Unit –II

Understanding Internet Basics

You can program for the Web, using your skills as a Visual Basic programmer, no matter what your level of experience with Internet technology. If you are new to the Internet or unfamiliar with its technology, Visual Basic allows you to quickly and easily produce functional applications. If you are more experienced with Internet technology, you can work at a more advanced level.

From one perspective, Internet technology simply provides another area for your development efforts. When you deploy Internet applications on the Web, you may go about it differently — incorporating HTML pages with your Visual Basic code, providing security features, and so on — but you're still calling methods, setting properties, and handling events. In this way, all of your knowledge as a Visual Basic developer can be carried into the Internet arena.

From another perspective, applying Internet technology enables you to extend your development skills in exciting new ways. For example, writing Visual Basic code that manipulates HTML pages allows you to decrease deployment costs, reduce client maintenance problems, and reach the broad audience of the Internet.

Internet Clients and Servers

A common way to think about Internet development is in terms of client/server relationships. In this case, the client is the browser, and the server is the Web server. Most interactions on the Internet or an intranet can be thought of in terms of requests and responses. The browser makes a request to the Web server (usually to display a page the user wants to see) and the Web server returns a response (usually an HTML page, an element, or an image) to the browser.

Internet vs. Intranet

The Internet encompasses two categories: the Internet and the intranet. The Internet is a global, distributed network of computers operating on a protocol called TCP/IP. An intranet is also a network of computers operating on the TCP/IP protocol, but it is not global. Generally, intranets are restricted to a particular set of users and are not accessible by the outside world. For example, many corporations use a corporate intranet to provide information to their employees, and run another Internet site for external users. Users within the company can access both the intranet sites and the Internet, but users outside the company can access only the company's Internet sites.

HTML Pages

HTML (HyperText Markup Language) is a language that allows you to display documents in a Web browser. You use HTML to create .htm files that are displayed in a browser. When you create an Internet application in Visual Basic, your user interface is usually made up of HTML pages rather than forms. In many ways, an .htm file (which allows you to display HTML pages) is similar to a Visual Basic .frm file (which allows you to display a Visual Basic form).

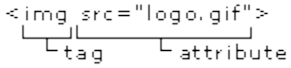
Note While the user interface is generally made up of HTML pages, it can also contain a mix of Visual Basic forms and HTML pages.

An .htm file is a text document that contains a series of tags that tell the browser how to display the file. These HTML tags supply information about the page's structure, appearance, and content. The following figure shows the relationship between page in the browser and its HTML tags:

HTML Page and Source HTML

In addition to describing the structural relationships among page elements, some HTML tags also contain attributes. Attributes provide details about a particular tag. For example, the tag that inserts an image onto a page contains an attribute that specifies the name of the file to insert. The tag is shown below.

HTML Tags and Attributes



Internet Object Models

You use the concepts of object-oriented programming in your Visual Basic Internet applications just as you do in forms-based Visual Basic applications. In Visual Basic Internet applications, you use Internet-related object models to access and manipulate information and controls on your HTML pages.

There are two types of Visual Basic Internet applications: IIS applications and DHTML applications. In IIS applications, you make use of the Active Server Pages (ASP) object model to retrieve information from the user, send information to the browser, and maintain information about the current session. In DHTML applications, you use the Dynamic HTML (DHTML) object model to manipulate the elements on an HTML page.

The important point to remember is that you access the information on your HTML pages through objects, regardless of whether the objects themselves are ASP or DHTML. The object models are explained in much greater detail in the chapters describing each type of application.

For More Information See "A History of Development on the Internet" for more information on the differences between IIS and DHTML applications. See the "Developing DHTML Applications" chapter for more information on using Dynamic HTML objects. See the "Developing IIS Applications with Webclasses" chapter for more information on using ASP objects. See the MSDN™ Web site at http://msdn.microsoft.com/ for details on using HTML and Internet technologies.

Web page

A document displayable in a web browser

Web site

A collection of webpages

Web Server

A computer that hosts a website

Search Engine

A website that helps you find web pages

HTML Basics

Welcome to HTML Basics. This workshop leads you through the basics of Hyper Text Markup Language (HTML). HTML is the building block for web pages. You will learn to use HTML to author an HTML page to display in a web browser.

Objectives:

By the end of this workshop, you will be able to:

- Use a text editor to author an HTML document.
- Be able to use basic tags to denote paragraphs, emphasis or special type.
- Create hyperlinks to other documents.
- Create an email link.
- Add images to your document.
- Use a table for layout.
- Apply colors to your HTML document.

Prerequisites:

You will need a text editor, such as Notepad and an Internet browser, such as Internet Explorer or Netscape.

Q: What is Notepad and where do I get it?
A: Notepad is the default Windows text editor. On most Windows systems, click your Start button and choose Programs then Accessories. It should be a little blue notebook.
Mac Users: SimpleText is the default text editor on the Mac. In OSX use TextEdit and change the following preferences: Select (in the preferences window) Plain text instead of Rich text and then select Ignore rich text commands in HTML files. This is very important because if you don't do this HTML codes probably won't work.

One thing you should avoid using is a word processor (like Microsoft Word) for authoring your HTML documents.

What is an html File?

HTML is a format that tells a computer how to display a web page. The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen.

- HTML stands for Hyper Text Markup Language
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page
- An HTML file must have an htm or html file extension

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

Save the file as **mypage.html**. Start your Internet browser. Select **Open** (or Open Page) in the **File** menu of your browser. A dialog box will appear. Select **Browse** (or Choose File) and locate the html file you just created - **mypage.html** - select it and click **Open**.

HTM or HTML Extension?

When you save an HTML file, you can use either the .htm or the .html extension. The .htm extension comes from the past when some of the commonly used software only allowed three letter extensions. It is perfectly safe to use either .html or .htm, but be consistent. **mypage.htm** and mypage.html are treated as different files by the browser.

How to View HTML Source

A good way to learn HTML is to look at how other people have coded their html pages. To find out, simply click on the View option in your browsers toolbar and select Source or Page Source. This will open a window that shows you the actual HTML of the page. Go ahead and view the source html for this page.

HTML Tags

- HTML tags are used to mark-up HTML elements
- HTML tags are surrounded by the two characters < and >
- The surrounding characters are called angle brackets
- HTML tags normally come in pairs like and
- The first tag in a pair is the start tag, the second tag is the end tag
- The text between the start and end tags is the element content

- HTML tags are not case sensitive, means the same as <B>

Logical vs. Physical Tags

In HTML there are both logical tags and physical tags. Logical tags are designed to describe (tothe browser) the enclosed text's meaning. An example of a logical tag is the tag. By placing text in between these tags you are telling the browser that the text has some greater importance. By default all browsers make the text appear bold when in between the and
tags.
Physical tags on the other hand provide specific instructions on how to display the text they enclose. Examples of physical tags include:

- : Makes the text bold.
- <big>: Makes the text usually one size bigger than what's around it.
- <i>: Makes text italic.

Physical tags were invented to add style to HTML pages because style sheets were not around, though the original intention of HTML was to not have physical tags. Rather than use physical tags to style your HTML pages, you should use style sheets.

HTML Elements

Remember the HTML example from the previous page:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
This is my first homepage. <b>This text is bold</b>
</body>
</html>
```

This is an HTML element:
This text is bold

The HTML element begins with a start tag:
The content of the HTML element is: This text is bold The
HTML element ends with an end tag:

The purpose of the tag is to define an HTML element that should be displayed as bold. This is

also an HTML element:

```
<body>
This is my first homepage. <b>This text is bold</b>

</body>
```

This HTML element starts with the start tag <body>,and ends with the end tag </body>. The purpose of the <body>tag is to define the HTML element that contains the body of the HTML document.

Nested Tags

You may have noticed in the example above, the <body>tag also contains other tags, like the tab. When you enclose an element in with multiple tags, the last tag opened should be the first tag closed. For example:

This is NOT the proper way to close nested tags.

This is the proper way to close nested tags.

Note: It doesn't matter which tag is first, but they must be closed in the proper order.

Why Use Lowercase Tags?

You may notice we've used lowercase tags even though I said that HTML tags are not case sensitive. <B>means the same as .The World Wide Web Consortium (W3C), the group responsible for developing web standards, recommends lowercase tags in their HTML 4 recommendation, and XHTML (the next generation HTML) requires lowercase tags.

Tag Attributes

Tags can have attributes. Attributes can provide additional information about the HTML elements on your page. The <tag>tells the browser to do something, while the attribute tells the browser how to do it. For instance, if we add the bgcolor attribute, we can tell the browser that the background color of your page should be blue, like this: <body bgcolor="blue">
This tag defines an HTML table: <table>. With an added border attribute, you can tell the browser that the table should have no borders: <table border="0">. Attributes always come in name/value pairs like this: name="value". Attributes are always added to the start tag of an HTML element and the value is surrounded by quotes.

Basic HTML Tags

The most important tags in HTML are tags that define headings, paragraphs and line breaks.

Basic HTML Tags

Tag	Description
<html>	Defines an HTML document

<body>	Defines the document's body
<h1> to <h6>	Defines header 1 to header 6
	Defines a paragraph
 	Inserts a single line break
<hr>	Defines a horizontal rule
<!-->	Defines a comment

Headings

Headings are defined with the <h1>to <h6>tags. <h1>defines the largest heading while <h6>defines the smallest.

<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
<h4>This is a heading</h4>
<h5>This is a heading</h5>
<h6> This is a heading</h6>

HTML automatically adds an extra blank line before and after a heading. A useful heading attribute is align.

```
<h5 align="left">I can align headings </h5>
<h5 align="center">This is a centered heading </h5>
<h5 align="right">This is a heading aligned to the right </h5>
```

Paragraphs

Paragraphs are defined with the tag. Think of a paragraph as a block of text. You can use the align attribute with a paragraph tag as well.

```
<p align="left">This is a paragraph</p>
<p align="center">this is another paragraph</p>
```

Important: You must indicate paragraphs with elements. A browser ignores any indentations or blank lines in the source text. Without elements, the document becomes one large paragraph. HTML automatically adds an extra blank line before and after a paragraph.

Line Breaks

The
tag is used when you want to start a new line, but don't want to start a new paragraph. The
tag forces a line break wherever you place it. It is similar to single spacing in a document.

This Code	Would Display
This is a para graph with line breaks	This is a para graph with line breaks

The
tag has no closing tag.

Horizontal Rule

The <hr>element is used for horizontal rules that act as dividers between sections, like this:

The horizontal rule does not have a closing tag. It takes attributes such as align and width. For instance:

This Code	Would Display
<hr width="50%" align="center">	

Comments in HTML

The comment tag is used to insert a comment in the HTML source code. A comment can be placed anywhere in the document and the browser will ignore everything inside the brackets. You can use comments to write notes to yourself, or write a helpful message to someone looking at your source code.

This Code	Would Display
This html comment would <!-- This is a comment --> be displayed like this.	This HTML comment would be displayed like this.

Notice you don't see the text between the tags <!--and -->. If you look at the source code, you would see the comment. To view the source code for this page, in your browser window, select **View** and then select **Source**.

> **Note:** You need an exclamation point after the opening bracket <!-- but not before the closing bracket -->.

HTML automatically adds an extra blank line before and after some elements, like before and after a paragraph, and before and after a heading. If you want to insert blank lines into your document, use the
tag.

Other HTML Tags

As mentioned before, there are logical styles that describe what the text should be and physical styles which actually provide physical formatting. It is recommended to use the logical tags and use style sheets to style the text in those tags.

Logical Tags

Tag	Description
<abbr>	Defines an abbreviation
<acronym>	Defines an acronym
<address>	Defines an address element
<cite>	Defines a *citation*
<code>	Defines computer codetext
<blockquote>	Defines a long quotation
<del>	Defines text
<dfn>	Defines a *definition* term
	Defines *emphasized* text
<ins>	Defines inserted text
<kbd>	Defines keyboard text
<pre>	Defines preformatted text
<q>	Defines a short quotation
<samp>	Defines sample computer code
	Defines **strong** text
<var>	Defines a *variable*

Physical Tags

Tag	Description
<b>	Defines **bold** text
<big>	Defines big text
<i>	Defines *italic* text
<small>	Defines small text
<sup>	Defines superscripted text
<sub>	Defines subscripted text
<tt>	Defines teletype text
<u>	Deprecated. Use styles instead

Character tags like and produce the same physical display as <b>and <i>but are more uniformly supported across different browsers.

HTML Character Entities

Some characters have a special meaning in HTML, like the less than sign (<) that defines the start of an HTML tag. If we want the browser to actually display these characters we must insert character entities in place of the actual characters themselves.

The Most Common Character Entities:

Result	Description	Entity Name	Entity Number
	non-breaking space		
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
"	quotation mark	&quot;	&#34;
'	apostrophe	' (does not work in IE)	&#39;

A character entity has three parts: an ampersand (&), an entity name or an entity number, and finally a

semicolon (;). The & means we are beginning a special character, the **;** means ending a special character and the letters in between are sort of an abbreviation for what it's for. To display a less than sign in an HTML document we must write: **<** or **<** The advantage of using a name instead of a number is that a name is easier to remember. The disadvantage is that not all browsers support the newest entity names, while the support for entity numbers is very good in almost all browsers.

Note: Entities are case sensitive.

Non-breaking Space

The most common character entity in HTML is the non-breaking space ** **. Normally HTML will truncate spaces in your text. If you add 10 spaces in your text, HTML will remove 9 of them. To add spaces to your text, use the   character entity.

This Code	Would Display
This code would appear as this.	This code would appear as this.

This Code	Would Display
This code would appear with three extra spaces.	This code would appear with three extra spaces.

HTML Fonts

The <font>tag in HTML is deprecated. The World Wide Web Consortium (W3C) has removed the <font>tag from its recommendations. In future versions of HTML, style sheets (CSS) will be used to define the layout and display properties of HTML elements. The <font>Tag Should **NOT** be used.

HTML Backgrounds

Backgrounds

The <body>tag has two attributes where you can specify backgrounds. The background can be a color or an image.

Bgcolor

The bgcolor attribute specifies a background-color for an HTML page. The value of this attribute can be a hexadecimal number, an RGB value, or a color name:

```
<body bgcolor="#000000">
<body bgcolor="rgb(0,0,0)">
<body bgcolor="black">
```

The lines above all set the background-color to black.

Background

The background attribute can also specify a background-image for an HTML page. The value of this attribute is the URL of the image you want to use. If the image is smaller than the browser window, the image will repeat itself until it fills the entire browser window.

<body background="clouds.gif">
<bodybackground="http://profdevtrain.austincc.edu/html/graphics/clouds.gif">

The URL can be relative (as in the first line above) or absolute (as in the second line above). If you want to use a background image, you should keep in mind:

- Will the background image increase the loading time too much?
- Will the background image look good with other images on the page?
- Will the background image look good with the text colors on the page?
- Will the background image look good when it is repeated on the page?
- Will the background image take away the focus from the text?

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body background="http://profdevtrain.austincc.edu/html/graphics/clouds.gif" bgcolor="#EDDD9E">
<h1 align="center">My First Webpage</h1>
<p>Welcome to my <strong>first</strong> webpage. I am writing this page using a text editor and plain
old html.</p>
<p>By learning html, I'll be able to create webpages like a<del>beginner</del> pro....<br>
which I am of course.</p>
</body>
</html>
```

Save your page as **mypage3.html** and view it in your browser. To view how the page should look, visit this web page: **http://profdevtrain.austincc.edu/html/mypage3.html**

Notice we gave our page a background color as well as a background image. If for some reason the web page is unable to find the picture, it will display our background color.

HTML Colors

Color Values

Colors are defined using a hexadecimal notation for the combination of red, green, and blue color values (RGB). The lowest value that can be given to one light source is 0 (hex #00). The highest value is 255 (hex #FF). This table shows the result of combining red, green, and blue:

Color	Color HEX	Color RGB
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)

Color Names

A collection of color names is supported by most browsers. To view a table of color names that are supported by most browsers visit this web page: **http://profdevtrain.austincc.edu/html/color_names.htm**

Note: Only 16 **color names** are supported by the W3C HTML 4.0 standard (aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow). For all other colors you should use the **Color HEX** value.

Color	Color HEX	Color Name
	#F0F8FF	AliceBlue
	#FAEBD7	AntiqueWhite
	#7FFFD4	Aquamarine
	#000000	Black
	#0000FF	Blue
	#8A2BE2	BlueViolet
	#A52A2A	Brown

Web Safe Colors

A few years ago, when most computers supported only 256 different colors, a list of 216 Web Safe Colors was suggested as a Web standard. The reason for this was that the Microsoft and Mac operating system used 40 different "reserved" fixed system colors (about 20 each). This 216 cross platform web safe color palette was originally created to ensure that all computers would display all colors correctly when running a 256 color palette. To view the 216 Cross Platform Colors visit this web page:
http://profdevtrain.austincc.edu/html/216.html

16 Million Different Colors

The combination of Red, Green and Blue values from 0 to 255 gives a total of more than 16 million different colors to play with (256 x 256 x 256). Most modern monitors are capable of displaying at least 16,384 different colors. To assist you in using color schemes, check out **http://wellstyled.com/tools/colorscheme2/index-en.html**. This site lets you test different color schemes for page backgrounds, text and links.

HTML Lists

HTML provides a simple way to show unordered lists (bullet lists) or ordered lists (numbered lists).

Unordered Lists

An unordered list is a list of items marked with bullets (typically small black circles). An unordered list starts with the tag. Each list item starts with the tag.

This Code	Would Display
  - Coffee - Milk 	▪ Coffee ▪ Milk

Ordered Lists

An ordered list is also a list of items. The list items are marked with numbers. An ordered list starts with the tag. Each list item starts with the tag.

This Code	Would Display
  - Coffee - Milk 	1. Coffee 2. Milk

Inside a list item you can put paragraphs, line breaks, images, links, other lists, etc.

Definition Lists

Definition lists consist of two parts: a **term** and a **description**. To mark up a definition list, you need three HTML elements; a container <dl>, a definition term <dt>, and a definition description <dd>.

This Code	Would Display
<dl> <dt>Cascading Style Sheets</dt> <dd>Style sheets are used to provide presentational suggestions for documents marked up in HTML. </dd>	Cascading Style Sheets Style sheets are used to provide presentational suggestions for documents marked up in HTML.

Inside a definition-list definition (the <dd>tag) you can put paragraphs, line breaks, images, links, other lists, etc

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body bgcolor="#EDDD9E">
<h1 align="center">My First Webpage</h1>
<p>Welcome to my <strong>first</strong> webpage. I am writing this page using a text editor and plain old html.</p>
<p>By learning html, I'll be able to create web pages like a pro....<br> which I am of course.</p>
```

Here's what I've learned:

How to use HTML tags
How to use HTML colors
How to create Lists

</body>
</html>

HTML Links

HTML uses the <a>anchor tag to create a link to another document or web page.

The Anchor Tag and the Href Attribute

An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc. The syntax of creating an anchor:
Text to be displayed
The <a>tag is used to create an anchor to link from, the href attribute is used to tell the address of the document or page we are linking to, and the words between the open and close of the anchortag will be displayed as a hyperlink.

This Code	Would Display
Visit ACC!	Visit ACC!

The Target Attribute

With the target attribute, you can define **where** the linked document will be opened. By default,the link will open in the current window. The code below will open the document in a new browserwindow:

VisitACC!

Email Links

To create an email link, you will use mailto: plus your email address. Here is a link to ACC's Help Desk:
Email HelpDesk
To add a subject for the email message, you would add ?subject=after the email address. For example:
Email HelpDesk

The Anchor Tag and the Name Attribute

The name attribute is used to create a named anchor. When using named anchors we can create links that

can jump directly to a specific section on a page, instead of letting the user scroll around to find what he/she is looking for. Unlike an anchor that uses href, a named anchor doesn't change the appearance of the text (unless you set styles for that anchor) or indicate in any way that there is anything special about the text. Below is the syntax of a named anchor:

Text to be displayed

To link directly to the top section, add a # sign and the name of the anchor to the end of a URL, like this:

This Code	Would Display
Back to top of page A hyperlink to the top of the page from within the file 10links.html will look like this: Back to top of page 	Back to top of page Back to top of page

Note: Always add a trailing slash to subfolder references. If you link like this: href="http://profdevtrain.austincc.edu/html", you will generate two HTTP requests to the server, because the server will add a slash to the address and create a new request like this: href="http://profdevtrain.austincc.edu/html/"

Named anchors are often used to create "table of contents" at the beginning of a large document. Each chapter within the document is given a named anchor, and links to each of these anchors are put at the top of the document. If a browser cannot find a named anchor that has been specified, it goes to the top of the document. No error occurs.

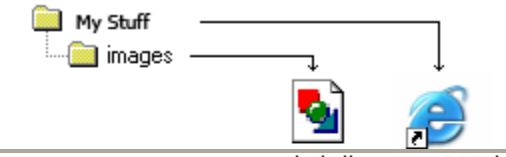
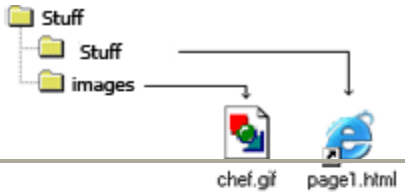
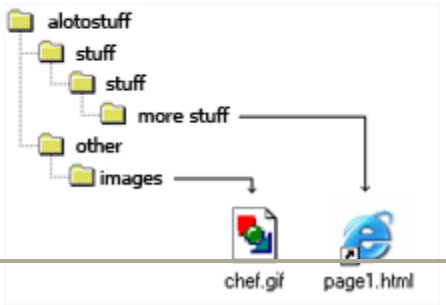
HTML Images

The Image Tag and the Src Attribute

The <img>tag is empty, which means that it contains attributes only and it has no closing tag. To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display on your page. The syntax of defining an image:

This Code	Would Display
<img src="graphics/chef.gif">	

Not only does the source attribute specify what image to use, but where the image is located. The above image, graphics/chef.gif, means that the browser will look for the image name **chef.gif** in a **graphics** folder in the same folder as the html document itself.

	src="chef.gif"means that the image is in the same folder as the html document calling for it.
	src="images/chef.gif"means that the image is one folder down from the html document that called for it. This can go on down as many layers as necessary.
	src="../chef.gif"means that the image is in one folder up from the html document that called for it.
	src="../../chef.gif"means that the image is two folders up from the html document that called for it.
	src="../images/chef.gif"means that the image is one folder up and then another folder down in the images directory.
	src="../../../other/images/chef.gif"means this goes multiple layers up.

The browser puts the image where the image tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

The Alt Attribute

The alt attribute is used to define an alternate text for an image. The value of the alt attribute is author-defined text:

<img src="graphics/chef.gif" alt="Smiling Happy Chef">

The alt attribute tells the reader what he or she is missing on a page if the browser can't load images. The browser will then display the alternate text instead of the image. It is a good practice to include the alt attribute for each image on a page, to improve the display and usefulness of your document for people who have text-only browsers or use screen readers.

Image Dimensions

When you have an image, the browser usually figures out how big the image is all by itself. If you put in the image dimensions in pixels however, the browser simply reserves a space for the image, then loads the rest of the page. Once the entire page is loads it can go back and fill in the images. Without dimensions, when it runs into an image, the browser has to pause loading the page, load the image, then continue loading the page. The chef image would then be:
<img src="graphics/chef.gif" width="130" height="101" alt="Smiling HappyChef">
Open the file **mypage2.html** in your text editor and add code highlighted in bold:

```
<html>
<head>
<title>My First Webpage</title>
</head>
<body>
<h1 align="center">My First Web page</h1>
<p>Welcome to my first webpage. I am writing this page using a text editor and plain old html.</p>
<p>By learning html, I'll be able to create web pages like a pro....<br> which I
am of course.</p>
<!-- Who would have guessed how easy this would be :) -->
<p><img src="graphics/chef.gif" width="130" height="101" alt="Smiling Happy Chef"
align="center"></p>
<p align="center">This is my Chef</p>

</body>
</html>
```

Tables

Tables are defined with the <table>tag. A table is divided into rows (with the <tr>tag), and each row is divided into data cells (with the <td>tag). The letters td stands for table data, which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

This Code	Would Display

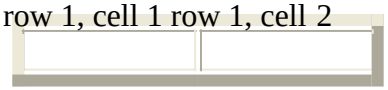
<table> <tr> <td>row 1, cell 1</td> <td>row 1, cell 2</td> </tr> <tr> <td>row 2, cell 1</td> <td>row 2, cell 2</td> </tr>	row 1, cell 1 row 1, cell 2 row 2, cell 1 row 2, cell 2

Tables and the Border Attribute

To display a table with borders, you will use the border attribute.

This Code	Would Display
<table border="1"> <tr> <td>Row 1, cell 1</td> <td>Row 1, cell 2</td> </tr> </table>	 row 1, cell 1 row 1, cell 2

and....

This Code	Would Display
<table border="5"> <tr> <td>Row 1, cell 1</td> <td>Row 1, cell 2</td> </tr>	row 1, cell 1 row 1, cell 2 

Open up your text editor. Type in your <html>, <head>and <body>tags. From here on I will only be writing what goes between the <body>tags. Type in the following:

```
<table border="1">
<tr>
<td>Tables can be used to layout information</td>
<td>  <img src="http://profdevtrain.austincc.edu/html/graphics/chef.gif"> 
</td>
</tr>
</table>
```

Headings in a Table

Headings in a table are defined with the <th>tag.

This code	Would Display
<table border="1"> <tr> <th>Heading</th> <th>Another Heading</th> </tr> <tr> <td>row 1, cell 1</td> <td>row 1, cell 2</td> </tr> <tr> <td>row 2, cell 1</td> <td>row 2, cell 2</td>	Heading / Another Heading row 1, cell 1 / row 1, cell 2 row 2, cell 1 / row 2, cell 2

Cell Padding and Spacing

The <table>tag has two attributes known as cellspacing and cellpadding. Here is a table example without these properties. These properties may be used separately ortogether.

This Code	Would Display
<table border="1"> <tr> <td>some text</td> <td>some text</td> </tr> <tr> <td>some text</td> <td>some text</td> </tr>	some text some text some text some text

Cellspacing is the pixel width between the individual data cells in the table (The thickness of the lines making the table grid). The default is zero. If the border is set at 0, the cellspacing lines will be invisible.

This Code	Would Display
<table border="1" cellspacing="5"> <tr> <td>some text</td> <td>some text</td> </tr><tr> <td>some text</td> <td>some text</td> </tr>	some text some text some text some text

Cellpadding is the pixel space between the cell contents and the cell border. The default for this property is also zero. This feature is not used often, but sometimes comes in handy when you have your borders turned on and you want the contents to be away from the border a bit for easy viewing. Cellpadding is invisible, even with the border property turned on. Cellpadding can be handled in a style sheet.

This Code	Would Display
<table border="1" cellpadding="10"> <tr> <td>some text</td> <td>some text</td> </tr><tr> <td>some text</td> <td>some text</td> </tr>	some text some text some text some text

Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell
<caption>	Defines a table caption
<colgroup>	Defines groups of table columns
<col>	Defines the attribute values for one or more columns in a table

Table Size

Table Width

The width attribute can be used to define the width of your table. It can be defined as a fixed width or a relative width. A fixed table width is one where the width of the table is specified in pixels. For example, this code, <table width="550">, will produce a table that is 550 pixels wide. A relative table width is specified as a percentage of the width of the visitor's viewing window. Hence this code, <table width="80%">, will produce a table that occupies 80 percent of the screen.

This table width is 250 pixels

This table width is 50%

There are arguments in favor of giving your tables a relative width because such table widths yield pages that work regardless of the visitor's screen resolution. For example, a table width of 100% will always span the entire width of the browser window whether the visitor has a 800x600 display or a 1024x768 display (etc). Your visitor never needs to scroll horizontally to read your page, something that is regarded by most people as being very annoying.

HTML Layout - Using Tables

One very common practice with HTML, is to use HTML tables to format the layout of an HTML page.
A part of this page is formatted with two columns. As you can see on this page, there is a left column and a right column.
This text is displayed in the left column.

An HTML <table> is used to divide a part of this Web page into two columns.
The trick is to use a table without borders, and maybe a little extra cell-padding.
No matter how much text you add to this page, it will stay inside its column borders.

```
<html>
<head>
<title>My First Web Page </title>
</head>
<body>
<table width="90%" cellpadding="5" cellspacing="0" >
   <tr bgcolor="#EDDD9E">
      <td width="200" valign="top"><img src="graphics/contact.gif" width="100" height="100"></td>
      <td valign="top"><h1 align="right">Janet Doeson</h1>
      <h3 align="right">Technical Specialist</h3></td>
    </tr>
    <tr>
      <td width="200">
         <h3>Menu</h3>
         <ul>
            <li><a href="home.html">Home</a></li>
            <li> <a href="faq.html">FAQ</a></li>
            <li> <a href="contact.html">Contact</a></li>
            <li> <a href="http://www.austincc.edu">Links</a> </li>
      </ul></td>
      <td valign="top"><h2 align="center">Welcome!</h2>
      <p>Welcome to my first webpage. I created this webpage without the assistance of a webpage editor.
Just my little text editor and a keen understanding of html.</p>
      <p>Look around. Notice I'm able to use paragraphs, lists and headings. You may not be able to tell, but
the layout is done with a table. I'm very clever.</p>
      <blockquote>
         <p>I always wanted to be somebody, but now I realize I should have been more specific.</p>
         <cite>Lily Tomlin </cite> </blockquote>                </td>
    </tr>
</table>
<hr width="90%" align="left">
<address>
    Janet Doeson<br> Technical
    Specialist<br> 512.555.5555
</address>
<p>Contact me at <a href="mailto:jdoeson@acme.com">jdoeson@acme.com</a></p>
</body>
</html>
```

Cascading Style Sheets (CSS)

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, as well as a variety of other effects.
CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.
Advantages of CSS:
n write CSS once and then reuse same sheet in multiple HTML
pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply to all the occurrences of that tag. So less code means faster download times.
- Easy maintenance

To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- Superior styles to HTML

CSS has a much wider array of attributes than HTML so you can give far better look to your HTML page in comparison of HTML attributes.
- Multiple Device Compatibility

Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- Global web standards

Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.