“Writing HTML”

(Self-education series: CompSci-101)

Book 1: “HTML 1 & 2”

Pre-requisites:

None

Level:

Beginner/Power User

Tier:

Public/All (GPL)

0. Standardly Speaking About HTML

Ahhh, there are always **rules** to follow. For HTML, fortunately, the rules are few in number and what they offer is large...

Objectives

This is just an introduction to some concepts behind HTML. After this lesson you will be able to:

- Express the importance of HTML standards
- Describe some of the differences between HTML 2.0, HTML 3.2, and HTML 4.0

Lesson

HTML, or **H**yper**T**ext **M**arkup **L**anguage, is how a web browser displays its multimedia documents. The documents themselves are plain text files (ASCII) with special "tags" or codes that a browser knows how to interpret and display on your screen.

About those standards

No kidding -- the World Wide Web is **exciting.** It is everywhere. It has exploded beyond everybody's expectations (Well, back in 1994 we thought it would be big ;-)

Keep in mind that the thing that makes the Web (and the Internet in general) work are agreed-upon rules ("standards") that allow users of almost any kind of computer to be able to communicate and share information.

Where does HTML fit into this?

What we cover in this tutorial is aimed toward producing documents that comply with current HTML standards.

By using "standard" HTML, your work is going to be most widely "shareable" in the fast changing future of the 'net. The early set of standards, known as HTML 2.0, are supported by nearly all web browsers in use right now.

Things got somewhat more complicated with the features included in HTML 3.2 since Netscape and Microsoft have introduced many features that go beyond standard HTML, and were at first supported by certain web browsers. The web really took off in popularity during the time of the 3.2 standard. By its original design, HTML was **not** designed as a

formatting tool, yet people have found ways (some might say "tricks") to attempt to use HTML for precise web page formatting.

The current set of proposed standards is HTML 4.0 which contain more features for HTML and some attempts to reduce the complexities of different web browsers. This version is starting to move towards a more "logical" method of formatting web pages, via "Style Sheets" which allow the precise formatting web designers wish for, and in a way that separates format from content, making it easy to update an entire web site. However, it will take some time before this functionality is common and there are still bothersome differences between different web browser software (some "standards", yes?) These "standards" turn out to be recommendations as no one has the authority to enforce them! (Note, as of 2006, web standards have made much more progress.)

What does this mean? For accessibility on the widest range of possible web browsers and versions out there, stick with the most basic set of HTML code. Of course, this may limit what you'd like to put in a web page! If you include HTML that may look snazzy only in Netscape but not Internet Explorer, you may turn people away from your site. Not only that, viewers of your web pages may not only be using different browsers, but their monitor size and fonts may not be the same as on the system you designed the pages.

After all, you are probably not going to spend all of this time designing web pages that are for your viewing only! The idea is to make something that the world can view. So the first section of lessons will take you through the most widely accepted features of HTML. From there, you can make the decision to use more of the "deluxe" features.

Review Topics

1. What is HTML?
2. Why should you be concerned about differences in HTML standards?

1. Creating Your First HTML Document

You are about to embark on a journey that will transform you from a mere **Internet Surfer** of the Web to an **Internet Author of Multimedia!**

Objectives

After this lesson you will be able to:

- Identify the meaning and purpose of HTML tags.
- Open up a workspace for creating new HTML documents.
- Use a text editor to create the basic HTML structure for any web page.
- Insert non-displayed comments into your HTML files.
- Open your document within your web browser to see how it is displayed.

Lesson

Now that you know what HTML is, let's start using it.

(Quick quiz -- what do those letters stand for? If you read the previous lesson you would know!).

What are HTML tags?

When a web browser displays a page, it reads from a plain text file, and looks for special codes or "tags" that are marked by the < and > signs. The general format for a HTML tag is:

```
<tag_name>string of text</tag_name>
```

As an example, the title for this section uses a **`header`** tag:

```
<h3>What are HTML tags?</h3>
```

This tag tells a web browser to display the text **What are HTML tags?** in the style of header level 3 (We'll learn more about these tags later). HTML tags may tell a web browser to bold the text, italicize it, make it into a header, or make it be a hypertext link to another web page. It is important to note that the ending tag,

```
</tag_name>
```

contains the "/" slash character. This "/" slash tells a web browser to stop tagging the text. Many HTML tags are paired this way. If you forget the slash, a web browser will

continue the tag for the rest of the text in your document, producing undesirable results (as an experiment you may want to try this later).

NOTE: A web browser does not care if you use upper or lower case. For example, <h3>...</h3> is no different from <H3>...</H3>

Unlike computer programming, if you make a typographical error in HTML you will not get a "bomb" or "crash" the system; your web page will simply look, well... wrong. It is quick and easy to go inside the HTML and make the changes.

Your browser has a small but open vocabulary! An interesting aspect of HTML is that if the browser does not know what to do with a given tag, it will ignore it! For example, in this document you are viewing, the header tag for this section *really* looks like this:

```
<wiggle><h3>What are HTML tags?</h3></wiggle>
```

but since your browser probably does not support a **<wiggle>** tag (I made it up, perhaps in the future it could cause the text to wave across the screen?), it proceeds with what it knows how to do. If I were programming a new web browser, I might decide to add the functionality for the **<wiggle>** tag into my software.

Opening Up Your Workspace

To complete the lessons in this tutorial, you should open a web window plus open your text editor application in a second window.

NOTE: This is a good place to remind you that we will provide directions that are somewhat general as the menu names and file names can differ depending on which web browser you are using. If our instructions say, "Select Open Location... from the File Menu" and your web browser does not have that *exact* choice, try to find the closest equivalent option in your own web browser.

In some web browsers (notable Internet Explorer), a new browser window opens with either a copy of the page you are viewing or your home page. Just ignore that for now, we will load new content in it below.

So you will want to be pretty comfortable jumping between different applications and windows on your computer.

Here are the steps for setting up your "workspace":

1. From the **File** menu of your web browser, select **New Window** or **New Web Browser** (The exact name of the menu command can be different depending on what browser you are using). A second web window should appear. Think of the first window as your "textbook" and the second clone window as your "workspace" for completing the HTML lessons.

2. Next, you need to jump out of the web browser, go to your desktop and open your text editor program.

   NOTE: You will need to move back and forth between the different windows to complete these lessons. This can be a challenge depending on the size of your monitor. You may choose to resize the two windows so that they all fit on your screen or layer your windows so you can click on any of them to bring it to the front.

   If you are using a word processor program to create your HTML, be sure to save in plain text (or ASCII) format.

If you are just starting out, we most STRONGLY recommend that you use the simplest text editor available -- *TextEdit* for the Mac OSX (but you need to know how to save files as Plain Text-- as an alternative, Mac users can download the free and wonderfully simple Plain Old HTML Editor) or the Windows *NotePad*. Why not use those nifty HTML editors? It is sound instructional design that you first learn the concepts and THEN look for shortcuts or helpers that make the work less tedious. When you have got a few lessons under your belt, then check out some HTML editors.

Also, it will help you if you first create a new directory/folder on your computer that will be your work area. You can call it **`workarea`** or **`myspace* (*Please don't sue me)`** or whatever you like; just make sure that you keep all of the files you create in this one area. It will make your life simpler... well, at least while working on this tutorial!

Creating Your HTML Document

An HTML document contains two distinct parts, the head and the body. The **head** contains information about the document that is not displayed on the screen. The **body** then contains everything else that is displayed as part of the web page.

The basic structure then of any HTML page is:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
<!-- header info used to contain extra information about
     this document, not displayed on the page -->
</head>

<body>

<!-- all the HTML for display -->
        :      :
        :      :
        :      :
</body>
```

```
</html>
```

The very first line:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
```

is not technically required, but is a code that tells the browser what version of HTML the current page is written for. For more information, see

http://www.w3.org/pub/WWW/TR/WD-html32#html

Enclose all HTML content within **`<html>...</html>`** tags. Inside is first your **`<head>...</head>`** and then the **`<body>...</body>`** sections.

Also note the **comment** tags enclosed by **`<!-- blah blah blah -->`**. The text between the tags is NOT displayed in the web page but is for information that might be of use to you or anyone else who might look at the HTML code behind the web page. When your web pages get complicated (like you will see when we get into tables, frames, and other fun stuff about 20 lessons from now!), the comments will be very helpful when you need to update a page you may have created long ago.

Here are the steps for creating your first HTML file. Are you ready?

1. If it is not open already, launch your text editor program.
2. Go to the text editor window.
3. Enter the following text (you do not have to press RETURN at the end of each line; the web browser will word wrap all text):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
<title>Volcano Web</title>
</head>
<!-- written for the Writing HTML Tutorial
by Molten Rock, February 31, 2057       -->
<body>
In this lesson you will use the Internet to research
information on volcanoes and then write a report on
your results.
</body>
</html>
```

NOTE: Look where the `<title>...</title>` tag is located. It is in the `<head>...</head>` portion and thus will not be visible on the screen. What does it do? The `<title>` tag is used to uniquely identify each document and is also displayed in the title bar of the browser window.

In lesson 3 you will learn how to add a string of text for a title that will appear directly on your web page.

Also note that we have inserted a comment tag that lists the name of the author and the date the document was created. You could write anything in between the comment tags but it is only visible when you look at the source HTML for a web page.

17. Save the document as a file called **`"volc.html"`** and keep it in the "work area" folder/directory you set up for this tutorial. Also, if you are using a word processor program to create your HTML, be sure to save in plain text (or ASCII) format.

    NOTE: For Windows 3.1 users, you *must* save all of your HTML files with names that end in `.HTM`, so in this case your file should be `VOLC.HTM`. Do not worry! Your web browser is smart enough to know that a file that has a name that ends in `.HTM` is an HTML file.

    You can create files with names like `VOLC.HTML` if you use Windows95 or a later Windows operating system.

    By using this file name extension, a web browser will know to read these text files as HTML and properly display the web page.

Displaying Your Document in a Web Browser

1. Return to the web browser window you are using for your "work space". (If you do not have a second browser window open yet, select **New Window** or **New Browser** from the **File** window.)
2. Select **Open File...** from the **File** menu. (Note: For users of Internet Explorer, click the **Browse** button to select your file)
3. Use the dialog box to find and open the file you created, **`"volc.html"`**
4. You should now see in the title bar of the workspace window the text "Volcano Web" and in the web page below, the one sentence of **`<body>`** text you wrote, "In this lesson..."

Check Your Work

A common mistake we hear is, "I cannot see the title!" You shouldn't! The text within the **`<title>...</title>`** tag is **NOT** displayed on the web page; you should see it in the title bar of the web browser window.

The most common mistake that beginners make here is that they try using a word processing program to type HTML and then are unable to open it in their browser, or if it does, the page is full of odd garbage characters. **When you are starting out, we urge you to use the most basic text editor.** Look for shortcuts later!

Review Topics

1. What are HTML tags?
2. Where is the text of the title tag displayed?
3. What steps are involved in creating a simple HTML document?
4. How do you create a comment tag?
5. How can you display your HTML document in a web browser?

Independent Practice

Think of a topic for your own web page. Now create your own HTML text file that includes a `<title>` tag and a few introductory sentences. Save the HTML file and Reload/Refresh it in your web browser. You might want to create a different folder/directory for this file so you do not get it mixed up with all of the volcano pages you will create for this tutorial.

Keep this file handy as you will add to it in later lessons.

2. Modifying an HTML Document

Now that you have created your first HTML document, you will learn how to swiftly make changes in your document and view the updates within your web browser.

Objectives

After this lesson, you will be able to:

- Re-open the workspace for your web page.
- Make changes in your HTML document using the text editor.
- Reload/Refresh the document in your web browser to see your changes.

Lesson

Re-opening Your Workspace

To complete this lesson, you will need to create a second web browser window and re-open the text editor window you used in the first lesson. Here are the steps for re-opening your workspace (remember that the exact name of the menu commands may be different depending which web browser you are using):

1. If not open, create a new web browser window by selecting **New Window** from the **File** menu.
2. Use the **Open File...** command from the **File** menu to find and open the HTML file you created in the previous lesson.
3. Re-open your text editor program.
4. In the text editor, open the file (**`"volc.html"`**) you created in the previous lesson.

   NOTE: If you are using Windows 3.1 computer then your file should be named `"VOLC.HTM"`. From now on, we will assume that you can easily re-open your workspace in this manner.

Making Changes in Your HTML Document

1. Go to the text editor window.
2. Below the text you typed from the previous lesson, press RETURN a few times and type the following text:

```
3.
4.              A volcano is a location where magma,
5.              or hot melted rock from within a planet,
6.              reaches the surface. It may happen violently,
7.              in a massive supersonic explosion, or more
```

```
8.              quietly, as a sticky, slow lava flow.
```

Note that this text should be **above** the **`</body>`** and **`</html>`** tags at the bottom of your HTML file.

9. Select **Save** from the **File** menu to update the changes in your HTML file.

Reload/Refreshing the Document in your Web Browser

Return to the web browser workspace where the previous version of your file was displayed. Note that the new text you entered in the previous steps may not yet be visible. To see the changes, use the **Reload/Refresh/Refresh** button or menu item in your web browser. This instructs your web browser to read in the same HTML file and display it with whatever changes have been made. You should now see the new text that you entered.

Note that the web browser ignores all blank lines and extra spaces (carriage returns) that you enter in the HTML file. It will also ignore any extra space characters (beyond the one between words). However, when you are writing HTML, it will help you greatly to separate major sections by some blank lines... when you need to go back and edit content, it makes it easier to locate the correct location to make the changes.

Of course, there will be times that you **want** your web pages to have blank space between sections (e.g. between paragraphs). You just passed a location in this very page! In Lesson 4 we will learn how to do this.

Drag and Drop Bonus!

There *may* be an easier way for you to load and view your HTML pages. You will have to arrange your computer desktop so that you can see the icon for your HTML files adjacent to your web browser window. Simply click and drag the icon for your **`"vol.html"`** or **`"vol.htm"`** file right into your web browser window. Voilà! your page will display if your computer supports drag and drop operations (It works for operating for Macintosh OS 7.5 and Windows 95 or newer).

Check Your Work

Compare your document, make sure it matches the text instructions in the **Making Changes in Your HTML Document** section of this lesson.

Review

Review topics for this lesson:

1. How did you re-open your workspace?
2. What steps did you use to make changes in your HTML document?
3. How did you display and view these changes in your web browser?

Independent Practice

As you did in the lesson, modify your own HTML document that you started in the last lesson. Add a few more sentences and see if you can successfully Reload/Refresh the modified document into your web browser.

3. Headings: Six Levels Deep

HTML provides tags for designating headings in six levels of significance. Your browser determines the exact font and size for display.

Objectives

After this lesson, you will be able to:

- Identify the different levels of headings in HTML and the tags associated with them.
- Place different level headings within your HTML document and view the changes within your web browser.

Lesson

HTML Headings

You created headings in HTML by "tagging" certain chunks of text with heading tags. The format for an HTML heading tag is:

```
<hN>Text to Appear in Heading</hN>
```

where **N** is a number from 1 to 6 identifying the heading size. Here are some examples of different heading sizes:

Heading Level 1

Heading Level 2

Heading Level 3

Heading Level 4

Heading Level 5

Heading Level 6

Heading levels range from level 1 (Most Important) to level 6 (Least Important). Like an outline, your heading levels should have logical, consistent order and be parallel.

Placing HTML Headings in Your Document

1. Re-open your workspace (if not already opened).
2. Go to the text editor window.
3. Open the HTML text file you created in lesson 2, **"volc.html"**.
4. First, we will use the tag to display the title as the biggest header, **<H1>**. Enter the following above the existing body text and **after** the **</head><body>** tags:

```
        <h1>Volcano Web</h1>
```

7. Below the text already entered, create other headings for future sections of your *Volcano Web* page.

   Enter the following headings inside the body of your web page (Note that some are **H3** and others are **H2** tags):

```
        <h2>Introduction</h2>

        <h2>Volcano Terminology</h2>

        <h2>Volcanic Places in the USA</h2>

        <h3>Mount St Helens</h3>

        <h3>Long Valley</h3>

        <h2>Volcanic Places on Mars</h2>

        <h2>Research Project</h2>

        <h3>References</h3>
```

24. Save changes in your text editor.
25. Return to your web browser, **Open** and **Reload/Refresh** the HTML file.

Note that on the computer you are using now, you can use the settings in your web browser to define the fonts and/or size of the headings. For example, on one computer you could have a browser display h1 tags as Times font and 36 point; h2 tags as Helvetica font and 24 point, etc. HTML codes designate only that the headers are of a certain type (h1 to h6); how it is displayed is controlled by the user of the web browser.

Check Your Work

If some of your headings do not appear correct, be sure to check that the starting tag and ending tags have the same heading level.

As an optional exercise, take a look at what happens when you make a typographical error. Open your HTML document in the text editor and delete the slash (/) in the **<h1>** tag, after the header **Volcano Web**:

```
          <h1>Volcano Web<h1>
[missing "/" ------------^^^]
```

Save the changes and Reload/Refresh into your web browser. Without the correct ending of the **h1** tag, your web browser interprets all of the succeeding text as part of that header! After trying this you should go back to your document and re-insert the slash in the correct spot.

Review Topics

1. What are the different levels of headings in HTML?
2. What are the tags associated with these different levels?
3. What steps did you use in placing headings in your HTML document?
4. What happens if you forget a slash at the end of a header tag?

Independent Practice

Add at least three headers of different levels to your own HTML document.

4. Breaking it up into paragraphs

So far you have created and modified HTML documents, and you should feel comfortable with the process of editing text and Reload/Refreshing it into your web browser. So now relax for this **fast** lesson on inserting paragraph breaks.

Objectives

After this lesson, you will be able to:

- Identify the paragraph break tag in HTML.
- Copy text from the web page and paste it in another document.
- Insert paragraph breaks into the text of your HTML document and view the changes within your web browser.

Lesson

HTML Paragraph Breaks

We've seen earlier that a web browser will ignore all of the CARRIAGE RETURNS typed into your text editor. But, wherever a browser sees the **`paragraph`** tag, it inserts a blank line and starts a new paragraph. The HTML code for forcing a paragraph break is:

```
<p>
```

Note that this tag is special in that it does not require an ending tag; for now you do not need to use:

```
</p>
```

In a later lesson we will see why we want to use **`<p>`** a closing **`</p>`** for the more current HTML coding standards. For basic HTML coding, let's keep it simple for now.

Also, the **`<h>`** tags have a built in break so it is unnecessary to put **`<p>`** tag before a header tag:

```
<p>
<h2>Blah Blah Blah Blah</h2>
```

Inserting Paragraph Breaks

Follow the directions below to insert and view a paragraph break in your HTML document.

1. Re-open your workspace (if not already opened).
2. Go to the text editor window.
3. Open your working document, **`volc.html`**, in the text editor (if not already opened).
4. First we want to move the sentences **`("A volcano is ....")`** so that they are under the **Introduction** heading. Use the mouse to **cut** and **paste** this text in the proper location.
5. After these sentences, we want to add some more text.

```
6.
7.        Volcanoes have been a part of earth's history
8.        long before humans. Compare the history of human
9.        beings, a few million years in the making, to that of
10.       the Earth, over four billion years in the making.
11.
```

12. Now, return to your HTML document in the text editor, and **type** this text after the existing sentences under the **`<h2>Introduction</h2>`** heading.
13. Save the changes in the text editor.
14. Return to your web browser.
15. If your working document is not visible, Use the **Open Local...** command from the **File** menu to open the document.
16. Select **Reload/Refresh** from the **File** menu. You should now see the two sentences of the Introduction. We now want to put a paragraph break *between* these sentences.
17. Again, return to your HTML document in the text editor.
18. After the second sentence under **`<h2>Introduction</h2>`** (the one that ends " as a sticky, slow lava flow."), press RETURN (it is not necessary but it makes the HTML more readable as you work on it), and then enter the **`paragraph`** tag:

```
19.       <p>
20.
```

This section should now look like:

```
<h2>Introduction</h2>
A volcano is a location where magma,
or hot melted rock from within a planet, reaches the surface.
It may happen violently, in a massive supersonic explosion,
or more quietly, as a sticky, slow lava flow.
<p>
Volcanoes have been a part of earth's history long before
humans. Compare the history of human beings, a few million
years in the making, to that of the Earth, over four
billion years in the making.
```

21. Save the changes in the text editor.
22. Return to your web browser and **Reload/Refresh** the document. The two sentences of the introduction should now be separate paragraphs.

Other types of breaks

To separate major sections of a web page, use the **`horizontal rule`** or **`hr`** tag. This inserts a straight line like you see right above the heading for this section.

The HTML format for a **`horizontal rule`** tag is:

```
<hr>
```

Let's try it now! Put an **`hr`** tag above the **Introduction** heading. This will help to separate the opening sentence of the lesson from the other portions that will follow.

And finally, there is the **`<br>`** tag which forces text to a new line like the **`<p>`** tag, but without inserting a blank line. You might use this tag when making a list of items, when writing the lines of a poem, etc. Compare the differences between using the **`<br>`** and **`<p>`** in these two examples:

Paragraph `<p>` tags Only

HTML	Result

`And then, we could all see`
`at once the brilliant purpose`
`of the paragraph tag.`
`<p>`
`Moving on...`
`<p>`
`the more tags you write,`
`the better you will feel?`
	sample web page
	And then, we could all see at once the brilliant purpose of the paragraph tag.
	
	Moving on...
	
	the more tags you write, the better you will feel?

Paragraph `<p>` and Line Break `<br>` tags

HTML	Result
`And then, we could all see`**` `** `at once the brilliant purpose`**` `** `of the paragraph tag.` **`<p>`** `Moving on...` **` `** `the more tags you write,`**` `** `the better you  will feel?`	**sample web page** And then, we could all see at once the brilliant purpose of the paragraph tag. Moving on... the more tags you write, the better you will feel?

The **
** tag can be used for a different layout style for your section headings. If you notice, the **header** tags **<h1>, <h2>, ...** automatically insert white space above and below the text of the header tag. Some web page authors prefer a style that controls this white space.

Section titles with Header Tags

HTML	Result

```
and in the end it was all
for naught.

<h4>The New Cheese Edict</h4>
Later, sir Longhorn declared
that all makers of cheese would
have to be certified before
commencing production.
```

sample web page

and in the end it was all for naught.

The New Cheese Edict

Later, sir Longhorn declared that all makers of cheese would have to be certified before commencing production.

Section titles with <b> and <br> tags

NOTE! The <b> tag is covered in the next lesson but all it does is make the text bold.

HTML	Result

```
and in the end it was all
for naught.
<p>
<b>The New Cheese Edict</b><br>
Later, sir Longhorn declared
that all makers of cheese would
have to be certified before
commencing production.
```

sample web page

and in the end it was all for naught.

The New Cheese Edict
Later, sir Longhorn declared that all makers of cheese would have to be certified before commencing production.

The difference may seem trivial now, but it opens up possibilities when later we learn to create text of different size and color for our section headings.

Check Your Work

Make sure you entered it as instructed in the **Inserting Paragraph Breaks** section of this lesson.

Review Topics

1. What is the HTML tag for a paragraph break?
2. What steps did you use for inserting a paragraph break in your document?
3. How did you display and view the changes in your web browser?
4. *** Extra Credit:** What is a horizontal rule `<hr>` tag? a `<br>` tag?

Independent Practice

Use the `<p>`, the `<hr>`, or the `<br>` tags to create paragraphs or sections in your own document.

---

5. Doing it with *Style*

Just like a word processor, HTML can tell a web browser to display certain portions of text in *Italic* or **Bold** Style or even a ***combination***.

Objectives

After this lesson, you will be able to:

- Identify the HTML tags for the text styles: **bold**, *italic*, and typewriter (`mono-spaced`).
- Enter text in your HTML document in these different text styles and view the changes within your web browser.

Lesson

HTML Style Tags

HTML offers several tags for adding style to your text. Just remember to be judicious and consistent in the use of styles; too much can make the text uncomfortable to read...

Style tags	
HTML	**Result**

`<b>This is Bold...</b>`
 	
`<i>This is Italic...</i>`
	sample web page
	This is Bold

```
<tt>This is Typewriter...</tt>
```

This is Italic

`This is Typewriter`

Note how you can combine the style tags as long as they are correctly nested, the italic tags are both *within* the bold tags. Note also, that the order does not matter.

HTML	Result

```
<i><b>This is Bold AND
Italic</b></i>

<b><i>And So is This</i></b>
```

sample web page

This is Bold AND Italic

And So is This

Furthermore, you can also add style to the text that appears in heading tags. Note how the different style tags are opened and closed around the words they style and how the heading tags surround the whole text for the heading.

HTML	Result

```
blah blah blah

<h2><i>New</i> and
<tt>Improved!</tt></h2>

blah blah blah
```

sample web page

blah blah blah

***New* and `Improved!`**

blah blah blah

Entering *Styled* Text in Your HTML Document

Follow these steps to apply style tags to your HTML document.

1. Re-open your workspace (if not already opened).
2. Return to your HTML document, **`volc.html`**, in the text editor.
3. Find the word "volcano" in the first sentence of the Introduction. We are going to make this bold to highlight an important word.
4. Insert the tags to make this word appear as bold text:

```
5.            <b>volcano</b>
```

6. Now we will modify the second paragraph with the **bold** and *italic* tags to emphasize a word. Enter **`<b>...</b>`** and **`<i>...</i>`** tags around the word **`billion`** so this section looks like:

```
7.            <p>
8.            Volcanoes have been a part of earth's history long
9.            before humans. Compare the history of human beings,
10.           a few million years in the making, to that of the
   Earth,
11.           over four <b><i>billion</i></b> years in the making.
```

12. Finally, we will use the **typewriter**, tag to indicate a special word. Under the **Volcano Terminology** heading, enter the following:

```
13.             The study of volcanoes, or <tt>Volcanology</tt>,
14.             includes many odd terms.
```

15. **Save** in the text editor and **Reload/Refresh** in your web browser.

Check Your Work

It is important when using style tags to properly terminate the tag(s) with the proper **</>** tag. Otherwise, all succeeding text will inherit this text style. It can look bizarre.

Review Topics

1. What are HTML style tags?
2. What are the different tags used for different styles of text?
3. What steps did you use in entering styled text into your HTML document?
4. ***Extra Credit:** How can these styles be useful in creating a web page or lesson?

Independent Practice

Try using the style tags for **bold**, *italic*, and `typewriter` to the text of your own web page. See if you can successfully combine styles... that are pleasing to read.

6. Lists, Lists, Lists

Lists can present items of information in an easy-to-read format. In fact, there is a list right here, lurking under the next heading!

Objectives

After this lesson, you will be able to:

- Identify HTML codes for creating unordered, ordered, and nested lists for a web page.
- Place different list types within your HTML document and view the changes within your web page.

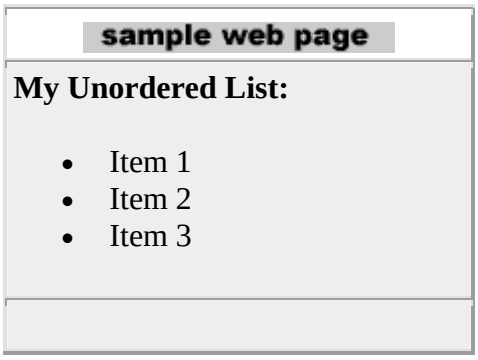
Lesson

Many web pages display lists of items -- these may be items preceded with a "bullet" (Unordered) or a sequentially numbered list (Ordered).

These lists are easy to format in HTML, and they may even be nested (lists of lists) to produce an outline format. Lists are also handy for creating an index or table of contents to a series of documents or chapters.

Unordered Lists

Unordered Lists, or `<ul> .. </ul>` tags, are ones that appear as a list of items with "bullets" or markers in the front. The bullet marks will depend on the particular version of your web browser and the font specified for displaying normal WWW text (e.g. for Macintosh, the bullets are the option-8 character -- in Times font this is a small square, in Geneva it is a large round dot).Here is an example of an unordered list:



And this is the HTML format for producing this format:

```
<b>My Unordered List:</b>
<ul>
  <li> Item 1
  <li> Item 2
  <li> Item 3
</ul>
```

The **<ul>** tag marks the beginning and end of the list, and the **<li>** indicates each **l**ist **i**tem.

Ordered Lists

Ordered lists are ones where the browser numbers each successive list item starting with "1." Note that the only difference is changing the **ul** tag to **ol** tag.

Using the example from above:

And this is the HTML format for producing this format:

```
<b>My Ordered List:</b>
<ol>
  <li> Item 1
  <li> Item 2
  <li> Item 3
</ol>
```

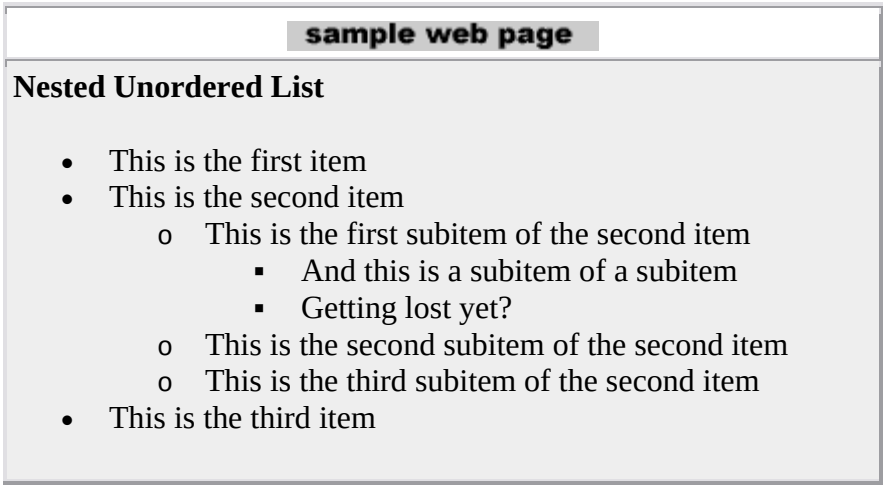
Nested Lists

Ordered Lists and Unordered lists can have different levels; each will be indented appropriately by your web browser. Your major concern will be to verify that each list is properly terminated and the nesting order is correct.

It can start to look complicated with all of those <ol>   tags floating around, but just try to remember the basic structure:

```
<ul>            <ol>
    <li>           <li>
    <li>           <li>
</ul>           </ol>
```

Here is an example of an unordered list with sublevels of other lists:



Note how the bullet marks change for different levels of the list.

And this is the HTML format for producing this format:

```
<b>Nested Unordered List</b>
<ul>
   <li>This is the first item
   <li>This is the second item
   <ul>
     <li> This is the first subitem of the second item
     <ul>
       <li> And this is a subitem of a subitem
       <li> Getting lost yet?
     </ul>
     <li> This is the second subitem of the second item
     <li> This is the third subitem of the second item
     </ul>
   <li>This is the third item
</ul>
```

Nested Lists -- Mixing them together

Not only can you include ordered lists within ordered lists, but you can also mix and match list types. Hold onto your hats! The HTML starts to look pretty ugly, but watch how lists completely contain other lists.

For example, this ordered list includes a nested unordered list:

sample web page

Nested Unordered List Nested Unordered List

1. This is the first item
2. This is the second item
    - This is the first subitem of the second item
        1. An this is a numbered subitem of a subitem
        2. An this is another numbered subitem of a subitem
        3. Getting lost yet?
    - This is the second subitem of the second item
    - This is the third subitem of the second item
3. This is the third item

And this is the HTML format for producing this format. Note how the HTML has been indented to make it easier to read:

```
<b>Nested Unordered List</b>
     <ol>
        <li>This is the first item
        <li>This is the second item
           <ul>
           <li> This is the first subitem of the second item
              <ol>
              <li> And this is a numbered subitem of a subitem
              <li> And this is another numbered subitem of a subitem
              <li> Getting lost yet?
              </ol>
           <li> This is the second subitem of the second item
           <li> This is the third subitem of the second item
           </ul>
        <li>This is the third item
     </ol>
```

Placing Lists in Your HTML Document

Using the **list** tags, you will now add an ordered and an unordered list to your *Volcano Web* page.

1. Re-open your workspace (if not already opened).
2. Open your HTML document in the text editor.
3. Under the **`Volcano Terminology`** header we will use an unordered list to display examples of technical words used in the study of volcanoes. Go to this section in your HTML document.
4. First add the following sentence.

```
        How many of these do you know?
```

5. Now enter the HTML format to create the list of terms:

```
        <ul>
            <li>caldera
            <li>vesicularity
            <li>pahoehoe
            <li>rheology
            <li>lahar
        </ul>
```

13. Now we will use an ordered list to define the required parts of the assignment in this lesson. Under the **`Research Project`** heading, enter the following:

```
    Your mission is to find information and report on a volcano,
    other than the ones listed above, that has erupted in the last
    100 years. Your reports must include:
    <ol>
      <li>Type of volcano
      <li>Geographic location
      <li>Name, distance, and population of nearest major city
      <li>Dates of most recent and most destructive eruptions.
      <li>Other events associated with the recent eruptions
      (earthquakes, floods, mudslides, etc)
    </ol>
    <p>
    Then, write a one page description on the major hazards to
  humans
    in the vicinity of this volcano. Speculate on what you would
  do
    if you were in charge of minimizing the risk to the
  population.
```

29. **Save** your HTML file and **Reload/Refresh** in your web browser.

Check Your Work

Make sure it matches the instructions in the **Placing Lists in Your HTML Document** section of this lesson.

Review Topics

1. How are lists valuable in a web page?
2. What is the HTML tag for a unordered list?
3. What is the tag for a ordered list?
4. How might you set up a nested list?
5. What steps did you use in adding a list to your HTML document?

Independent Practice

You may want to experiment with changing the ordered list you created to one that is unordered. Also, insert an ordered or an unordered list of items to your own WWW document. Be sure to verify that it displays correctly in your web browser. Can you create a list that includes sub-lists?

7. Graphics à la "the Web"

Sending text over the Internet is just old fashioned e-mail. People have been doing it for decades! When you can include **Pictures**, your message can be much more informative! *Is the going rate still 1000 words per picture?*

Objectives

After this lesson, you will be able to:

- Identify the graphic formats for the World Wide Web.
- Discuss key points to consider when including graphics in WWW documents.
- Download a graphic file to your computer.
- Use the correct HTML format for including pictures in your web page.

Lesson

Lean back and relax! This lesson is mostly an introduction to graphics for the Web. But we'll have you do a little activity below.

The Web's Graphic Format

There are numerous file formats for computer graphics... PICT, GIF, TIFF, PNG, not to mention EPS, BMP, PCX, JPEG...

It sounds like cryptic poetry. Bad poetry, Klingon poetry, Geek poetry!

The way a web browser displays graphics in HTML format indicates the location of a graphic file in a single format that can be interpreted by different types of computers. For example, when the information in that format is received by your Macintosh computer, the web browser knows to display it as a picture format for Macintosh. However, when that same information is received by your Windows browser, it is displayed as a Windows graphic.

In technical jargon, we would say that this picture format is **platform independent.** HTML itself is platform independent, since plain text characters can be understood by any computer.

The standard format that can display **within** a web page is GIF or Graphics Interchange Format. The GIF compresses the picture information (reduces the file size) and translates it to binary code that can be sent over the Internet. GIF compression is most effective on graphics that have contiguous areas of solid color, and compression is even greater when the color is continuous in the horizontal direction. GIF images have the feature of defining a color to be "transparent" so images can appear to have non-rectangular boundaries. They can also be saved in the "interlaced" format so that when you see a web page, the images start to appear soon and "dissolve" to the final image.

The other file format used on the web is JPEG (named after the Joint Photographic Expert Group that designed this format). In the early web years, JPEG images were **not** displayed in the page but were displayed in a separate window, using an external "helper" application. But most web browsers these days support JPEG (Or JPG) images to be displayed right in the web page too.

JPEG compression is very effective for photographic images where the colors can vary spatially over short distances ("grainy" images). JPEG offers some dramatic compression in filesize, sometimes by a factor of 10 (e.g. a 1500 kb file reduced to 150 kb), which may be at a trade-off for some image quality. JPEG images do not have the ability to have transparency.

More and more graphics programs have built-in features to save files as GIF format. Newer ones such as ImageReady from Adobe and Fireworks from Macromedia have been specifically designed for creating web graphics.

Some Points to Consider When Using Graphics

For this tutorial, you do not have to use one of these graphics programs. We will show you how to get a copy of the images that you will need.

However, as you begin to develop your own web pages, you should become familiar with creating pictures in either GIF or JPEG format. If your web pages include graphics, consider the following:

- Large and numerous images may look great on a high-end computer, but they will frustrate users who must wait for images to be sent over the network. As a suggestion, keep the total file size of all images on a web less than 100k (we aim for less than 50k each).
- Not all of us have a 21-inch computer monitor! Keep graphic images no wider than 480 pixels and no higher than 300 pixels to avoid forcing users to scroll or resize their web browser window.
- Color gradients may look pretty but for GIF images they do not compress as much as solid color areas and they can sometimes come out "banded".
- Some graphics programs offer options for "no dithering" when converting to GIF -- this can reduce the amount of "noise" in a solid background.

- Many dark grey tones on Macintosh computers are not discernible on Windows computers.
- Rather than displaying all of the images on the web page, have them linked as external images that are downloaded only when a viewer clicks on a hypertext item (this will be covered later in lesson 8a and in lesson 8d). If you have numerous pictures to display, try to break the web page into a series of linked pages.
- A single image (e.g. a small "bullet") can appear several times in a web page with little added delay each time you use that same image.
- Many web browsers "cache" images (storing them on your computer) meaning that using the same file in several web pages will load them from the viewer's computer rather than loading them across the Internet.
- Most importantly, make sure that the images are ones that **add** meaning to your HTML documents.

You may design a beautiful web page, loaded with large pictures, that may load nicely from your computer, but may be excruciatingly slow by a viewer using a slow modem over a busy network. The 'net is a busy place and getting busier every second.

Saving and Including Pictures in Your Web Page

For the next lesson you will first need to download a copy of a GIF image of a volcano (watch out for that hot lava!).

Check Your Work

Check to see that the file, **`lava.gif`**, is saved in the same directory/folder as your HTML file, **`volc.html`**. If it is not there, check to see if you accidentally saved it in another directory/folder. Then, move it to the correct location.

Review Topics

1. What are the two graphic formats used for the World Wide Web?
2. How can a graphic file display on different computers?
3. What are some key points to consider when including graphics in web pages?
4. How did you save the lava graphic for use in your WWW document?

Independent Practice

- Surf the web and browse for pictures. Try to download at least one image that might be useful for your page.

7a. Inline Graphics

WWW Mathematics:

```
Text + Pictures = Multimedia
Multimedia + WWW = Global HyperMedia
```

Got it?

Objectives

After this lesson, you will be able to:

- Place an inline image within your HTML document.
- Select how the pictures align with surrounding text.
- Modify the inline image tag to account for viewers using a non-graphic browser.
- Specify the dimensions of inline image.

Lesson

Let's see how with HTML you can include pictures like the "Big M" in a web page...

HTML Tags for Inline Graphics

An "inline" image is one that appears within the text of a WWW page, such as

 this picture of "Big M".

The HTML format for the inline **`image`** tag is:

```
<img src="filename.gif">
```

where **`filename.gif`** is the name of a GIF file that resides *in the same directory/folder* as your HTML document. By "inline", this means that a web browser will display the image in between text.

Note how the text immediately follows the "Big M" above. What if we want the "Big M" sitting on its very own line? To force the image to appear on a separate line,

simply insert a **paragraph** tag before the **image** tag:

```
<p> <img src="filename.gif">
```

Alignment of Text and Inline Graphics

With an attribute to the **<img...>** tag, you can also control how text adjacent to the image aligns with the picture. The **align** attribute, added inside the **<img>** tag, can produce the following effects:

align=top

```
<img align=top src="filename.gif">
```

sample web page

 **is for Music. Music excites the sole and awakens spirit.....**

align=middle

```
<img align=middle src="filename.gif">
```

sample web page

 **is for Music. Music excites the sole and awakens spirit.....**

align=bottom (default)

```
<img align=bottom src="filename.gif">
```

sample web page

 **is for Music. Music excites the sole and awakens spirit.....**

Note how the text aligns only for one line... (shrink or stretch the width of your WWW browser window to see what happens.) In a later lesson, we will see a way to align blocks of text so that they flow around the side of an image.

Placing an Inline Image in Your HTML document

In this exercise, you will add an introductory picture of a volcano to your lesson.

1. Re-open your workspace (if not already open).
2. Open your **`volc.html`** document with the text editor.
3. Above the **`<h1>Volcano Web</h1>`** heading, enter the following:

```
4.     <img src="lava.gif">
```

   This HTML format will insert, at the very top of your page, the lava picture that you downloaded in the previous lesson.

5. Save and Reload/Refresh in your web browser.

In placing the image, you may have wondered why we did not need to put a **`<p>`** tag after the image. This is because the following text was a header. A web browser **always** inserts a paragraph break before and after an <h1,h2,h3...> tag.

The `alt="..."` attribute

If your web pages will be viewed by users using a text-only browser (such as lynx), they will not be able to view any inline images. Or sometimes, users will turn off the loading of inline images to save time on downloading over slow network connections. An attribute for the **`<img ...>`** tag allows for substitution of a descriptive string of text to hold the place of the image.

Under these conditions, a viewer with a text browser will see a place holder so that the top of our lesson page looks like:

sample web page

```
[IMAGE]
```

```
                          Volcano Web
In this lesson you will use the Internet to research information
on volcanoes and then write a report on your results.
 ---------------------------------------------------------------
In this Lesson...
```

This lets the viewer know that there is a graphic inserted at the top of this page. You could modify the **<img>** tag so that rather than using the place holder, it displays a text string. For example, in our lesson we could add "A Lesson on:" by modifying the **<img>** to read:

```
<img alt="A Lesson on:" src="lava.gif">
```

The **alt="..."** attribute replaces the place holder with a text string so that from a text-only browser (or when loading of images is shut off), it would now appear:

sample web page

```
A Lesson On
                          Volcano Web
In this lesson you will use the Internet to research information
on volcanoes and then write a report on your results.
 ---------------------------------------------------------------
In this Lesson...
```

At this time, make this same edit to your HTML file for the **<img>** tag that displays the picture of the volcano.

Height and Width attributes

Another option you may want to include in your **<img...>** tags are two attributes that give the dimensions of the image in pixels. Why? Normally, your web browser has to determine these dimensions as it reads in the image; the loading of your page can be faster if you specify these numbers in your HTML.

The format for including this option is:

```
<img src="filename.gif" width=X height=Y >
```

where **X** is the width of the image and **Y** is the height of the image in pixels.

You can usually use some sort of graphics program or utility to determine these numbers. Another way to find the dimensions of an image is to load it into your web browser -- you may be able to drag and drop the icon for the image into your browser window -- and the height and width will be displayed in the title bar of the browser.

For our example in this lesson, the **lava.gif** image is 300 pixels wide and 259 pixels high. So you should edit your **volc.html** file to read:

```
<img alt="A Lesson on:" src="lava.gif" width=300 height=259>
```

NOTE: the order of the attributes inside the <img> tag does not matter.

Often I am asked if you can alter the size of the image by inserting numbers other than the actual dimensions of the image. The answers is **yes** but the results may be undesirable. If you insert larger numbers (to make the image bigger) the result will be a "blocky" picture. Sometimes this can be a useful effect on images with large areas of solid color. Take a look at our example of **Going from Small to Big**. If you use lower numbers (to make the image smaller) the result may be a distorted picture. Also, the full size image still has to be downloaded, so there is no real savings in terms of time to download the image. Any re-sizing of the image requires extra "work" by the web browser to recalculate the page layout.

You could experiment and see for yourself. I just might be wrong, and cows fly too. LOL!

You can also specify the size of an inline image in dimensions that are percentages of the current browser window size, so that the image will resize itself if the viewer expands or reduces the size of their browser window.

Check Your Work

Make sure you entered it as instructed in the **Placing an Inline Image in Your HTML Document** section of this lesson.

If you see a picture icon with a question mark:

first check that the HTML file and the image are in the same folder/directory. Then, you may want to make sure that the file name entered in the **<img... >** tag matches the name of the file.

NOTE: Some computer systems (UNIX) are case sensitive for the names of files, meaning that the file "lava.GIF" is NOT the same as "lava.gif". Other computers

(Macintosh) consider them as the same files. Even if your computer does not differentiate between capital and small case, we suggest that you be consistent in the naming of files and how they are referred to in your HTML and use all lower case letters. (Many WWW servers run UNIX. Most actually).

Review Topics

1. What is the HTML format for an inline image?
2. What type of tag must you put before an image tag to make the image appear on a separate line?
3. How did you add the lava picture to your document?
4. What does the `alt="...."` attribute do? What does the `height="...."` attribute do?

Independent Practice

Add an inline image to your web page using a GIF picture file that is stored on your computer or one that you have downloaded from the Internet.

8. Linking it with Anchors

Relax... this lesson is quick and easy! In fact, it is just information for you to read...

What is a URL?

The real power of the web is the ability to create hypertext links to related information. That other information may be other web pages, graphics, sounds, digital movies, animations, software programs, contents of a file server, a log-in session to a remote computer, a software archive, or an "ftp" site.

The World Wide Web uses an addressing scheme known as **URL**s, or **U**niform **R**esource **L**ocators (sometimes also called "Universal Resource Locator"), to indicate the location of such items. These hypertext links, the ones usually underlined in blue, are known as **anchors**.

In the next lessons we will:

- Review the concept of URLs.
- Find and copy URLs from your web browser to your HTML text document.
- Write an HTML anchor to link to another document in the same directory as our first document.
- Write an HTML anchor to link to another document in a different directory as our first document.
- Write an HTML anchor to link to another web document on the Internet.

- Write an HTML anchor that links to another section of the same document.
- Incorporate a graphic that acts as a "hyperlink" to another document.

Wow! That sounds like a lot to do! Don't worry -- it is no more complex than what you have done up to this point.

After all, without the hypertext, we would be only calling this **"Writing TML"** and not **Writing HTML**

8a. Linking to Local Files

Can my document talk to my document? Well, they can at least be linked!

Objectives

After this lesson, you will be able to:

- Create a link to an HTML document in the same directory/folder as your main document.
- Create a link to display a graphic image.
- Create a link to a file in a different directory/folder than your main document.
- Reorganize the structure of your web.

Lesson

Now, you will take your first step of "anchoring" by creating a hypertext link to a second web page. These links are called "local" because they reside on the same computer as the working document (they do not have to venture out on the Internet). You will also be shuffling around the parts of your growing web site (do you see how this becomes more than just a "home page"?).

Link to Local Files

The simplest **anchor** link is one that opens another HTML file in the same directory as the presently displayed web page. The HTML format for doing this is:

```
<a href="filename.html">text that responds to link</a>
```

Think of it as "a" for **a**nchor link and "href" for "**h**ypertext **ref**erence".

The filename must be another HTML file. Whatever text occurs after the first **>** and before the closing **** symbols will be the "hypertext" that appears underlined and "hyper."

Now follow these steps to build an anchor link in your HTML document to a local file:

1. Open your HTML document, **volc.html**, in the text editor.
2. First, under the **Volcanic Places in the USA** heading, enter the following text which introduces the two volcanoes discussed in later sections.

```
3.
4.       Listed below are two places in the United States that are
      considered "active" volcanic areas.
```

5. Below the "Mount St. Helens" heading, enter:

```
6.
7.       On May 18, 1980, after a long period of rest, this quiet
8.       mountain in Washington provided <a href="msh.html">
9.       detailed observations</a> on the mechanics
      of highly explosive eruptions.
```

The text "detailed observations" will link the viewer to a second HTML document called msh.html. This second HTML file does not yet exist; we will construct it in steps (14) thru (24).

10. Save and close your HTML document
11. Now, with your text editor, open a window for a **New** document.
12. Enter the following text in the new window:

```
13.
14.      <html>
15.      <head>
16.      <title>Mount St Helens</title>
17.      </head>
18.      <body>
19.      <h1>Mount St Helens</h1>
20.      The towering pine trees of this once-quiet mountain
21.      were toppled over like toothpicks.
22.      </body>
23.      </html>
```

24. Save this file as **msh.html** in the same directory/folder as your working HTML file (**volc.html**).
25. **Reload/Refresh** **volc.html** in your web browser.
26. Test the hypertext link for the words "detailed observations". When selected, it should connect you to the new page about Mount St. Helens.

Anchor Link to a Graphic

In lesson 7a, we learned how to display an "inline" graphic that would appear in your web page. With the **anchor** tag, you can also create a link to display a graphic file. When the anchor link is selected, it will download the image file and display the image by itself in your web browser.

NOTE: Most web browsers will display such a link to an image file directly in your web browser. Depending on the web browser, and the preferences/settings on your computer, you may be promoted to either save the file or to select an application to display the file. Regardless of the action, if you get that far, the link to the image file has succeeded.

The simplest anchor link is to a file in the same directory/folder as the document that calls it. The format for creating a hypertext link to a graphic is the same as above for linking to another HTML document:

```
<a href="filename.gif">text that responds to link</a>
```

where **filename.gif** is the name of a GIF image file.

Now follow these steps to add a link to a graphic file in your HTML document:

1. Download a copy of a GIF image of Mount St. Helens
2. Open the **msh.html** file in the text editor.
3. Modify the text to include a link to the image of Mount St. Helens.
4.
5.
```
      The towering pine trees of this once-quiet mountain
    were <a href="filename.gif">toppled over like
toothpicks</a>.
```

6. **Save** the **msh.html** file and **Reload/Refresh** in your web browser
7. Now click on the link you just created in step (3).
8. A picture of Mount St. Helens should be displayed.

Links to other directories

The **anchor** tags can also link to an HTML document or graphic file in another directory/folder in relation to the document that contains the anchor. For example, in our lesson, we may wish to keep all of the graphics in a separate directory/folder called **pictures**. As you create more and more HTML files, keeping the image files in its own area will make things a bit more organized for you. Let's do that now:

1. From your computer system, create a sub-directory/folder called **pictures** in the same location where your **volc.html** file is stored.
2. Move the **filename.gif** file to this new sub-directory/folder.

3. Open the **msh.html** file in your text editor.
4. Edit the **anchor** tag for the graphic to read:

```
5.
6.        The towering pine trees of this once-quiet mountain
7.        were <a href="pictures/filename.gif">toppled over
       like toothpicks</a>.
```

NOTE: With HTML you can direct your web browser to open any document/graphic at a directory level *lower* (i.e. a sub-directory or folder within the directory/folder that contains the working HTML file) by using the "/" character to indicate the change to a sub-directory called "pictures."


work area


volc.html


pictures


filename.gif

---

8. Save the HTML document and **Reload/Refresh** in your web browser.

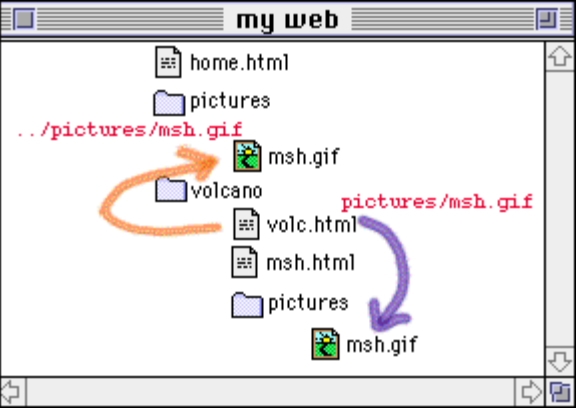
If all went well, the link in the sentence describing the blown-down trees should now call up the graphic file stored in the **pictures** sub-directory/folder.

Anchor Links to a Higher Level Directory

The types of links we have constructed here are known as "relative" links, meaning a web browser can construct the full URL based upon the current location of the HTML page and the link information in the **<a href=...>** tags. This is very powerful because you can build all your web pages on one computer, test them, and move them to another computer -- all the relative links will stay intact.

In this lesson we saw how to construct a hyperlink to a document that is stored in a directory **lower** than the working HTML page. Note that you can also construct a link that connects to a **higher** level directory as well by using this HTML:

```
<a href="../../home.html">return to home</a>
```



Each instance of **`"../"`** the URL of an anchor link tells the web browser to go to a higher level directory/folder relative to the current page; in this case to go up two directory/folder levels and look for a file called **`home.html`**.

In our example, let's say that our **`pictures`** sub directory was not in the same directory/folder as the **`volc.html`** file but was actually one level higher.

In the previous section we constructed a link from the **`volc.html`** file to the **`filename.gif`** file in a subdirectory:

```
<img src="pictures/filename.gif">
```

Now, we want to reorganize our web structure so that the pictures folder/directory is at a higher level. The link is now written:

```
<img src="../pictures/filename.gif">
```

so the web browser looks for a folder called "pictures" that is stored one level up from our **`volc.html`** file.

An advantage of this structure is that it would be easier to store a large number of graphics in this upper folder/directory that can be shared in other web pages. We may do another lesson on landforms that makes use of the pictures stored in this folder/directory.

So now it is time to do a little re-organizing of our HTML files. This requires that you are familiar with moving files and directories around on your computer. **Read this carefully! It may be feeling like it's getting complicated, but it will all be clear soon!**

1. First, create a new folder/directory and name it **`volcano`** (it is recommended to keep the file names in all lower case).
2. Now, move the two HTML files **`volc.html`** and **`msh.html`** into this new folder/directory.
3. Move the **`pictures`** folder/directory (along with the **`filename.gif`** file inside) so that it is in the **same** level as the new **`volcano`** folder/directory. Also, move the **`lava.gif`** file that we added in lesson 7a into the pictures folder.
4. So your entire **`workarea`** directory should now contain two subdirectories -- one that holds your HTML files (**`volcano`**) and another one that holds the graphics (**`pictures`**):

`work area`

`pictures`

 **lava.gif**

 **filename.gif**

 **volcano**

 **volc.html**

 **msh.html**

---

5. We've moved some things around so now we will have to update the anchor links in our HTML files. First, look at the first local link we built in the **volc.html** file:

```
6.
7.      <h3>Mount St Helens</h3>
8.      On May 18, 1980, after a long period of rest, this quiet
9.      mountain in Washington provided <a href="msh.html">detailed
10.     observations</a> on the mechanics of highly explosive
11.     eruptions.
```

   NOTE: Since the msh.html file is still in the same relative directory as volc.html, we do not have to change any of this HTML! Can you see how relative file linking is one of the powerful features of HTML?

12. But now let's look at the link to the picture of Mt. St Helens that we created in the **msh.html** file:

```
13.     The towering pine trees of this once-quiet mountain
14.     were <a href="pictures/filename.gif">toppled over
      like toothpicks</a>.
```

   Open this file in your text editor and edit the link to read:

```
The towering pine trees of this once-quiet mountain
were <a href="../pictures/filename.gif">toppled over
like toothpicks</a>.
```

   This relative link tells the web browser to go up one level from the current folder/directory (**volcano**) and look there for another folder/directory called **pictures** that contains a GIF image called **filename.gif**

15. You will have to update the **<img...>** tag that displays the title graphic. Open the **volc.html** file in your text editor and modify the line just below the **<body>** tag to read:

```
  <img alt="A Lesson on:" src="../pictures/lava.gif" width=300
height=259>
```

16. Save your file. You should then **Open** the **`volc.html`** file in your web browser and test the link to **`msh.html`** and then try the link to the picture of Mount St Helens.

One More Small Change

This last small step may not be obvious, but we will explain it shortly. The last thing you should do in this lesson is to change the name of your working file from **`volc.html`** to **`index.html`**. You should do this using the normal way of editing a file's name from the computer desktop (on the Macintosh click on the file name; on Windows right-mouse click on the icon and select the option for **Rename**). Note also for Windows users that if you use a special editor program to create HTML files, you will not see the ".html" extension on the desktop file name, so in that case, you would change the file name from **`volc`** to **`index`** because under the hood, the computer knows that there is a ".html" at the end.

Why are we doing this? Let's say you have finished this lesson and are ready to store it on a World Wide Web server for the world to see. And let's assume that the Internet address for this server at CULM University is:

```
http://www.CULMu.edu/
```

And your file will be stored in a series of directories:

```
--= top level of server: www.CULMu.edu
   /courses
        /science
            /geology
                /volc.html
```

so that the URL for the Volcano Web might be:

```
http://www.CULMu.edu/courses/science/geology/volcano/volc.html
```

Pretty long, eh? Now here is the promised explanation -- on most WWW servers you can designate one standard name that is the "default" web page for that directory and on most systems that name is.... **`index.html`**. What this means is that the Internet address:

```
http://www.CULMu.edu/courses/science/geology/volcano/
```

is equivalent to

```
http://www.CULMu.edu/courses/science/geology/volcano/index.html
```

This might make you think that it is a lot of energy to cut 20 letters out of a URL! But it does tend to make your URL look a bit more professional -- If you were creating the Longhorn Cheese Home page,

```
http://www.cheese.com/longhorn/
```

looks less redundant in print than

```
http://www.cheese.com/longhorn/longhorn.html
```

which comes into play when people read about your URL and are trying to connect by typing it into their web browser.

Note also that this special file name **`index.html`** is used on most web servers but it might also be **`default.htm`**-- check with the people that run your web server.

Check Your Work

You will first see your *Volcano Web* page. When you click on the hypertext for **`detailed observations`**, your web browser will display a new page. Finally, when you click on **`toppled over like toothpicks`**, your web browser will display in an external window a picture file that is stored in a sub folder/directory.

Use the web browser's **back** button twice to return to this page. If your web page was different from the sample, review the text you entered in the text editor.

Review Topics

1. What were the steps you used in creating a link within your document to a local file?
2. What steps did you use to create a link which displayed a graphic in an external window?
3. How did you create a link to a file in a lower directory/folder than your main document? a higher directory?
4. What is the significance of a file called **`index.html`** on a WWW server?

More Information

You can use the anchor link tag to build a hypertext link that can download **any** kind of file, not just HTML or image files. Often, this is used to provide links to download software files or document files.

The method to do this, is exactly the same as you have seen for linking to HTML files- use the `<a href="...">link text</a>` tag structure, except the part in the quotes is the name of the file:

- `<a href="myfile.zip">Download a Windows .ZIP file</a>` (compressed archive of multiple files)
- `<a href="myfile.exe">Download a Windows .EXE file</a>` (an executable file, e.g. software)
- `<a href="myfile.sit.hqx">Download a Macintosh .HQX file</a>` (compressed archive of multiple files)
- `<a href="myfile.pdf">Link to a Adobe Acrobat (PDF)</a>` (document file, either viewed in the browser or downloaded)
- `<a href="myfile.doc">Link to a MS Word file</a>` (document file, either viewed in the browser or downloaded)
- `<a href="myfile.ppt">Link to a MS PowerPoint file</a>` (presentation file, either viewed in the browser or downloaded)

Independent Practice

Create a second HTML document that uses the HTML formatting that you are familiar with at this point. Return to the first one you created and make an anchor that links to this new one.

8b. URLs: Pointers to the Internet

`URL? Earl? Yurl? hurl? gyrl?`
It's getting time to link to that Big Wide Web using the web's addressing scheme.

Objectives

After this lesson you will be able to:

- Identify the function of Uniform Resource Locators (URLs).
- Recognize the structure of a URL.
- Examine the URLs in the hypertext links of any web page.

Lesson

Note: For this lesson, you will not need your HTML text file. This is another low-effort lesson!

What is a URL?

The **Uniform Resource Locator (URL)** is what the WWW uses to find the location of files and documents from computers on the Internet. On your web browser screen, the URL for this document is typically displayed in the upper part of the Web browser window. The URL includes:

- an identifier for the type of Internet server;
- an Internet address; and
- a file path to the particular item of interest.

The URL is what you will need to build a link from the web page that you are creating to connect to some other piece of information available on the Internet.

How are URLs Structured?

The structure of a URL is:

```
type://in.ter.net.address/directory/sub-directory/.../filename
```

The **"type"** indicates the type of Internet server being accessed:

http
: a web server, "HttP" stands for HyperText Transfer Protocol

gopher
: an Internet Gopher site or menu driven directories of files and information

ftp
: a File Transfer Protocol (FTP) site, archives of files.

telnet
: initiates a Telnet session to log on remotely to another computer When selected, your web browser will launch a Telnet external program and connect to the specified site.

WAIS
: Wide Area Indexed Server -- a site to search a collection of subject oriented documents by keywords

file
: A file on your local computer system (hard drive, floppy, local file server)

The **type** is always followed by **"://"** and the Internet address of a remote computer. This is in the structure of:

```
host.domain.domain.domain
```

For example:

```
machine.department.church.org
123.45.6.78
office.company.com
agency.branch.gov
machine.organization.country
```

If the URL is to the main level of this host (its "home page"), then the URL is terminated with a slash "/". If you are linking to a sub-directory or a file, you must also add the exact path to that item using the slash character to indicate the entire file path.

Note: For most web servers spelling does count! So does capitalization! File names on UNIX computers are case sensitive, meaning that a file named

```
SpiffyText.html
```

is a different file than

```
spiffytext.html
```

Experimenting With URLs

Note that URLs can link to any site, directory, subdirectory, text file, image, digital movie, or sound file on any Internet site that is set up for access. The best way to see different URLs is just to move your mouse over any hypertext link in any web page -- if you look at the bottom of your web browser, it should display the URL that you would connect to if you clicked on that link. You could go to a big site such as Yahoo and "peek" at URLs.

Here is an easy way to copy a URL for a link in any page. You first must access the "secret" pop-up menu from any hypertext link in a web page -- click and hold the mouse for Macintosh; click the right mouse button for Windows and Unix. From this menu, select **Copy This Link Location** (or similar menu item). After releasing the mouse button, jump to any text document and select **Paste** from the **Edit** menu. Voilà! You've just nabbed a URL from a link in the web page (this way, you can copy a URL without even visiting the page it links to!)

Review Topics

1. What purpose do URLs serve for the World Wide Web?
2. Where are URLs found on a WWW screen?
3. What is the basic structure of an URL?
4. How can you see the URLs that a hypertext link will jump to?

Independent Practice

Find some sites on the Internet that intrigue you. For each one, record its name and its URL displayed near the top of your browser window. See if you can copy and paste the

URLs into a text document. You will use this list later to add links from your own web pages to these sites that you found.

8c. Links to the World: Internet Sites

You've been revving your engines, and itching to hit the Information SuperHighway, right? Here, we will extend our use of **`anchor`** tags to create links to resources out there on the Internet.

Objectives

After this lesson you will be able to:

- Create an anchor to link to an Internet site.
- Quickly copy the URL for a site and use it in your HTML document.

Lesson

HTML for Anchors to the Internet

Linking to a site on the Internet combines what we have worked on earlier in the lessons on **Links to Local Files** (Lesson 8a) by incorporating what we have learned about **URLs** (Lesson 8b). The full HTML format for an **`anchor`** link to an item on the Internet is:

```
<a href="URL">Text to Activate Link</a>
```

where **`URL`** is the full Uniform Resource Locator, in quotes, the address for the Internet site where you want to send the viewer. The string **`Text to Activate Link`** is what will show up as hypertext in your web browser (usually but not always) underlined and in blue. When a viewer clicks on this hypertext, the web browser will link them to the Internet site indicated by the URL. Remember that a URL can be a link to another World Wide Web (WWW) server, Gopher server, FTP site, or any text, graphic, sound, video file on these servers.

Now, we will add a hypertext link to a site that has information about volcanoes on the planet Mars. Follow these directions to add anchor links on your HTML document:

1. Open your **`index.html`** file in the text editor.
2. Below the heading, **Volcanic Places on Mars**, enter the following text:

```
3.
4.   <a href="http://solarviews.com/eng/mars.htm">
5.   Mars</a> has its fair share of volcanic landforms,
```

```
6.   including the largest known volcano in the solar system,
7.   <a href="http://www.solarviews.com/browse/mgs/mgsmons1.jpg">
8.   Olympus Mons</a>
```

Note: We've made a link to two different types of information. The first hyperlink connects to a web page that describes information about the planet Mars. The second is a link to a large JPEG image of a Martian volcano.

9. **Save** and **Reload/Refresh** in your web browser.

Note: We have shown you how to link directly to an image from another web server. You could quite easily use an off-site URL in your `IMG` tags for your own web pages. We strongly urge you to contact the site's creator and ask permission.

In some cases, web site owners are penalized or charged for excessive accesses. For more information on this issue, see a Plea from the Widows Web (http://www.widowsweb.com/widows/plea.html) (Note, as of April 2002, this web site has vanished, but you can find it's remnants (http://web.archive.org/web/*/http:/www.widowsweb.com/widows/plea.html) from the Internet Archive's WayBack Machine).

A Quick 'n Easy Way to Enter URLs in Anchor Tags

As you navigate among different web pages, the URL of the currently visible page can be viewed at the top of the web browser window (You may have to look for a menu option to **Display URLs**). For example, in this document, the URL looks something like:

```
http://www.Ilovemusicchruch.org/tshirts.htm
```

You can use your mouse to select and **copy** a URL from the web browser display and then paste it in the **`anchor`** tag of your HTML document. This is much more efficient than writing URLs down on paper (some are quite long!).

Now we will add some links to other sites that we will list under the **References** section of our Volcano lesson. One such site that might offer relevant information is the US Geological Survey.

Follow these steps:

1. Open your HTML document **`index.html`** in the text editor
2. Under the heading "References", enter:

```
3.
4.
5.   Use these references to start your research:
6.
```

```
7.   <ul>
8.   <li> <a href="">Educational Resources from the
9.   USGS</a>
10.  </ul>
```

NOTE: We've constructed the hypertext link but we still need to enter a URL between the quotes.

11. Connect to the US Geological Survey Education Index. (http://info.er.usgs.gov/education/index.html)
12. From the web page, use the mouse to **Select** the URL as displayed in the URL display field.
13. From the **Edit** menu, **Copy** the URL.
14. Now, return to your HTML document **`index.html`**
15. Click the mouse once between the two quote marks you inserted in step #2 and **Paste** the text you copied in step #5. The final **`anchor`** tag should look like:

```
16.
17.  <a href="http://info.er.usgs.gov/education/index.html">
18.  Educational Resources from the USGS</a>
```

Note: You have just set up the HTML structure for an Unordered List, with each list item a hypertext link to a site that offers information about volcanoes. For a review of lists, see lesson 6

Check Your Work

If your links do not properly connect to a remote site, review the text you entered in the text editor. Note that your list of references may be different from the example.

Review Topics

1. What is the address for an item on the Internet?
2. What steps did you take for creating an anchor link to the NASA Internet Site?
3. What shortcut was identified for putting an URL into an anchor link?
4. Tell a colleague or friend about any other links that you put in your document.

Independent Practice

Create anchor links from your own web page that connect to some of the URLs addresses you have discovered

8d. Links to Sections of a Page

You have seen how to link to other web pages, whether they are ones you created or have found elsewhere on the Internet. What if you wanted to connect to a specific section **within** a document? YOU CAN!

Objectives

After this lesson you will be able to:

- Create a hidden reference marker for a particular section within an HTML file.
- Build a hypertext link that will connect to a particular section within an HTML file.
- Build a hypertext link that will connect to a particular section within another HTML file.
- Create a hypertext table of contents for a web page.

Lesson

The Named Anchor

A **named anchor** is a hidden reference marker for a particular section of your HTML file. This might be used to link to a different section of the same page if it is long, or to a marked section of another page. For example, on this page you are viewing, I could create a hidden marker called "check" that marked the heading below "Check Your Work". Then, I write an anchor link that connects to this section of this document. Once you click on a link to this named anchor, your web browser will jump so this line is at the top of the screen.

The HTML format for creating the named anchor is:

```
<a name="NAME">Text to Link To</a>
```

or for the link you just tried above:

```
<a name="check">Check Your Work</a>
```

Notice how this subtly differs from the anchor link **`<a href=...`** that we learned about in lesson 8a.

Writing a Link to a Named Anchor

When you want to create a hypertext link (or an "anchor link", see lesson 8a) to a named anchor, use the following HTML:

```
<a href="#xxxxx">Text to act as hypertext</a>
```

or for the link you just tried that sent you to the section below:

```
Go to  <a href="#check">Check Your Work</a>
```

The "#" symbol instructs your web browser to look through the HTML document for a named anchor called **`"xxxxxx"`** or in our example **`"check"`**. When you select or click on the hypertext, it brings the part of the document that contains the named anchor to the top of the screen.

Adding Named Anchors to the *Volcano Web* Lesson

Now, we will build a table of contents for our lesson that will appear at the top of our *Volcano Web* page. The entries for this will be the headings we have already created. Each will act as a hypertext link to a particular section of our lesson.

The first step is to create a named anchor for each of the heading sections in your *Volcano Web* lesson:

1. Open your **`index.html`** file in the text editor
2. Find the heading for the **Introduction**. Change it from:

   ```
   <h2>Introduction</h2>
   ```

   so that it looks like:

   ```
   <h2><a name="intro">Introduction</a></h2>
   ```

   NOTE: You have just marked the line that contains the header "Introduction" with a hidden reference marker, the named anchor "intro".

3. In a similar manner, change all header **`<h2>`** tags for the other sections:

```
    <h2><A NAME="term">Volcano Terminology</a></h2>

    <h2><A NAME="usa">Volcanic Places in the USA</a></h2>

    <h2><A NAME="mars">Volcanic Places on Mars</a></h2>
```

```
10.
11.    <h2><A NAME="project">Research Project</a></h2>
```

12. If you **Reload/Refresh** now from your web browser, you will not notice any visible change. The named anchor tags we have just added are hidden from the user's view.

Adding Links to a Named Anchor in the Same Document

Now we will set up a table of contents that will appear at the top of the screen. We will use an unordered list (see lesson 6 for more on lists) to create this list:

1. If not already open, open your **`index.html`** file in the text editor.
2. Below the first sentence under the **Volcano Web** heading enter the following text:

```
3.
4.   <hr>
5.   <b>In this Lesson...</b>
6.   <ul><i>
7.   <li>Introduction
8.   <li>Volcano Terminology
9.   <li>Volcanic Places in the USA
10.  <li>Volcanic Places on Mars
11.  <li>Research Project</i>
12.  </ul>
```

NOTE: This index is marked off above and below by a solid line using the <hr> tag. The *Italic* style is used on the entries for the text. At this point we have only entered the text of the index entries. Below we will add the HTML to make the links active.

13. **Save** and **Reload/Refresh** into in your web browser.

Finally, we want to make each item of the list act as a hypertext link to another section of the document. To do this, we will use a variation of the basic anchor link lessons 8a. Rather then linking to another file, we link to one of the named anchors (the hidden markers that you created above) within the same HTML file. We indicate a named anchor by preceding the reference marker name with a "#" symbol:

1. Open your **`index.html`** file in the text editor
2. Go to the first list item in your index section. Change it from:

```
3.
4.       <li>Introduction
```

to look like:

```
<li><a href="#intro">Introduction</a>
```

5. You should now be able to fill in the other links to named anchors so that the section looks like:

```
<hr>
<b>In this Lesson...</b>
<ul><i>
<li><a href="#intro">Introduction</a>
<li><a href="#term">Volcano Terminology</a>
<li><a href="#usa">Volcanic Places in the USA</a>
<li><a href="#mars">Volcanic Places on Mars</a>
<li><a href="#project">Research Project</a></i>
</ul>
```

16. **Save** and **Reload/Refresh** into your web browser. When you click on an item in your index, the browser should display the associated section at the top of your screen.

NOTE:Your web browser will try its best to bring to the top of the browser the text enclosed by the `<a name="xxx">...</a>` occurs. In this case, there is no content after the last few examples, so when you click on the link to `Research Project` it will *not* jump to the very top of the screen. If you really wish to make this happen, until we write more content under this heading in youre document, you can insert 10-20 of `<br>` tags.

Adding Links to a Named Anchor in Another Document

You can create a link that jumps to a section of another HTML document that is marked by a named anchor. The HTML for building a link to a named anchor in another local HTML document is:

```
<a href="file.html#NAME">Text to activate link</a>
```

and if the document exists on another web site:

```
<a href="http://www.cheese.org/pub/recipe.html#colby">Colby Cheese</a>
```

In lesson 8a we created a hypertext link that jumped from the section of our lesson on Mount St. Helens to **`msh.html`**, a separate HTML file. Now we will add a link in that second document that will return to the original section of the main Volcano page.

1. Open your **`msh.html`** file in the text editor
2. Near the bottom of the HTML (but above the **`</body>`** tag) enter the following text:

```
<hr>
<a href="index.html#usa">Return to <i>Volcano Web</i></a>
```

NOTE: We have included the *Italic* Style tag ... within the text marked by the named anchor "usa".

6. **Save** and **Reload/Refresh** into your web browser. When you click on one of the hypertext links at the bottom of the **`msh.html`** page, you should jump back to the "Volcanic Places in the USA" section of the *Volcano Web* lesson.

In this case the link is just the name of another HTML file, **`msh.html`**, in the same directory as the **`index.html`** file. However, you can use a full URL (see lesson 7) to link to a named anchor in an HTML file on a remote computer. For example, to create a link to the "Introduction" section of an HTML file stored on the CULM WWW server, the syntax would be:

```
<a href="http://www.Ilovemusicchurch.org/final/index.html#intro">
   Introduction to Volcano Web</a> ( THIS EXAMPLE DOES NOT EXIST )
```

The reference marker **`"#anchor_name"`** is tacked onto the end of the URL.

Check Your Work

If the named anchor links do not properly connect, review the text you entered in the text editor.

Review Topics

1. How do you identify a named anchor?
2. What are the steps for creating a link to a different section within a document?
3. How do you modify an anchor link to connect to a named anchor in another HTML document?
4. How do you create a table of contents for a web page?
5. What is the difference in function between the tags **`<a href="...>`** and **`<a name="...>`** ?

Independent Practice

Create named anchors for the headings in your own web page and build an index that will link to these sub-sections.

8e. HyperGraphics

Text does not have a monopoly on being "hyper"... expand the versatility of your web pages by having **pictures** act as hyperlinks (Just try clicking the cube!).

Objectives

After this lesson you will be able to:

- Insert a graphic button in your web page that links to another HTML document.
- Insert a small graphic that acts as a "postage stamp" button for a link to display a larger copy of the image.

Lesson

A HyperLink Button

From the previous lessons, you have (hopefully) become comfortable with creating hypertext, text within your documents that connects a viewer to related information. You can also use inline images (see Lesson 7a) to act in a "hyper" manner. If you recall, in an earlier lesson we linked some text in our **`Volcano Web`** page, **`index.html`**, to a second page, **`msh.html`**, that described Mount St. Helens. Now in the latter page, we want to add a button that when clicked will link a reader back to the main lesson page.

The way to do this is to put the HTML tags for inline images *within* the hypertext portion of the anchor tag:

```
<a href="fileX.html"> <img src="graphic.gif">
Go to Document X</a>
```

In your web page, this HTML code will display an inline image and the text **`Go to Document X`**. Both will act as hyperlinks; clicking on either the text or the picture will tell your web browser to go to the HTML file **`fileX.html`**. The image alone could be a hyperlink. In the World Wide Web, a "HyperGraphic" generally is surrounded by a colored box matching the color of hypertext on your web page, so you know that it is "active".

NOTE: Many browsers now can alter the color of hypertext and some pages have suppressed the display of the outline around HyperGraphic links. Generally, you can identify a hyperlink area on a web page by looking for a change in the cursor as it passes over a "hot" region. The cursor usually changes from an arrow to a hand

when it passes over an active link.

From a design standpoint, we recommend that if you use pictures to act as hyperlinks that you offer also a text link or use the ALT= attribute in the <IMG...> tag in case the viewer has turned off the loading of images.

Now we will create a "hyper" graphic button. First, you need to get a copy of an arrow button from the Lesson 8e Image Studio.

You should now have a copy of the image file somewhere on your computer (You should move it to the **`pictures`** folder/directory in your workarea).

Next, add the HTML to make the button work:

1. Open the second HTML file, **`msh.html`** in your text editor.
2. At the bottom, modify the last line to:

```
3.
4.   <hr>
5.   <a href="index.html#usa"> <img src="../pictures/left.gif">
6.   Return to <i>Volcano Web</i></a>
```

Note: The `inline`image tag (<img...>) is completely within the anchor between the `>` that marks the end of the URL and the `</a>` that marks the end of the hypertext. Also note how the `<i>` tag is used within the active hypertext to emphasize the title of the lesson. And finally, we have used the `<hr>` tag to put a horizontal rule or a line above the button graphic (for more on this tag see lesson 4).

7. **Save** the HTML file.
8. Return to your web browser, and select **Open Local** from the **File** menu to read in the **`msh.html`** file.
9. Select **Reload/Refresh** from the **File** menu to update the changes.
10. Test the hyperlink to the Mount St. Helens page and the new button that should return you to the *Volcano Web* page.

"Postage Stamp" Images

Previously, we advised against using large inline images in your web pages because viewers might have to wait a long time for images to download to their computer. One way around this is to create miniature-size copies of the graphic, or "postage stamps" which are displayed as inline graphics. Then, using the same steps as above, you can

make the "postage stamp" image act as a hyperlink that links to the full size image. In this way, the large images are downloaded **only** if the viewer decides to see it.

First, you need to get a copy of the two image files from the Lesson 8e Image Studio. (These files should be stored in your **`pictures`** folder/directory of your workarea).

Next, create the postage stamp link in your main text file:

1. Open the **`index.html`** file in your text editor.
2. Under the heading **Long Valley** enter the following:

```
    This field seismometer measures earthquakes associated
    with subsurface volcanic forces and may help to predict
    future events. It sits on a plateau known as the "Volcanic
    Tableland" formed by a major eruption  600,000 years ago.
    <p>
    <a href="../pictures/seismo.jpg">
      <img src="../pictures/stamp.gif" ALT="link to large image"
      WIDTH="62" HEIGHT="85">
      -- [full size image, 55k] --
    </a>
```

The inline image, `stamp.gif` acts as a hyperlink to a larger image, `seismo.jpg`. When a user clicks on either the "postage stamp" or the text "-- [full size image, 55k] --", your web browser will display the larger image in a browser page.

Note the use of the dimensions of the `stamp.gif` image in the `<img...>` tag as well as the `ALT=...` attribute.

In our hypertext link we provide information that this image is 55k in size. By doing this, you provide the viewer the choice if they want to download an image of that size... If the link leads to something that is 1.6 Mb, as a viewer you might want to know that before you tried to view such a large file size.

13. **Save** and **Reload/Refresh** in your web browser.

Check Your Work

Some of the more common mistakes are discrepancies between spelling of the file names and the HTML code for the anchor links or not having the image files in the same directory as the HTML files. If you see an icon of a broken picture:

then it usually means the HTML does not match the file listed in the **<img>** tag or that the image is not in a GIF or JPEG format.

Review

Review topics for this lesson:

1. How did you create the graphic button in your web page?
2. How are "postage stamp" links useful in including graphics in your web page?
3. How did you create your "postage stamp" link in your document?

Independent Practice

Try to add buttons that link two web pages to each other. In a later lesson we will learn how to avoide the "box" around a hypergraphic.

That wraps it up for “Writing HTML” guide. You might want to try analyzing some smaller, ready-made HTML files on your own.

Further HTML and JavaScript classes and documents are available through the church as well as beginner and advanced classes/documents on all computer science topics, with an emphasis on object oriented programming.

Troll X
Principal MIS
CULM IT/IS Group
Church of Universal Love and Music
www.Ilovemusicchurch.org
closingtimeproductions@yahoo.com
Admin