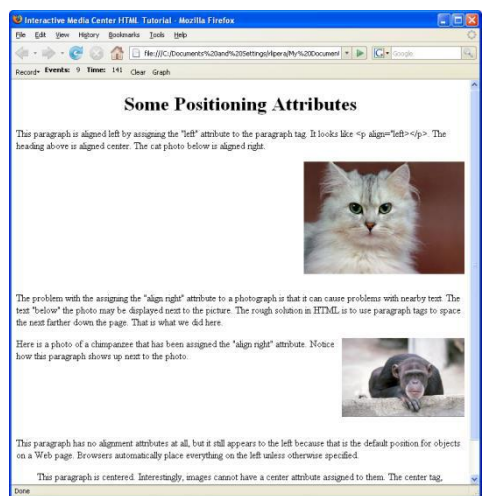
On the right is an example of a Web page that uses a table to control placement of objects. The table is invisible.

On the right is the same page, but this time the table's borders are shown. The table's borders are easily seen and you will be able to observe how it is used to position objects on the page.
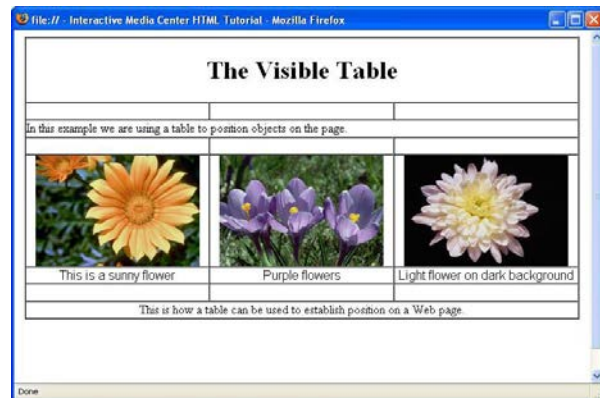
The table tag is a container tag with three main parts. Tables consist of rows and cells, and there are tags for each. Tables must have all the tags written in the proper order for them to work. The tags are:

- The table tag- <table></table>
- The table row tag- <tr></tr>
- The table cell tag- <td></td>

Below is an example of a simple table.

This table has three rows and three columns. To begin building the table we will start with the opening and closing table tags.

    <table></table>

To make it easier to add the rest of the tags we will put the opening and closing table tags on two lines.

    <table>
    </table>

Now we add the first table row tags.

```
<table>
<tr></tr>
</table>
```

The row is not complete until the three table cells are added.

```
<table>
<tr><td></td<td></td><td></td></tr>
</table>
```

Now the table is built like this:

If you are using Notepad, it is a simple "cut and paste job" to install the other two rows.

```
<table>
<tr><td></td<td></td><td></td></tr>
<tr><td></td<td></td><td></td></tr>
<tr><td></td<td></td><td></td></tr>
</table>
```

Now the table has this structure:

Where is the table? The table is really there. But we did not apply any attributes to the table, so although it exists, it is invisible. To make the table appear an the screen, we must define how it will look. The first step will be to add borders. Type in 'border="1"' into the opening table tag as shown below. This creates table borders 1 pixel thick.

```
<table border="1">
<tr><td></td<td></td><td></td></tr>
<tr><td></td<td></td><td></td></tr>
<tr><td></td<td></td><td></td></tr>
</table>
```

The table is now visible. But it looks very small because we have not applied any size attributes.

In addition, there is a curious feature of HTML and XHTML that will cause the table to look odd. Unless there is an object of some kind in a table cell, most browsers will not

display the cell or the border around it. This can cause some peculiar effects. Some browsers will not show the cell, others will. In the table below there is an object in all cells except the center one. Depending on the browser you are using, you may or may not see the center cell correctly.

Text Object		
Text Object		

NOTE: Do you see a center cell, or do you see a "pillow-like" effect? Try viewing this page with another browser. What do you see then?

Objects in cell can be text, images, or even other tables. These are called "nested tables." Objects can also be special pieces of code. The most commonly used one for this purpose is the "nonbreaking space."

The nonbreaking space is a true object that is invisible. It is used to create additional spacing in lines of text and as invisible objects in table cells so that the cells are visible in all browsers. The nonbreaking space is installed by typing a simple ASCII code into the table cell.

Note: ASCII codes are numerical representations of characters that are found on a keyboard and other characters for which there are no keys. In computers and the Web they are used to cause a character to display on the screen when either there is no key or it is not desirable to use the actual keystroke character. Sometimes ASCII codes are used to hide email addresses from Web robots that collect the addresses for spamming. Details and a list of ASCII codes can be found at www.asciitable.com.

The ASCII code for a non breaking space is . By placing a nonbreaking space in the center table cell, it will appear correctly on the Web page as shown below.

Text Object		
Text Object		

As with the previous example, this will depend on your browser. In some instances you may not see any change at all because the cell was always properly displayed. Using

objects such as the nonbreaking space assures that you tables will always appear on the page as you intend.

We can add nonbreaking spaces to the table on which we have been working so that the code looks like this:

```
<table border="1">
<tr><td> </td<td> </td><td> </td></tr>
<tr><td> </td<td> </td><td> </td></tr>
<tr><td> </td<td> </td><td> </td></tr>
</table>
```

To cause the table to be displayed larger we will give it a width attribute of 300 pixels. The example below shows how to do this.

```
<table width="300"border="1">
<tr><td>&nbsp</td<td>&nbsp</td><td>&nbsp</td></tr>
<tr><td>&nbsp</td<td>&nbsp</td><td>&nbsp</td></tr>
<tr><td>&nbsp</td<td>&nbsp</td><td>&nbsp</td></tr>
</table>
```

The "300" refers to the width of the table in pixels. Unless specified otherwise, the default measurements are always in pixels.

Whitespace with Preformatted Text:<pre> tag

Advanced Table Formatting

You are not limited to formatting a table by specifying only its size and borders. There are many options that you can use to make a table appear the way you want it to.

Cell padding and cell spacing are important attributes that can be assigned to a table. Cell padding is the space between an object in a table cell and the sides of the table cell. Below are two examples that demonstrate what this means.

There is a table with no cell padding

There is a table with a cell padding attribute of 10 pixels.

Notice that in the second example there is more space around the text. It is easier read and has a more pleasing appearance. The padding attribute is placed in the opening table tag. The code for the table is:

```
<table width="284" border="1" cellspacing="0" cellpadding="10">
<tr>
<td align="center">There is a table with a cell padding attribute of 10
pixels.</td>
</tr>
</table>
```

Cell spacing refers to the space between cells. Here is an example of a table without cell spacing:

Now here is the same table with 10 pixels of cell spacing.

While this looks like the borders have increased in size, in actuality they have not. To see what thicker borders look like in a table without cell spacing, examine the sample below:

To make the difference between cell spacing and borders as clear as possible, here is the same table with 10 pixels cell spacing and 5 pixels for the table borders.

Using different combinations of cell padding, cell spacing, and borders allows you to control the appearance of your tables so that they can be as useful as possible. Here is a table that combines all three attributes:

Cell Padding 8 pixels	Table borders are 4 pixels
Cell spacing is	The table is 300

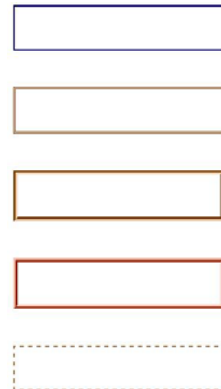|	
10 pixels	pixels wide

The code for this table is:

```
<table width="300" border="4" cellspacing="10" cellpadding="8">
<tr>
<td>Cell Padding 8 pixels</td>
<td>Table borders are 4 pixels</td>
</tr>
<tr>
<td>Cell spacing is 10 pixels</td>
<td>The table is 300 pixels wide</td>
</tr>
</table>
```

Note: The 3-d effect of standard table borders is an old fashioned style. Many Web page builders today want to use table borders, but not the old-fashioned "picture frame"s that we see above. It is possible to create borders in several styles using Cascading Style Sheets. But, the use of CSS is beyond the scope of this tutorial. You may wish to visit our Cascading Style Sheet online tutorial at http://library.albany.edu/csstut/ for more information. Examples of CSS borders are on the right.

Color is added to tables in three different ways:

1. Table background color
2. Cell background color
3. Border color

Below is a table with the background color of #993300. It has standard 1 pixel borders.

The code for the table is:

```
<table width="300" border="1" cellpadding="0" cellspacing="0"
bgcolor="#993300">
<tr>
<td> </td>
<td> </td>
</tr>
<tr>
```

```
<td> </td>
<td> </td>
</tr>
</table>
```

Because the table background color is an attribute that affects the entire table, the color is specified within the opening part of the table tag.

Here is a table in which one cell has the background color of #993366:

The code for that table is:

```
<table width="300" border="1" cellpadding="0"
cellspacing="0"> <tr>
<td bgcolor="#993366"> </td>
<td> </td>
</tr>
<tr>
<td> </td>
<td> </td>
</tr>
</table>
```

Notice that the background color for the table is in the opening part of the table cell tag. We can create a table that has a background color and a cell that has a different background color, such as:

Examine the code for this table closely:

```
table width="300" border="1" cellpadding="0" cellspacing="0"
bgcolor="#993300">
<tr>
<td bgcolor="#993366"> </td>
<td> </td>
</tr>
<tr>
<td> </td>
<td> </td>
</tr>
</table>
```

You will see that the background color is applied to the entire table, but the background color is specified for only one cell. This is a very important concept that can be further illustrated with the table below.

The code for this table is:
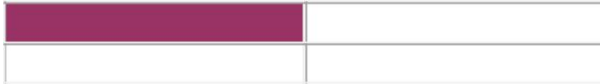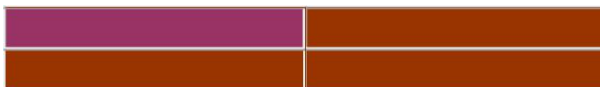
```
<table width="300" border="1" cellpadding="0" cellspacing="0"
bgcolor="#993300">
<tr>
<td bgcolor="#993366"> </td>
<td> </td>
</tr>
<tr>
<td bgcolor="#FFCC33"> </td>
<td bgcolor="#0000FF"> </td>
</tr>
</table>
```

Three of the cells have colors specified, one cell does not. The cell without a color allows the table background color to show through. The background cell colors, on the other hand, have priority over the table background color. Thus, even though the table background color is specified, the cell background colors are displayed! The example below illustrates this a bit better.

Cell Background Color	Table Background Color
Cell Background Color	Cell Background Color

Confused? This is a concept that may not be easy to grasp at first. The file here, http://library.albany.edu/imc/html_tut/table_demo.txt, is a basic Notepad document with the code necessary to create a table. You can open this file, edit it, and then save it with the file name extension **.html**. Try editing the table with different background colors and borders to see what happens.

Background images follow the same priority rules as color. Both tables and table cells can have background images.

Here's a table with a clock photograph for a background:

Notice that the image repeats. This repetition is called "tiling" because the placement of the image is similar to how tiles are laid.

Below is a similar table with two rows and two columns.

Now one of the cells has a toy duck in the background. It also tiles, but only inside the cell. The code for this table is:

```
<table width="267" border="0" cellpadding="0" cellspacing="0"
background="images/clock_60.jpg">
<tr>
<td width="171" background="images/duck_small.jpg"> </td>
<td width="96"> </td>
</tr>
<tr>
<td> </td>
<td> </td>
</tr>
</table>
```

The backgrounds in both the table and the table cell are added as attributes.

The last point about tables that must be touched on here is that there are four alignment attributes for tables. They are default, left, center, and right. The default causes the table to be aligned to the left side of the page unless the visitor's browser causes it the table to be aligned otherwise. Left alignment mandates the table appear on to the left side of the page. Center specifies centering. Right causes the table to be aligned to the right side of the page. These attributes are placed in the opening part of the table tag. The syntax is:

```
<table align="left">
<table align="center">
<table align="right">
```

Default alignment does not require an attribute.

Images

HTML Inserting Images

Images are important parts of Web pages. HTML and XHTML allows you to place images in a Web page and assign attributes to control size and placement. In this next section you will learn about image and image coding.

The image source tag is a single-part tag. It is:

       `<img src="nameofyouimage.gif" />`

Here's an example of simple image placement:

The code for the image is:

       `<img src="images/dogonhat.jpg" alt="" width="205" height="170" />`

The information in the code tells us that the image name is "dogonhat.jpg" and that it is located in the images folder of the Web site. The code also specifies that the size of the image is 205 pixels wide and 170 pixels high.

> NOTE: There are three image file types that can be used on the Web. They are JPG (Joint Photographers Experiment Group), GIF (Graphics Interchange Format), and PNG (Portable Networks Graphics). For more information about these three types of file formats visit Appendix 1 at the end of this document.

Image dimensions are specified as attributes. Specifying dimensions assures that the browser will display the image as you wish. In addition, you can use the dimension attributes to manipulate an image to your advantage. For instance, below is a very small image displayed in its "natural" size, 2 pixels wide by 2 pixels high:

Can you see it? The image is so small that on some monitors it is nearly invisible. However, we can cause it to be displayed much larger by changing its dimensions. Here it is 40 pixels wide by 40 pixels high:

The code for image is:

```
<img src="images/orangedot.gif" alt="Orange dot 40 x 40" width="40" height="40" />
```

Because in both instances the same image is being displayed, there is no delay caused by multiple downloads or by having a larger file size.

Here is the same orange dot displayed as a line:

The code is:

```
<img src="images/orangedot.gif" alt="Orange dot as line" width="300" height="4" border="0" />
```

Note: Changing an image's dimensions in the HTML code does not actually change the image itself. You are only telling the browser how to display the image. Causing a small image to be displayed large does not increase file size or download time. Conversely, give a large image smaller dimensions does not improve download times for file size.

Beware when manipulating photographs, certain types of drawings, and any image that has fine lines, fades, feathering, or other subtle effects. Making a small image to be displayed larger than it's default size requires "spreading" the data (in the form of pixels) of the image over a broader area. The result will be loss of detail and pixillation.

On the right is a photograph that is displayed at its default dimensions, 150 pixels wide by 100 pixels high.

Here is the same photograph with the dimension attributes set to 400 pixels wide by 267 pixels high. Note that the image is highly pixilated.

Images can have the alignment attributes of left and right. The syntax for the alignment is:

```
<img src="yourimage.gif" alt=" align="left" />
<img src="yourimage.gif" alt=" align="right" />
```

Alt tags are special attributes that are applied to images to explain what the image is and what it is about. Individuals who are visually impaired often rely upon a screen-reading Web browser to "read" the text of a Web page to them out loud. An image's alt tag will be read by the browser. This makes in imperative that alt tags were well-formed.

Many search engines also use alt tags as an component in their algorithms that calculate result placement. As with screen reading browsers, search engines rely on properly written alt tags. Below is an image that has an alt tag applied to it.

The alt tag for this image is "Green 1940 Ford Deluxe hot rod automobile" and it the entire image tag is:
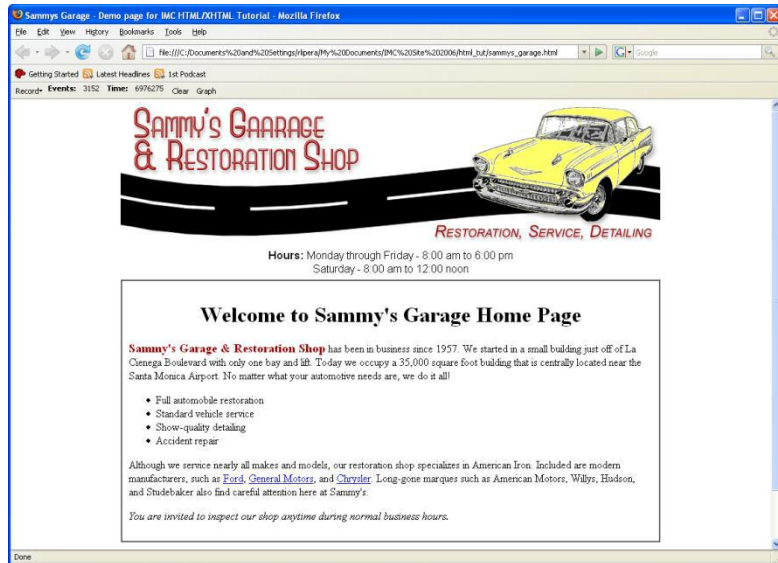
```
<img src="images/green40.jpg" alt="Green 1940 Ford Deluxe hot
rod automobile" width="300" height="261" />
```

Putting It All Together

In this section of the tutorial you will assemble a simple Web page. Among the tasks you will perform will be placing an image, placing and formatting text, creating a table, and assigning attributes to various objects.

The illustration on the right shows what the page should look like when completed.

Thinking back to the beginning of this tutorial, you will remember that you should create a root folder for your project. Do that now.

Start NotePad, if you have not already started it, and create a new file. Name it "index.htm" and save it in the root folder.

Now type in the opening and closing HTML tags. They are shown below.

```
<html>
</html>
```

An important component of the opening HTML tag is the doctype (document type) statement. The doctype statement "tells" the browser important details about the Web page and the code that is used to make it. While it looks complicated, it really isn't. Add a doctype statement just before the opening HTML tag so that the code looks like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
</html>
```

Further information about the doctype statement can be found on the World Wide Web Consortium's Web site at www.w3.org/QA/Tips/Doctype.

One last technical detail will help make sure your Web page is rendered properly by the browsers. All you have to do is add an attribute to the opening HTML tag that refers to "name space." The name space is a complicated concept for novice HTML writer so we will not go into detail here. If you are interested in further information, visit www.w3.org/TR/REC-xml-names. Add the name space attribute to the opening HTML tag so that it looks like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
</html>
```

Now all the preparations are completed and you can start the "meat and potatoes" of Web page building.

Add your opening and closing head tags. The new code should look like:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
</head>
</html>
```

Now add the opening and closing body tags:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
</head>
<body>
</body>
</html>
```

If this begins to look confusing, add some spaces between the opening and closing tags of the head and body. This will help clarify where you will be adding code.

All HTML Tutorials

This and other handouts are available online on our Web site

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

</head>
<body>

</body>
</html>
```
                          HTML Title tag

Save your new Web page. You could view the page in your browser at this point,
however because you have not added any viewable data, the page will appear blank. To
test this, start your browser, click File, and then select Open File or Open, depending on
which browser you use. Navigate to your new Web page, then click OK or Open. A
blank Web page will appear on your screen.

Adding An Image

The next thing that you will do is to add a header image to your page. You must first
download and save the image into your images folder. Go to
http://library.albany.edu/imc/html_tut/download_images/garageheader.jpg to
view the header image named "garageheader.jpg." Right click on the image in your
browser, then click Save Image or Save Picture As (again, depending on your browser)
to save the image in the images folder in your project's root folder.

You will want the header image to appear in the center of your Web page. Add the
center tag as shown below:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

</head>
<body>
<center></center>

</body>
</html>
```

Now add the img src (image source) tag that will place the header image into your page.
Because you have saved that image into your images folder, the code will look like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

</head>
<body>
<center><img src="images/garageheader.jpg" /></center>

</body>
</html>
```

The image is 800 pixels wide and 208 pixels high. Add the dimensions attributes to the img src tag. Don't forget to add an alt tag, too. A suggestion is "Sammy's Garage."The code will now look like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

</head>
<body>
<center><img src="images/garageheader.jpg" alt="Sammy's Garage"
width="800" height="208" /></center>

</body>
</html>
```

Adding A Simple Table

Now you will add a simple table without borders and limited attributes. This table will be used to accurately place the hours of operation for Sammy's Garage on the Web page. You will also apply some text formatting.

Add the table tag, table row tag, and table cell tag to your Web page as shown here:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

</head>
<body>
<center><img src="images/garageheader.jpg" alt="Sammy's
Garage" width="800" height="208" /></center> <table>

<tr>
<td></td>
</tr>
</table>

</body>
</html>
```

Add the text to the table cell that tells about Sammy's hours of operation.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

</head>
<body>
<center><img src="images/garageheader.jpg" alt="Sammy's Garage"
width="800" height="208" /></center>
<table>
<tr>
<td>Hours: Monday through Friday - 8:00 am to 6:00 pm<br />
Saturday - 8:00 am to 12:00 noon </td> </tr>

</table>

</body>
</html>
```

You can save your page now and view it with your browser. You will notice that the text and table are on the left side of the screen and does not have a particularly pleasing appearance. The next step is to add some attributes to the opening table tag so that it is centered. Then set the table to be 800 pixels wide and add 6 pixels of cell padding. The borders should be set to 0 pixels as should the cell spacing. The code should look like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

</head>
<body>
<center><img src="images/garageheader.jpg" alt="Sammy's Garage"
width="800" height="208" /></center>
<table width="800" border="0" align="center" cellpadding="6"
cellspacing="0">
<tr>
<td>Hours: Monday through Friday - 8:00 am to 6:00 pm<br />
Saturday - 8:00 am to 12:00 noon </td> </tr>

</table>

</body>
</html>
```

You can save and view your page to see how adding these attributes have change the appearance of the table and text.

You have not formatted the text yet. That will be your next step. Add the opening and closing font tags "around" the text with the attribute to cause the text to be displayed in Arial. Add the line break tag, <br /> so that the line ends after "6:00 pm" The code will look like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

</head>
<body>
<center><img src="images/garageheader.jpg" alt="Sammy's Garage"
width="800" height="208" /></center>
<table width="800" border="0" align="center" cellpadding="6"
cellspacing="0">
<tr>
<td><font face="Arial>Hours:Monday through Friday - 8:00 am to
6:00 pm<br />
Saturday - 8:00 am to 12:00 noon</font></td>
</tr>
</table>
```

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

Add the strong tag around the word "Hours."

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

</head>
<body>
<center><img src="images/garageheader.jpg" alt="Sammy's Garage"
width="800" height="208" /></center>
<table width="800" border="0" align="center" cellpadding="6"
cellspacing="0">
<tr>
<td><font face="Arial"><strong>Hours:</strong>Monday through Friday -
8:00 am to 6:00 pm<br />
Saturday - 8:00 am to 12:00 noon</font></td>
</tr>
</table>

</body>
</html>
```

If you save and view your page now you will discover that the text is off to the left. Add the align center attribute to the table cell to cause the text to appear in the center. The code so far is:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

</head>
<body>
<center><img src="images/garageheader.jpg" alt="Sammy's Garage"
width="800" height="208" /></center>
<table width="800" border="0" align="center" cellpadding="6"
cellspacing="0">
<tr>
<td align="center"><font face="Arial"><strong>Hours:</strong>Monday
through Friday - 8:00 am to 6:00 pm<br />
Saturday - 8:00 am to 12:00 noon</font></td>
</tr>
</table>

</body>
</html>
```

Main Body Information

In this series of steps you will add the main table for the page that has the important information about Sammy's Garage. The table will have these attributes:

- 800 pixels wide
- 1 pixel border
- 10 pixel cell padding
- 0 pixel cell spacing
- white background color
- centered on the page
- The table cell align attributes should be "left" and "top"

See if you can add the necessary code to make that table appear on your page. If you are not sure what the code should be, check the box on the right.

```
<table width="800" border="1"
align="center"
cellpadding="10"
cellspacing="0"
bgcolor="#FFFFFF">
<tr>
<td align="left"
valign="top"></td>
</tr>
</table>
```

Now you can add the text to the table and format it for appearance, with a list, h1 heading, and links. The illustration below shows how the text should look.

Welcome to Sammy's Garage Home Page

Sammy's Garage & Restoration Shop has been in business since 1957. We started in a small building just off of La Cienega Boulevard with only one bay and lift. Today we occupy a 35,000 square foot building that is centrally located near the Santa Monica Airport. No matter what your automotive needs are, we do it all!

- Full automobile restoration
- Standard vehicle service
- Show-quality detailing
- Accident repair

Although we service nearly all makes and models, our restoration shop specializes in American Iron. Included are modern manufacturers, such as Ford, General Motors, and Chrysler. Long-gone marques such as American Motors, Willys, Hudson, and Studebaker also find careful attention here at Sammy's.

You are invited to inspect our shop anytime during normal business hours.

Go ahead and try it. If you are not satisfied with the results you can check the code in Appendix 2.

Finishing Touches

There is very little left to do to complete the project. A simple paragraph at the bottom of the page with the address of Sammy's Garage will finish it. Be sure to center the paragrah, apply the strong tag to the text, and make the font Arial. The code is:

```
<p align="center"><strong><font face="Arial>Sammy's Garage &amp;
Restoration Shop<br />
1234 Airport Avenue<br />
Los Angeles, CA 90405</font></strong></p>
```

All Web pages should have a title. This title displays on the top bar of the browser and may appear in the bookmarks, depending on the browser. It is also used by search engines. Add a title tag to the head part of the Web page. The code is:

```
<title>Sammys Garage - Demo page for IMC
HTML/XHTML Tutorial</title>
```

To see the code for the entire page visit Appendix 3.

Save and view your page in your browser.

That's it! How does your page look?

If You Continue On...

If you elect to continue further exploration of HTML and XHTML you will discover how to enhance your Web pages even more. Among the things that you can do are:

- Add meta tags to help make search engines list your page
- Control the appearance of text and images
- Format tables for appearance
- Embed media files
- Add attributes to objects

There are numerous resoures on the World Wide Web. For more information, be sure to visit:

- The Interactive Media Center's Introduction to Web Design online tutorial at http://library.albany.edu/imc/webdesign/
- The IMC's Color and the Web online Tutorial at http://library.albany.edu/imc/webcolor
- The IMC HTML/XHTML tag list at http://library.albany.edu/imc/tagsheet.htm
- The World Wide Web Consortium at http://www.w3.org

Good luck with your Web page building!

Appendix 1
Image File Formats

There are three types of image files formats that are used on the Web. They are **J**PG (Joint Photographers Experimental Group), **G**IF (Graphics Interchange Format), and **P**NG (Portable Networks Graphics. (Although the BMP file format is displayed by most browsers, it is not used on the Web for practical reasons.) JPG and GIF are the oldest of the three formats.

The JPG is a compressed format. Data is lost as the file is saved. The degree of compression is set by the image editing software, such as Photoshop or PaintShop Pro. JPG is best suited for photographs because of the way that it handles colors. Many thousands of colors can be displayed, although the actual number of colors will vary by image depending on how much compression has been chosen.

The JPG photograph on the right has been prepared using very little compression. The color is good and there is a high level of detail in the shadows.

The same image is displayed on the right. But, in this example it has been highly compressed. You will notice small "artifacts" that some times appear as dots or ghosting. More compression results is smaller file sizes and quicker downloads. It is necessary to select a compression level that balances quality with file size and download times.

The GIF image format is designed for images that have solid colors, such as drawings and logos. It is limited to no more than 256 colors, although most software allows the user to select less colors for small file sizes and quicker downloads.

The GIF format does not handle photographs well because of the limited number of colors. However, it does handle solid colors extremely well. The image on the right is a GIF. Notice that the there are no artifacts of pixilation in the solid colors.

The photograph on the right has been saved as a GIF. Notice that there is a banding effect in the sky. Because GIF images are limited to no more that 256 colors, many of the colors in a typical sky photo will not be viewable. Those colors will be substituted by the closes available GIF color. The result will often be banding, stripes, or occasionally a "moire" or "watered-silk" effect.

 The PNG format was originally intended to replace the GIF format. However, that did not come to pass. PNG images handle solid colors as well as GIFs do, and they also manage the photographic color well. Notice that in the photograph on the right that the colors are accurate and the sky does not display any of the banding effect that appears in the GIF example above.

Appendix 2

<h1 align="center">Welcome to Sammy's Garage Home Page</h1>
<p>Sammy's Garage & Restoration Shop has been in business since 1957. We started in a small building just off of La Cienega Boulevard with only one bay and lift. Today we occupy a 35,000 square foot building that is centrally located near the Santa Monica Airport. No matter what your automotive needs are, we do it all!</p>

Full automobile restoration
Standard vehicle service
Show-quality detailing
Accident repair

<p>Although we service nearly all makes and models, our restoration shop specializes in American Iron. Included are modern manufacturers, such as Ford, General Motors, and Chrysler. Long-gone marques such as American Motors, Willys, Hudson, and Studebaker also find careful attention here at Sammy's.</p>
<p>You are invited to inspect our shop anytime during normal business hours.</p>

Appendix 3

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Sammys Garage - Demo page for IMC HTML/XHTML Tutorial</title>
</head>
<body>
<center><img src="images/garageheader.jpg" alt="Sammy's Garage"
width="800" height="208" /></center>
<table width="800" border="0" align="center" cellpadding="6"
cellspacing="0"> <tr>
<td align="center"><font face="Arial"><strong>Hours:</strong> Monday through
Friday - 8:00 am to 6:00 pm<br />
Saturday - 8:00 am to 12:00 noon</font></td>
</tr>
</table>
<table width="800" border="1" align="center" cellpadding="10" cellspacing="0"
bgcolor="#FFFFFF">
<tr>
<td align="left" valign="top"><h1 align="center">Welcome to Sammy's Garage Home
Page</h1>
<p><strong><font color="#990000" size="4">Sammy's Garage &amp; Restoration
Shop</font></strong> has been in business since 1957. We started in a small building
just off of La Cienega Boulevard with only one bay and lift. Today we occupy a
35,000 square foot building that is centrally located near the Santa Monica Airport. No
matter what your automotive needs are, we do it all!</p> <ul>

<li>Full automobile restoration</li>
<li>Standard vehicle service</li>
<li>Show-quality detailing</li>
<li>Accident repair</li>
</ul>
<p>Although we service nearly all makes and models, our restoration shop specializes in
American Iron. Included are modern manufacturers, such as <a
href="http://www.ford.com">Ford</a>, <a href="http://www.gm.com">General
Motors</a>, and <a href="http://www.chrysler.com">Chrysler</a>. Long-gone marques
such as American Motors, Willys, Hudson, and Studebaker also find careful attention
here at Sammy's.</p>
<p><em>You are invited to inspect our shop anytime during normal business
hours.</em></p></td>
</tr>
</table>
<p align="center"><strong><font face="Arial, Helvetica, sans-serif">Sammy's
Garage &amp; Restoration Shop<br />
```

```
1234 Airport Avenue<br />
Los Angeles, CA 90405</font></strong></p>
</body>
</html>
```

IT PRESENT

WEB DESIGN & PROGRAMMING

WEB TUTORIALS