

# Basic Website Maintenance Tutorial\*

## Introduction

You finally have your business online! This tutorial will teach you the basics you need to know to keep your site updated and working properly. It is important to note that this tutorial is not a substitute for a good web designer and cannot replace comprehensive web design training. However, you should come away from the tutorial with an understanding of these basic concepts:

1. How to copy your site to your hard drive.
2. Basic structure and requirements of XHTML tags
3. How to insert text on a web page
4. Understanding URLs and paths
5. How to write a link
6. How to insert an image on a web page
7. How to edit a website menu
8. How to add a page to your site

For information on uploading your pages to the server, see the FTP tutorial.

\*This tutorial is compatible with Windows systems.

## Copying Your Website

You should always have at least one back-up copy of your site to prevent its loss or corruption by power surges, system crashes, or errors you create when making changes in the site. Later in the tutorial you'll learn about FTP software and you will be able to apply these concepts you learn now to copy your site directly from the server. For now, you'll learn to copy your site from a CD onto your harddrive. It is recommended that you maintain a copy of your site at all times, before and after you begin working on it. You may want to copy your site to two separate directories; one that will hold your safety copy and one that will hold your working copy.

### To copy your website:

1. Insert the CD containing the files for your website into your CD drive.
2. From the Windows Start menu, double-click My Computer.
3. In My Computer directory, double-click the drive where your CD is located. The CD directory will open, listing all the files on the CD.
4. Click on a file to highlight it, then press Cntl-c to copy it to the system clipboard.

**Note:** You can copy more than one file at a time by holding the shift key and using the arrow keys to highlight additional files. Press Cntl-c to copy the files to the system clipboard.

5. From the Windows Start menu, double-click My Documents. The My Documents directory opens.
6. Create a new folder in My Documents by clicking Make a New Folder in the menu on the left (WinXP) or right-clicking anywhere in the directory and selecting New>Folder from the pop-up menu. Type a name in the folder's highlighted space and click enter.

**Note:** You can create the folder for your site anywhere on your system; for simplicity, this tutorial uses the My Documents directory.

7. Open the folder you just created by double-clicking it. A blank directory will open. Click anywhere in the directory and press Cntl-v to paste the contents of the clipboard into the directory. If you weren't able to copy all the files from your site in step 4, go back and finish copying the remaining files into the new directory on your hard drive.

# XHTML Tag Structure & Requirements

We do not recommend that you alter styles in your website—doing so can literally make your website fall apart. However, you will need to update your content from time to time. While it is not the purpose of this tutorial to teach you XHTML code, you need to have an understanding of the basic structure and requirements of XHTML tags to keep your site valid and functioning properly.

XHTML is the upgraded version of HTML and demands these requirements:

1. XHTML does not allow presentation tags, such as `<font>` `<bg color="blue">` etc. Instead, these presentation elements or *styles* are applied in CSS (Cascading Style Sheets) and/or internal styles in the head of the document.

**Note:** Again we **strongly** urge you not to tinker with the styles in your site.

2. All XHTML tags and their attributes must be lowercase.
3. All XHTML tags must be closed. Most tags come in pairs such as these table tags:

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

The first bracket opens the table tag. The slash character / in the second bracket closes the table tag. Some tags do not come in pairs, such as this horizontal rule tag:

```
<hr />
```

Instead of a closing bracket as in the table tags above, a slash character closes a single bracket tag as in the horizontal rule tag. Notice the closing slash is preceded by a space.

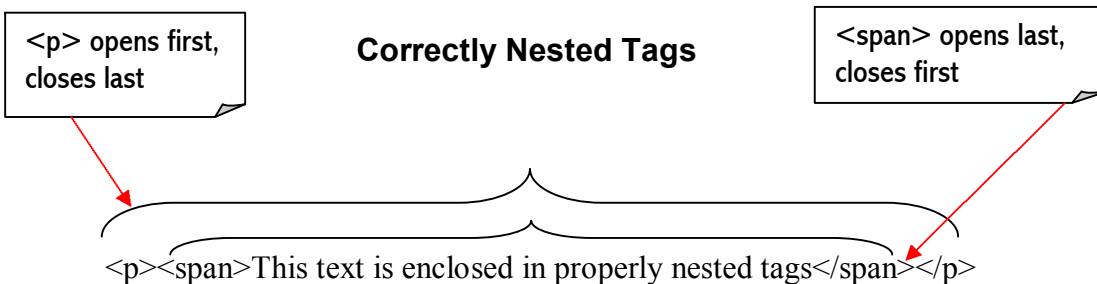
4. All XHTML documents must have a document type declaration. Notice that every page in your website begins with the following required code:

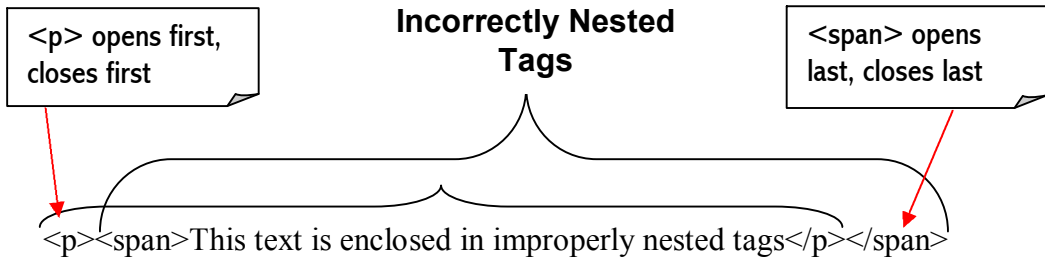
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

**Note:** Any new pages you create in your site must also begin with this declaration.

5. All tags must be properly nested. This means when a set of tags is enclosed in another set of tags, inner tags should close before outer tags, in the opposite order in which they were opened.

**Example:**





If you will adhere to these rules as you make changes in your pages, you should have no difficulty maintaining the validity and function of your site.

## Preparing The Tutorial Document

Now you're going to get some practice inserting text into a web page. Before you begin this part of the tutorial, create a folder in the My Documents directory of your hard drive where you'll save all the documents you'll be creating in the tutorial. Name the folder Web Tutorial.

You will need a text editor for this part of the tutorial. On a Windows computer you can use Windows Notepad. Open Notepad from the Start menu>All Programs>Accessories>Notepad.

**Note:** Do not use a word processing program, such as Microsoft Word, to create HTML documents. Such programs generate code that you do not want in your pages. Use *only* a text editor.

Highlight and copy (Ctrl+c) the code below, beginning with the tag that contains the document type declaration and ending with </html>.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
</head>

<body>
<p>Quis nostrud exercitation ut labore et dolore magna aliqua. Consectetur adipisicing
elit, velit esse cillum dolore in reprehenderit in voluptate. Eu fugiat nulla
pariatur.</p>
</body>
</html>
```

Paste the copied text (Ctrl+v) into the new Notepad document and save the document in the Web Tutorial folder with the file name hello.html. Notice that you are replacing the default .txt file extension with the html extension. This extension causes the page to display as a web page instead of a text document. After you save the file, close it.

Now navigate to the Web Tutorial folder and double-click hello.html. The document opens as a web page. In the browser toolbar, click view>view source. You can also right-click and select view source.

AHA! The Notepad document opens to reveal the HTML code that creates your web page! Now you can see both sides of the web page: the pure text code that will be sent to a web browser and the displayed web page that results from the code being translated by the web browser.

# Inserting Text in Your Web Page

Now that you've created a document to practice on, let's learn how to replace text in the document.

Below is the code you copied into your document. I've bolded the text you need to replace. Notice this text is inside a set of paragraph tags `<p></p>`. In most circumstances, the text you will change in your own web pages will be enclosed in paragraph tags.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
</head>

<body>
<p>Quis nostrud exercitation ut labore et dolore magna aliqua. Consectetur
adipiscing
elit, velit esse cillum dolore in reprehenderit in voluptate. Eu fugiat nulla
pariatur.</p>
</body>
</html>
```

## To replace the paragraph text:

1. Click and drag the mouse or hold the shift key and use the arrow keys to highlight the text inside the paragraph tags.
2. With the text highlighted, begin typing your new text or paste (ctrl+v) text you've copied from another source.
3. Save your document (Ctrl+s) and open hello.html. The web page now displays the text you added.

**Note:** You can undo a paste or any other command by pressing ctrl+z.

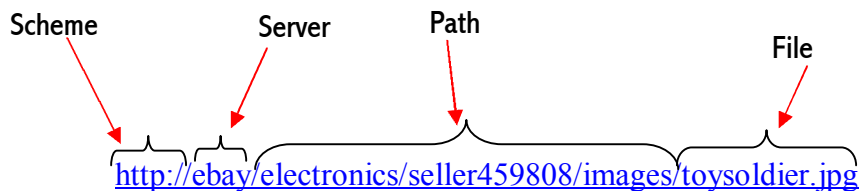
# Understanding URLs

You know how to add text to your web page. But what if you want to add a graphic? What if you want to link to another page or add a page to your web site? All of these are added by linking to the files where they are contained. To accomplish these things, you need to learn about paths and URLs.

The URL (Uniform Resource Locator) tells where a file is located and what the browser should do with it. The two types of URL you will be concerned about in this tutorial are *relative* and *absolute* URLs.

## Absolute URLs

Absolute URLs begin with `http://` which tells the browser that the file is on the internet. This file could be anywhere on the internet, *including* the site that holds the link. Absolute URLs show the entire path to the file, including the scheme (usually called the protocol), the server name, the complete path, and the file name.



## Relative URLs

Relative URLs tell the browser the file referenced in the link is located on the same site as the page containing the link.

In other words, absolute URLs tell the browser how to find the file from anywhere on the internet. Relative URLs tell the browser how to find the file *relative* to the document that contains the link.

In order to write the URL accurately so the browser can find the correct file, you need to understand **paths**. In the simplest terms, a path is a folder-by-folder map to a file. A linked file can be located in the [same folder](#), in a [folder lower](#) in the file hierarchy, or in a [higher folder](#) in the file hierarchy, depending on the location of the file containing the link.

## Paths to Same Folder Files

Figure 1 below shows a sitemap for a typical website with a hierarchy of folders. The file called skin.html contains a link to a file called shipping.html. Since the files are located in the same folder, the path is simply the file name, shipping.html.

### Example:

A link in skin.html points to shipping.html, located in the same folder:

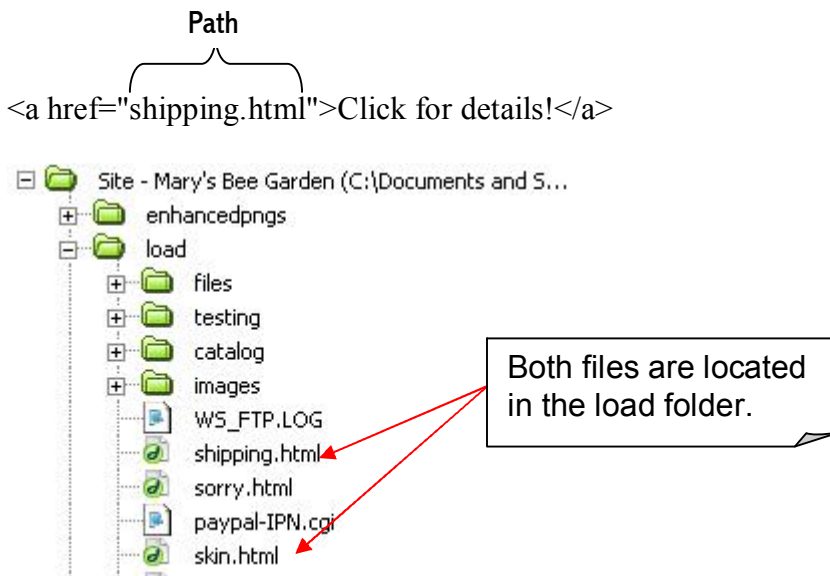


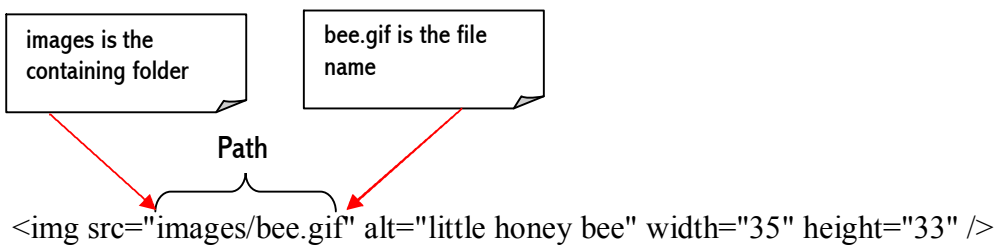
Figure 1

## Paths to Files Lower in File Hierarchy

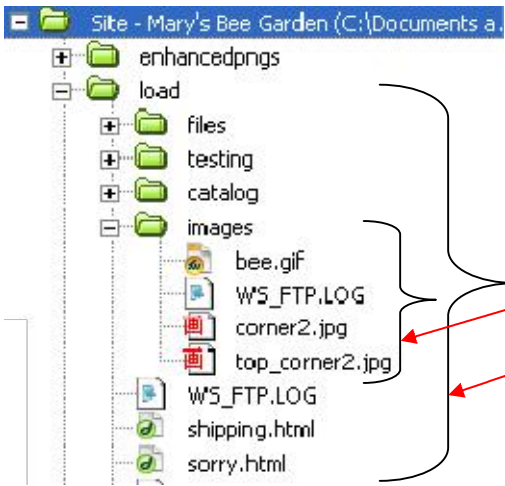
The file called shipping.html contains a link to the image file called bee.gif. Notice that bee.gif does not share the same folder as shipping.html. In this case, bee.gif is located in a folder *lower* than the load folder where shipping.html is located. The correct format for the path then is:

Containing folder/filename

### Example:







shipping.html contains a link to bee.gif

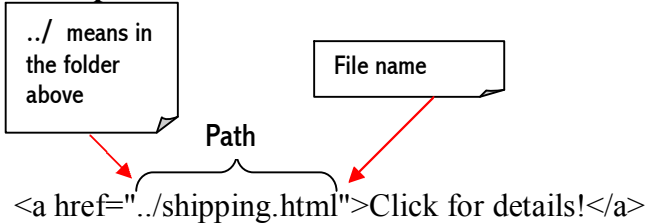
bee.gif is located in the folder **below** shipping.html

## Paths to Files Higher in File Hierarchy

The file named lotionsoap.html also contains a link to shipping.html. As illustrated in Figure 3, the files are not in the same folder. In this case, shipping.html is located in a folder *higher* than the catalog folder where lotionsoap.html is located. The correct format for the path then is:

../file name

### Example:



The `../` tells the browser the linked file is located in a folder one level higher than the folder containing the current file.

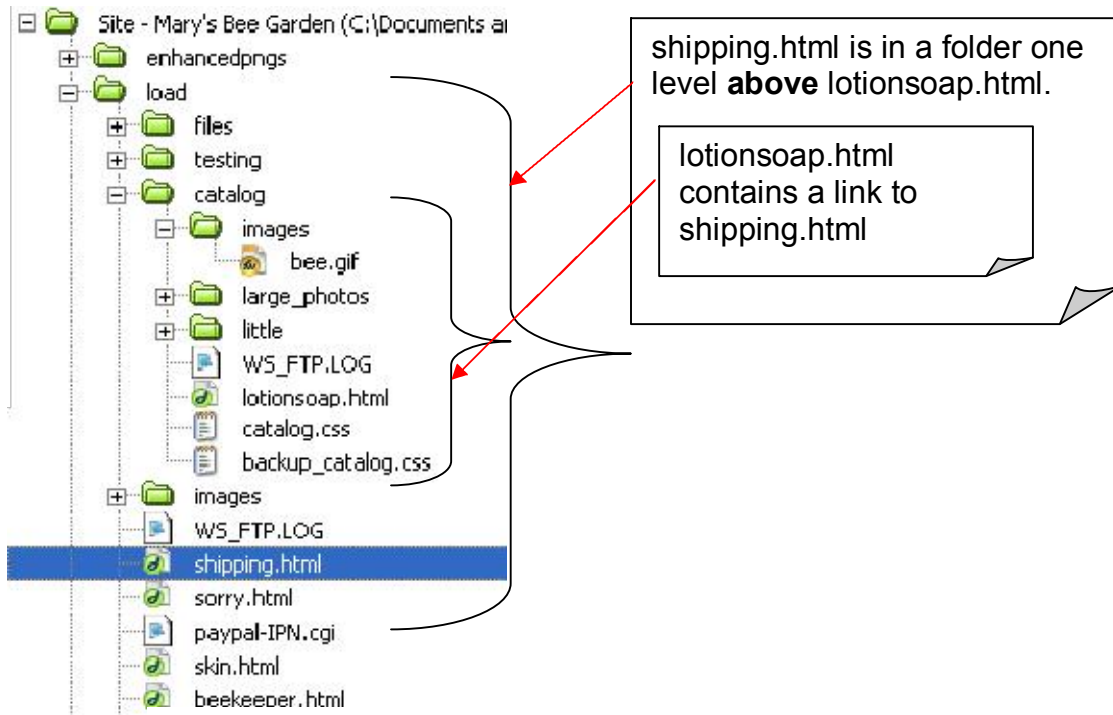


Figure 3

Understanding paths is important for adding linked elements to pages, and for saving and uploading files to the appropriate folders. It is very important to maintain the hierarchy of the files and folders when saving and uploading files. If you inadvertently change the file hierarchy when saving or uploading files, the paths in your site documents will be inaccurate and won't work. If you are having trouble with dead links, make sure the paths in your links reflect the hierarchy of files and folders in your site.

## Linking

Now that you understand the format required for paths, you are ready to learn to create links in your documents.

The format for a link is:

```
<a href="filename.html">display on webpage</a>
```

Earlier in the tutorial you created a web page called hello.html. Open hello.html, save as links.html. Now you have two documents in the same folder. Now you'll add code to each document that will link one to the other.

1. In the links.html source code, type the following:

```
<a href="hello.html">My Welcome Page</a>
```

2. Save the document (Ctrl+s)

3. Open hello.html and add the following in the source file:

```
<a href="links.html">My Favorite Links!</a>
```

4. Save the document (Ctrl+s)

5. Refresh the browser view of either document by pressing F5.

Your page should display one of the links you just created. When you click on the link it should take you to the page with the other link you created. Click on that link and it should return you to the previous page. If you have results different from this, go back and check to make sure you typed the code correctly. Pay careful attention to spelling, brackets `<>` and closing marks `/`.

When you've successfully created these links, let's create something with linking that is a bit more challenging by changing the hierarchy of the folders.

## Links with Complex Paths

As you've seen, creating simple links is a straightforward process. Creating links with more complex paths is requires a bit more care, but it's not difficult. To get some practice in this new challenge, we'll create a new folder named "lower" inside the Web Tutorial folder. Drag the links.html document into the lower folder. Now open either the links.html document or the hello.html document and click on the hyperlink. Both hyperlinks are dead because we changed the hierarchy of the folders and files.

Now we'll change the paths in the links to reflect the new hierarchy.



The link.html document contains a link to a file (hello.html) one level **above** it's own location. Recall the syntax for referring to a file in the folder above is:

```
../filename
```

**Note:** If the file were located two levels above, the syntax would be:

```
../../filename
```

So our hyperlink code from links.html to hello.html is:

```
<a href="../hello.html">My Welcome Page</a>
```

The hello.html document contains a link to a file (links.html) one level **below** it's own location. Recall the syntax for referring to a file in a folder below is:

Containing folder/filename

So our hyperlink code from hello.html to links.html is:

```
<a href="lower/links.html">My Favorite Links</a>
```

Replace the code for each hyperlink, save the documents and click the links again. Your links should once again link the pages without error. If you do not get these results, go back and check spelling, brackets and closing slashes.

## Adding Images to Web Pages

Now that you understand links and the importance of accurate paths, you will find it a very simple process to add images to your web pages. It is important to note that adding images can change your site in unpredictable ways. It is best to replace images with images of a similar size. If you are adding an image to an area of the site that previously has had no image, be certain that you are adding it where it will not interfere with navigation or logo areas. It is important to view your pages in different browsers to make sure the page design is stable after adding large elements (like images) that can effect the design.

The syntax for adding an image is:

path and filename  
of the image

```

```

Now we'll add an image to each of our web pages. To hello.html, we'll add aunt\_pearl.jpg. To links.html, we'll add tastee\_kreme.jpg. Notice that each html document occupies a different level in the hierarchy of the site. This will effect the way we write the path to the image files.



First we'll add an image to hello.html. Notice the images folder is one level below below the folder that holds hello.html

1. Open the source code for hello.html.
2. Immediately after the first <body> tag, add a set of paragraph tags:  
`<p></p>`
3. Between the paragraph tags, type the following:  
``
4. Save the file and view it in the browser.

Now we'll add an image to links.html. Notice the images folder is located a level above links.html. However, this path is a bit more complicated than for the image we just added. Although the folder named "lower" is located on the same level as the "images" folder, the server must climb up the lower folder to get to the images folder. Then it must climb down into the images folder.

1. Open the source code for hello.html.
2. Immediately after the first <body> tag, add a set of paragraph tags:

```
<p></p>
```

3. Between the paragraph tags, type the following:

```

```

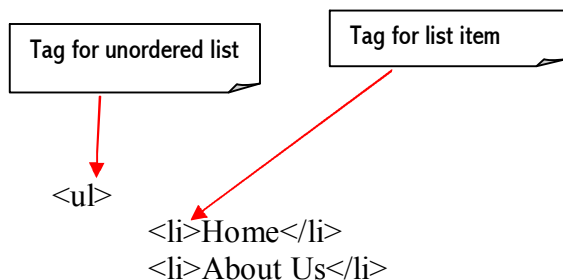
4. Save the file and view it in the browser.

## Editing Website Menus

Links are necessary for website menus. Commonly, website menus are presented in a list format. Because HTML lists are easily styled with CSS, it is very easy to add or remove menu items from menu lists.

### Creating Menu Lists

To create a menu list, copy the code below and paste it into hello.html.



```
<li>Contact Us</li>
<li>Sitemap</li>
</ul>
```

**Note:** Ordered lists, (numbered lists), open with the `<ol>` tag and close with the `</ol>` tag.

Save (Ctrl+s) the document then refresh (F5) the browser display. An unordered list (list without numbering) displays on the page.

To create a menu from the list, we have to enter a link for the list items. For the purpose of this tutorial, we're going to link the menu items to pages that don't exist— these links will look right, but they won't work. Feel free to create links to actual pages if you choose.

```
<ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="about.html">About Us</a></li>
  <li><a href="contact.html">Contact Us</a></li>
  <li><a href="sitemap.html">Sitemap</a></li>
</ul>
```

After you link the menu items as shown above, save (Ctrl+s) the document and refresh the browser (F5).

Now you have a list of links, or a menu. In most sites, menu lists will be styled and appear much more attractive than ours when displayed in a browser. Nevertheless, the code in the HTML styled lists will look very similar to ours.

## Editing Menu Lists

Now that you have a menu, let's see how easy it is to change it. For this example, we'll add a links for our `hello.html` and `links.html` pages. We want the list items to display "Welcome" and "Links" in the browser. The link for `hello.html` should appear after the home page and the link for the `links.html` should appear just above the sitemap link.

To add an item to the menu, you'll need a set of `<li>` tags for the list item and `<a href>` tags for the hyperlink:

1. Type a set of list item tags where you want the new menu item to appear.

```
<li></li>
```

2. Type the code for the link between the `<li>` tags.

```
<li><a href="hello.html"></a></li>
```

3. Between the link tags, type the text you want displayed in the browser.

```
<li><a href="hello.html">Welcome</a></li>
```

4. Save the document and refresh the browser.

In this tutorial we've done things step-by-step to illustrate how HTML works. But you can take some shortcuts to speed your update time. A fast way to add a menu item is to

copy an existing item, paste it where you want your new link, then change the display text and link for the new item. Do this now to add the links.html document to the menu just above the sitemap. Remember the link code must reflect the path to the links.html document!

Your new code should look like this:

```
<ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="hello.html">Welcome</a></li>
  <li><a href="about.html">About Us</a></li>
  <li><a href="contact.html">Contact Us</a></li>
  <li><a href="lower/links.html">Links</a></li>
  <li><a href="sitemap.html">Sitemap</a></li>
</ul>
```

Now let's remove an item from the menu. As you might have guessed, this is an easy thing to do. Simply highlight the entire list item for that menu and hit delete. Remove the menu item for the sitemap.

Your new code should look like this:

```
<ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="hello.html">Welcome</a></li>
  <li><a href="about.html">About Us</a></li>
  <li><a href="contact.html">Contact Us</a></li>
  <li><a href="links.html">Links</a></li>
</ul>
```

One **very important** thing to remember when you change any menu details on a site is that menus are typically used across entire sites. If you are using static pages on a website, you must change **every occurrence** of the menu in the site. This can be time consuming and greatly increases the likelihood of errors on your site. So if you must change a menu, change it, but be accurate and consistent in your changes. Depending on the size of your website, it is often more cost efficient to get your web designer to make these types of changes. What takes you hours could take a web designer a mere few minutes.

## Adding and Removing Pages in Your Site

Now that you understand how to create links and how to edit menus, it's easy to understand how to add or remove pages from your site.

### Adding Pages

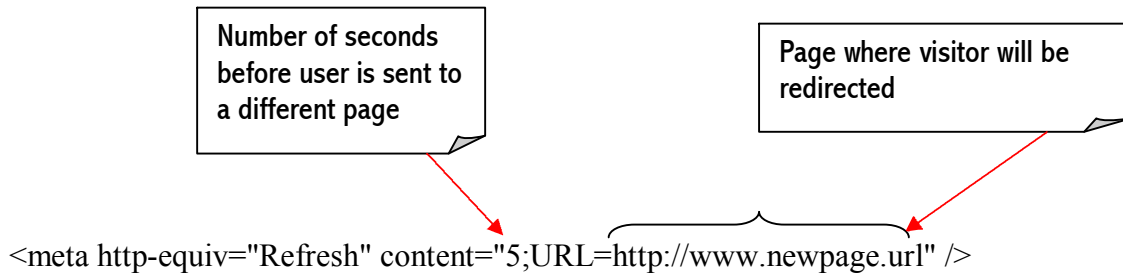
Earlier in the tutorial we created the links.html page simply by opening the hello.html page and saving it with a new name. This did not change the hello.html page, but created a brand new page, identical to hello.html. You can create a new page for your website using this same method. Not only will your page be laid out correctly, but it will also contain all the necessary links to style sheets and images used in your page design and menus.

After you've created a new page from an existing one, all you have to do is edit the details of the contents as we have been doing throughout this tutorial.

## Removing Pages

Removing pages from your site is as simple as removing them from menus and if necessary, deleting them from the server.

Of course, you should be very cautious about removing pages from you site because other sites may link to your site through the very page you remove! Rather than removing a page completely from the server, you may want to add a simple redirect in the <head>section of the page:



Whether you add or delete pages from your site, you'll probably want some or all of the menus on your site to reflect these changes. As discussed earlier, this can be a big job depending on your site.

When you've made all your changes to the new and existing pages and changed your menus accordingly, you'll upload it to your server.