ZERO TO HERO

HTML

Learn HTML Programmig With Ultimate Zero to Hero Programming Crash Course for Beginners

PAUL MADOFF

HTML

Learn HTML With Ultimate Zero to Hero Programming Crash Course for Beginners

Paul Madoff

Zero to Hero

Introduction

I want to thank you and congratulate you for downloading the book, *“HTML: Learn HTML With Ultimate Zero to Hero Programming Crash Course for Beginners”*.

This book contains proven steps and strategies to learn how to create programs using HTML, the language of the internet – even if you currently have 0 or minimal knowledge about the programming language.

This eBook will teach you the basics of writing programs in HTML. Although you may feel that using a WYSIWYG HTML editor can produce faster results, these editors come with major flaws that you will be better off learning to write your own codes in HTML. In no time, you can create functional and better looking webpages.

Thanks again for downloading this book, I hope you enjoy it!

Chapter 1: Creating Your First HTML Page

HTML is the language behind the Internet. It is what makes the web tick. Most websites that you visit probably use this language. Almost all computers in the world understand it as it is among the most universal ways to create documents. It may not be equipped with the best formatting tools available, and there may be no guarantees that your webpages will look and work exactly the same way for every type of browser. However, to say it bluntly, without HTML, the internet, as we know it today, would not exist.

Of course, you can always choose to use a WYSIWYG HTML editor to build websites. However, you have to be aware of the 3 maid advantages that come with such programs:

1. The codes they create are not always fully compatible.

2. WYSIWYG or "what you see is what you get" editors normally use excess codes in creating a specific look on a page that only tend to slow down the loading times.

3. Some editors, change the HTML codes you enter manually.

Due to these glaring cons, you are better off writing your HTML codes the old-fashioned way - manually. It may be true that many programmers don't do it simply because it takes longer to write by hand because it requires you to have a bit of understanding about the language, still you can be sure that once you learn how to program using HTML, you will be able to come up with better-looking and better-functioning web pages.

The goal of this eBook is to help you learn the basics in creating programs using HTML.

What Software Do You Need?

There is no specialist software that you need to buy or install in order to create HTML codes. In fact, a lot of experienced web designers claim that the best websites they have created were written in good old trusty Notepad! Although you can use Notepad for this eBook, we recommend using FrontPage. After all, it is free and you can easily download it.

One of the main benefits of using an HTML editor is that it will make your codes easier to read by color coding them. You can easily “clean up” your script once you are done, and there are buttons that come with the software that you can use to insert repetitive codes.

At any rate, it doesn’t matter whether you use FirstPage 200, Notepad, or any other HTML editor. This eBook will teach you the basics of the language, regardless of the editor you prefer.

Understanding HTML

HTML, per se, is very easy to understand and learn – if you know the basics. HTML is primarily made up of tags, which are actually pieces of text enclosed in <>. This is how it typically looks like.

<tag>

Opening tags and closing tags are the two types of tag. The main difference between the two is that a closing tag has a / before the text, *i.e.*

```
</tag>
```

Tags always come in pairs like the example below:

```
<tag></tag>
```

Don’t worry if, at this point, you find the above confusing. It will be explained in more detail, anyway. For now, remember that anything you find in-between the opening and closing tags means that the tags are applicable to them. Let us use the <center> tag for this example. This tag is used to center align text, and the syntax is something like this:

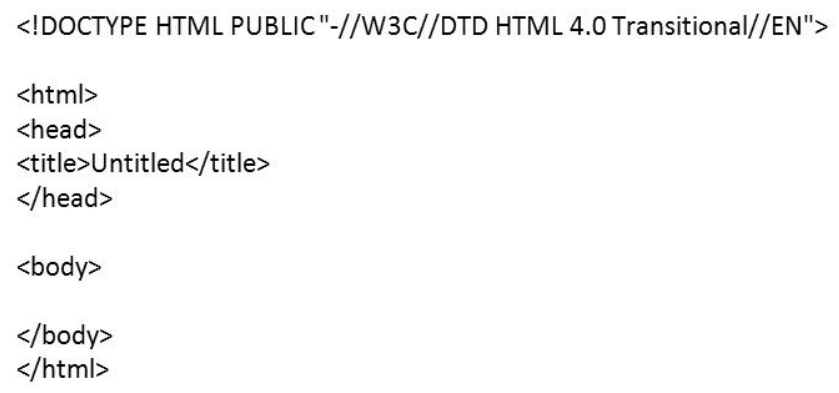
```
<center>This line of text is centered</center>
```

Almost all tags come with a closing tag, but there are a few that don’t. Just remember this basic format:

```
<Tag>Text</Tag>
```

How to Declare HTML

Now, open the HTML editor you prefer to use. Most HTML editors already contain some entered codes. If yours does not have any, just enter the following script:



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
<title>Untitled</title>
</head>

<body>

</body>
</html>
```

Let’s discuss each of the lines of code above. The first line

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

Tells the browser which specific language is to be used for the page. Although it is not really necessary, it is still ideal to include it.

The next line,

<html>

gives the instruction to the browser that the HTML document begins here.

<head>

You guessed it right! The header section starts here. This section contains all the page configuration options including the title.

```
<title>Untitled</title>
```

Here, the browser is told what title to display for the page that will appear in the title bar found at the top portion of the browser. Just input the title you want for your page in-between the <title>tags.

```
</head>
```

By now, you should already be able to tell that this ends the header section of the code.

```
<body>
```

```
</body>
```

Anything you put between the two tags will form part of the web page’s body. You will usually find texts, images, and videos, among others, here. Obviously, this is the most important part of the webpage as it contains all the meaty parts, so to speak.

```
</html>
```

The tag indicates the end of the script.

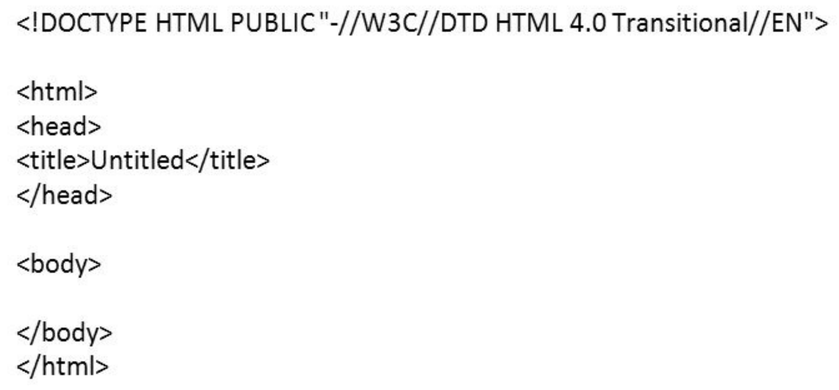
Chapter Summary

To sum everything we covered in this chapter, you learned how to structure your HTML code using the <tag>text</tag> format. Keep in mind that tags will apply to anything found between an opening tag <tag> and a closing tag </tag>. Finally, you now know how to declare HTML documents and how the title of the page is set.

In the next chapter, we will cover adding text to pages and formatting them.

Chapter 2: Adding Text

Let us begin the chapter by reviewing the HTML code we created in the previous chapter.



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
<title>Untitled</title>
</head>

<body>

</body>
</html>
```

Now, let us discuss how you can create a basic and simple homepage. First you need to add a title for the page. To do this, simply replace Untitled to, for example, “My Own Homepage”.

In the code above, just modify the appropriate section form

<title>Untitled</title>

to

<title>My Own Homepage</title>

Using the <font> Tag

The <font> tag is among the most versatile tags in HTML, and one of the most commonly-used. When used in its basic form, the pair of tags will display text on the webpage in the default font. Let’s begin by displaying the following text:

Welcome to Our Homepage

To display the text on the screen, you have to do the following code:

```
<font>Welcome to Our Homepage</font>
```

The code above should be placed between the <body> tags. The text will then be displayed in the standard font size, in Times New Roman, and in black. You’re right; it is not the most interesting and most attractive look that your homepage can have.

The Size, Color, and Face

For every string of text, you can set how it will appear on the page in terms of size, color, and face; and these are the initial tag attributes that you will encounter in this tutorial. Let us begin with the first attribute, FACE. Rather than adding a new line for the font face tag (or the font you will view the text on) to the code, you instead create the font tag this way:

```
<font face="Verdana">Welcome to Our Homepage</font>
```

You will notice that the font name is enclosed in “ “ after an ‘equal to’ sign. You don’t have to include it, however, in the end tag.

You can add more than just one attribute to a tag; thus it is very easy to display the text in a bigger or smaller size. One thing that you need to remember, though, is that the sizes used in HTML aren’t like the typical font sizes that are measured in terms of point sizes or pt. HTML sizes are expressed in a single digit that relates to a standard font size. To illustrate:

HTML Font Size	Standard Font Size
1	8 pt
2	10 pt
3	12 pt
4	14 pt
5	18 pt
6	24 pt
7	36 pt

If you wish to make the title larger, just modify the tag to look something like this:

```
<font face="Verdana" size="7">Welcome to Our Homepage</font>
```

By this time, you may have already noticed that it is quite easy to manipulate and add options once you have learned how to properly use a tag. Just make sure to use American English to spell the tags. The color attribute is a bit unlike the others in the sense that it can be modified using HTML color words (standard color names), although not all color names will work with this command. HEX codes can likewise be used. These are codes that use the format # followed by six digits (#000000). The first two digits represent the amount of red color to be used, the next two for green, and the last two digits for blue. To make the text color red, you can use either of these two codes:

```
<font face="Verdana" size="7" color="red">Welcome to Our Homepage</font>
```

or

```
<font face="Verdana" size="7" color="#FF0000">Welcome to Our Homepage</font>
```

Centering the Text

You may want to see the text at the center of the page so it would look like a typical title. You can do this by using the <center> tag. Just input everything you want to be centered between the <center> tags. Here is how you do it:

```
<center>
<font face="Verdana" size="7" color="red">Welcome to Our Homepage</font>
</center>
```

On the screen, you will see the following text:

Welcome to Our Homepage

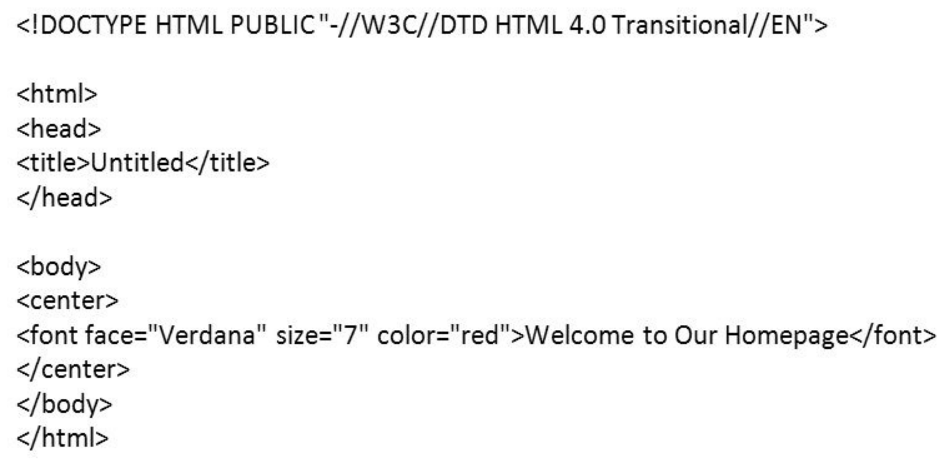
Chapter Summary

To sum up what we have so far discussed, you learned how you can display text on your webpage and how you can for mat the text size, color, and font. You can now also center items on the page. Finally, we you have created your own code to use for your site.

For the next chapter, we will cover more topics on text formatting including making paragraphs and positioning text on the page. We will also discuss other tags that you can use to help in your webpage design including making lists.

Chapter 3: Positioning Text

Let’s start with the script you created in the previous chapter that should by now look something like this:



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
<title>Untitled</title>
</head>

<body>
<center>
<font face="Verdana" size="7" color="red">Welcome to Our Homepage</font>
</center>
</body>
</html>
```

The code above should display the phrase “Welcome to Our Homepage” in big, red, Verdana font right at the center of your webpage.

Now, we will discuss how you can position the text (or whatever you like) on the page. You will likewise learn how to use other helpful tags in HTML.

The <p> Tag

The P in the <p> tag is for paragraph. The tag is used to break the lines of text to form paragraphs. When defining a paragraph, just insert the text inside the opening and closing <p> tags. The enclosed text will then be grouped into a paragraph and a blank space will be created at the end of the paragraph similar to how the paragraphs are spaced in this eBook.

There is an attribute that can be added to the <p> tag – the align option. This option allows you to specify any of the three types of text alignment similar to how a word processor does it: left, center, or right. Following is an example of a code written to align text to the left:

```
<p align=”left”>Text</p>
```

With the <p> tag, you now have two options on centering text. How you do it is entirely up to you. You can either use the <p align=”center”> tag or the <center> tag. Although both codes produce the same results, most programmers prefer to use the latter as it is shorter and easier to type. Being shorter, it will also help reduce the page’s loading time. Among the three <p> tags, the align=left option is the least used. Why? – Simply because most browsers align text to the left by default, but there are some programmers who use it just to be sure.

The <Br> Tag

There may be times when you do not feel like leaving a space after paragraphs. You can do this by using the <Br> (or break) tag. This tag comes in handy whenever you need to begin a new line. Anywhere you insert it, the tag will initiate a break and start a new line. If you want to start a new line without leaving a space, just use the <Br> tag. Use <Br><Br> when you want a line break. Note that no closing tag is necessary for the <Br> tag.

For instance:

The text here is on a line
The text here is written on the next line

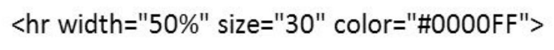
The text here appears after a line break

The text here appears after 3 <Br> tags.

The <hr> Tag

Another very useful tag, the <hr> tag is used to break up a page. When you use it, a horizontal line similar to the one below will be inserted.

You can tell that this tag is very simple to use. No closing tag is required. Although there are a number of attributes available for this tag, they are used only on rare occasions. One option is to modify the height text height (in pixels), the text width (in terms of pixels or % of window), and the text color (available only for Internet Explorer). Following is an example of how you can create a line that is 30 pixels tall, 50% of the window, and in blue color, if you are using IE, that is. Otherwise, you may see it in any shade of gray.



```
<hr width="50%" size="30" color="#0000FF">
```

Comment Tags

A comment tag comes in handy whenever you need to insert notes in your HTML script that you do not want visitors to see on your page such as for copyright notices. You can also use comment tags to provide information on what a particular section of the code is about, to send notes to people who will read your codes, or for anything else you deem fit. There are webhosts who insert comments to mark specific areas to insert banners. In some cases, the cue is a specific comment that you add. When adding comments, use the following format:

```
<!-- Your comment -->
```

Anything enclosed in a <!-- --> tag will be ignored by the browser.

Chapter 4: Hyperlinks and Bookmarks

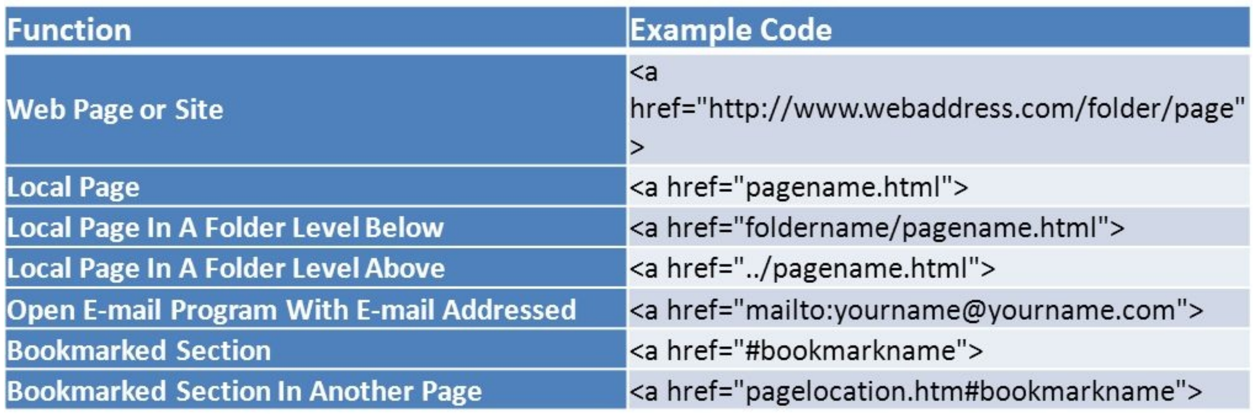
If you regularly surf the net, you should be familiar with hyperlinks – what it is, and what it is for. In the remote case that you don't have any clue on hyperlinks, it is simply a sting of text that once clicked, will take you to another webpage. Another term you should be familiar with is bookmarking. It is a method of marking a webpage or a point in your page so you can easily hyperlink back to it, anytime you want to. This chapter will be all about hyperlinks and bookmarks.

The <a> Tag

You use the <a> tag when you want to create hyperlinks and bookmarks. The "a" stands for anchor. We will be discussing its functions for the rest of this chapter.

<a href>

Href is short for Hyperlink REFerence. The href variable is necessary when creating hyperlinks. This is done by enclosing an image or text inside a pair of href tags like in the example below: <a href="pageurlhere">Place Text Here</a>
You can use hyperlinks to specify a number of things



Function	Example Code
Web Page or Site	<a href="http://www.webaddress.com/folder/page">
Local Page	<a href="pagename.html">
Local Page In A Folder Level Below	<a href="foldername/pagename.html">
Local Page In A Folder Level Above	<a href="../pagename.html">
Open E-mail Program With E-mail Addressed	<a href="mailto:yourname@yourname.com">
Bookmarked Section	<a href="#bookmarkname">
Bookmarked Section In Another Page	<a href="pagelocation.htm#bookmarkname">

Bookmarks

Creating bookmarks on a page is very easy using <a> tags. This time, however, instead of the href variable, the name variable is used, just like in the example below: <a name="top"> First Line of Text In My Favorite Page</a> The code above will create a bookmark named top that will bring you to the text enclosed by the opening and closing tags. This can then be linked through a standard hyperlink like this: <a href="#top">Go Back To the Top</a>

Anything can be used to name bookmarks. It’s all up to you. Bookmarks come in handy in very long pages as they can be used to get you to other parts of the page quickly.

Chapter 5: Images and Backgrounds

Images play a very important role for an HTML page. They make a page look more interesting when used as background, and they make the page look a lot better when used as a design element. Simply put, images provide the big difference that distinguishes HTML pages from regular e-mails and printed pages.

HTML and Images

You can easily add an image to a page. That is what the <img> tag is for. However, you need to learn how to use variables with the tags to make them work. Otherwise, all you will get are plain and bland pages that will not be helpful for your website.

When inserting images, you have to use the src= variable to select the particular image you want inserted. Similar to a hyperlink, the variable can either be a direct reference (such as a website's URL) or just a relative reference. Following is an example on how to do it:

```
<img src="http://www.gowansnet.com/images/gnet.gif">
```

By default, including images in hyperlinks will display the images sporting a blue border around them. You can turn off the border, though simply by using the border=”0” variable when declaring the image.

Resizing Images

Basically, there are two variables that you can use to resize an image inside the browser – height and width. It is recommended that you specify the size using the given variables even if you have no intention of resizing the image. For one, the image will appear in the browser with a placeholder so there will be no major changes to the page when loading the image.

You can also use the tags to make the images smaller or larger. The dimensions are expressed in pixels like in the example below:

```
<img src="http://www.gowansnet.com/images/gnet.gif" border="0" width="80" height="30">
```

or

```
<img src="http://www.gowansnet.com/images/gnet.gif" border="0" width="10" height="10">
```

Note that if you are trying to make the image smaller, it is perhaps better that you use an image editing program to resize it. This is because if you use an editing program, the image will load more quickly because of the resulting image’s smaller file size. You lose this advantage if resizing is done in the browser.

Alt

The last image variable is the alt. Using it will instruct the browser on what alternative text to use for the image in case the browser’s image settings are turned off. Following is an example on how the variable is used:

```
<img src="http://www.gowansnet.com/images/gnet.gif" alt=" Gowansnet Logo">
```

One good tip to remember is to use jpeg or gif images because these file types come with smaller file sizes, and therefore would load faster.

Background Colors

The page background colors can easily be changed through the <body> tag’s bgcolor variable. This is how it is used:

```
<body bgcolor="#0000FF">
```

The code above would change the page background color to blue. As an alternative, you can also use a color world in HTML.

The bgcolor variable is pretty straightforward and simple to use as there is not much complications in it. When thinking of the text colors to use on your background, remember that light-colored text will only work on dark backgrounds and dark-colored text work best with light backgrounds. For instance, never use red on blue, or blue on red. In general, black text on white background is ideal.

Background Images

As previously mentioned, a background image can be used on a webpage. To define simply, a background images are images tiled behind the text on a webpage. There is a <body> tag variable that can be used for this purpose, and it is used this way:

```
<body background="mypic.gif">
```

In some cases, including a background color is a good idea, so your visitors can clearly read the text on a comfortable color while waiting for the background image to load. Again, do not forget the rule of thumb in using color contrasts.

Chapter Summary

To recap what we have so far discussed in this book, you have learned how you can create HTML pages, format text, split text into paragraphs, insert images, create links between webpages, and create links that will direct you to various parts of your webpage

Chapter 6: Advanced Text Formatting

There are a lot of great elements that make up HTML. Unfortunately, many of these elements are overlooked or simply not known to most web designers today. While you can already create and develop a decent website just with sufficient knowledge of HTML basics, you can make fully compatible and more useful pages by taking advantage of the numerous advanced features of the program. Thus, it is highly recommended that you learn these little-known tags.

More <font> By now, you should be aware the one of the most-frequently used HML tags is the font tag. To recap, this is the tag used to specify the text font, color, and size. Today, a lot of developers think that this tag is obsolete because of the bigger freedom and control provided by stylesheets. However, the tag still offers an easy way to modify a webpage's look.

Basically, the font tag is used to set the text font this way: <font face="Verdana">Text</font> This, however, poses a problem, specifically that of various types of computers accessing the page. Unlike printing that relies on the printer to render the page to be printed, displaying a page on screen depends on the user's computer; thus, a webpage can look different on different computers. The difference becomes even more noticeable with the attribute font face, especially when the user's computer doesn't have the specified font installed. Thus, HTML has a built-in system that allows specifying multiple fonts. Following is an example on how you can do it: <font face="Arial, Verdana, Sans-serif"> The code above tells the user's browser to first try the Arial font, and if it is not found, then try Verdana, and if it still isn't found, then go for the standard font – Sans-serif. With this tag, you have some control over how your webpage is displayed by the browser that does not have the appropriate font. Admittedly, it is not perfect, but the tag certainly has good uses, especially for compatibility purposes, one of which is if you don't want to use a stylesheet but want a non-standard font for your page. It is also great to use the tag if you know how your pages will look on other computers.

Italic, Bold, Strikethrough, and Underline Another advanced method to control text appearance, aside from modifying the font, size, and color, is to have the 4 standard text formats applied, which all have corresponding tags in HTML.

For bold text, you have two options. These are: <b>Text</b> or

<strong>Text</strong> You can use either of the above options. For accessibility purposes, however, experts recommend using the <strong> option. This is because screen readers, or programs that read webpages to blind individuals, will easily see that the text is in bold, and will express it by speaking in the appropriate tone.

In like manner, you can italicize text this by including: <i>Text</i>

or

<em>Text</em>

Again, it is up to you on which option to use, although screen readers can usually recognize the <em> option more easily, and emphasize the text accordingly.

To underline text, use: <u>Text</u> It is, however, recommended not to use this attribute unless really necessary as it might confuse readers who may think that the underlined text is a link to another page.

To strikethrough or cross out text, use: <strike>Text</strike> Strikethrough is one tag that is not frequently used, although it has some important uses such as when you need to track changes to your text.

Subscript and Superscript The subscript (text slightly lowered) and superscript (text slightly raised) tags are not often used as their most common uses are for scientific and mathematical applications, although there are many other uses for the tags. Likewise, the tags are not difficult to use. For example: 9 x 9 = 9<sup>2</sup>

$9 \times 9 = 9^2$

Preformatted Text HTML was designed in such a way that multiple spaces in documents are ignored. If it encounters 2 standard spaces, for example, HTML renders them as a single space. Though code indentation is allowed without any changes to the screen presentation, there are some types of content that are difficult to display like preformatted tables created in plain text. To address this problem, the <pre> tag can be used. "Pre" stands for preformatted text and is used to tell the browser to display the text in exactly the same way it does in the source docuent. It is inserted in the code this way: <pre> Code Location Sales

------ ---------- -------

1234 Miami 11,500

5678 Chicago 20,000

9101 LA 12,400

</pre>

If you do not use the <pre> tags, the text would appear this way:

Code Location Sales
------ ---------- -------
1234 Miami 11,500
5678 Chicago 20,000
9101 LA 12,400

However, after including the tags your text will display as: Code Location Sales
------ ---------- -------

1234 Miami 11,500

5678 Chicago 20,000

9101 LA 12,400

Chapter 7: Lists, Headings, and Base

Even with the wide acceptance of CSS, there are still many HTML features that are frequently used, and so far, have no real replacements yet. Among these is the header tag which has been proven to be very reliable when using CSS.

Lists

In various instances, you may need to write lists in HTML. Sure, it would be quite easy to just make the list by typing the text you want to include in the list as well as just numbering the list or putting asterisks (*) before each item. There are, however, more flexible and relatively easy methods in HTML, the most basic of which is the unordered or bulleted list. Following is an example of an unordered list: To-Do List

- Build Website
- Upload Webpages
- Become Rich

Here is how you will code it in HTML:

```
<ul>To-Do List:
<li>Build Website</li>
<li>Upload webpages</li>
<li>Become Rich</li>
</ul>
```

The <ul> tag lets the browser know that it is the beginning of an unordered list, and will prompt the browser to the text indented from that point onwards. The <li> opening and closing tags, on the other hand, indicate the beginning and end of the items in the list. This is automatic and doesn't need a <br> tag. The </ul> tag then closes the list.

In some cases, you may need to insert sub-items in some of your list items, such as in the following: Things To Do:

- Build Website
    - Create webpages
    - Check webpages
- Upload webpages
- Become Rich

You would then have to write the code this way: <ul> To-Do List:

```
<li>Build Website</li>
<ul><li>Create webpages</li>
<li>Check webpages</li></ul>
<li>Upload</li>
<li>Become Rich</li>
</ul>
```

Now, what you’ve done is to put a sub-list on one of the list items. Bear in mind that the browser can keep tabs of all of these, and output the correct format for so long as you do not forget to properly include closing tags.

Another popular feature of HTML is the numbered list. This feature allows you to create a list with automatic numbering, similar to how a word processor does it. The code is similar to that of an unordered list, but with minor differences. This is the tag to use: <ol>

This means Ordered List, an example of which follows: To-Do List:

1. Build Website

2. Upload

3. Become Rich

You need to write the following code in HTML:

```
<ol>To-Do List:
<li>Build Website</li>
<li>Upload</li>
<li>Become Rich</li>
</ol>
```

Similar to an unordered list, you can do nesting or even do a combination.

Headings

A lot of people do not realize that there are numerous pre-formatted headings in HTML as the <font> tag, among others, is frequently used to do the formatting. However, there are actually 6 pre-formatted headings, each characterized by a level. The sizes range from <h1> to <h6> with <h1> being the largest, and <h6> the smallest. Following are some examples: H1 Heading Sample

```
<h1>H1 Heading Sample</h1>
```

H2 Heading Sample

<h2>H2 Heading Sample</h2>

H3 Heading Sample

<h3>H3 Heading Sample</h3>

H4 Heading Sample

<h4>H4 Heading Sample</h4>

H5 Heading Sample

```
<h5>H5 Heading Sample</h5>
```

H6 Heading Sample

```
<h6>H6 Heading Sample</h6>
```

These headings, though not particularly great-looking, have 2 very important applications. First, the headers are structured based on the level of importance, and as such, intelligent browsers and software can tell which headers are more important than the others on the same page. Likewise, speech browsers that help sight-impaired people can take full advantage of this feature.

The second, and probably the more important application of the pre-formatted headings is that if you are using CSS, changing the format from one level to another is very easy, allowing you to format your headings any way you prefer.

This is great as it allows you to have structured headings that fit exactly to your website.

Base

Another one of the lesser-known tags, the base tag can be very useful in certain situations. Its two main attributes are href and target.

The attribute href is primarily used to specify the base URL for a particular page, and can be quite helpful in properly interpreting relative hyperlinks. To cite an example, you may want to have a link that points to:

myfolder/mywebpage.html

If you webpage can be found at

http://www.mywebsite.com/mywebpage.html

The proper URL to load each time the link is clicked should be

http://www.mywebsite.com/mfolder/mywebsite.html

You can, however, set the document base to http://othersite.com, and the link to load would then be:

http://myothersite.com/myfolder/mywebpage.html

This is applicable to images as well as other relative URLs provided to documents. To implement the <base>, you have to use:

```
<base href="http://myothersite.com">
```

Place the tag in your HTML script, specifically in the <head> section.

Though you may not immediately see the benefits, this tag can be extremely useful, especially if you need to include pre-made pages on a different server, or if your page can be accessed from a number of domain names. By using this tag, you need not edit all your links. All you have to do is to update the document’s <base>

On the other hand, the attribute target, will come in handy if you have frames on your website. When working with frames, you can write the hyperlink’s target frame as follows:

```
<a href="ourpage.html" target="contentsframe">
```

This would then load the file ourpage.html into the frame named contentsframe. However, if you want to open all the links in a specific frame, you can use the <base> tag in the following manner:

```
<base target="contentframe">
```

Just like with standard targets in HTML, you can also try:

_blank that opens in a new window

_parent that opens in the frameset parent
_self that opens in the current frame
_top that opens in the entire browser window without frames

Both of the base attributes – href and target – can be used in combo in the base tag.

Chapter 8: Basic Forms

Currently, interactivity is becoming more and more a priority for most websites. Though backend software such as ASP or PHP run a great majority of these systems, a front-end interface that the site visitors can use is still imperative. The process of transmitting information to a program or code on a website is done through HTML forms.

Basic Ideas about Forms

Simply put, an HTML form is similar to a paper form. In an online form, you are given a number of related sections or boxes in webpage where you need to enter some information. There are various methods to input data, the most common of which is to type your inputs. You may also be given the option to tick boxes or select items from a list.

The idea of dealing with forms in HTML is just like how you would deal with printed forms. You can find grouped or single items together in one page or form. HTML then instructs the browser where to send the gathered information once the form is filled up.

How to Define a Form

The first step in creating a form is to define it as a whole. To do this, you will need to use the following tags:

```
<form>
</form>
```

As you may know by now, the tags apply to everything found inside the opening and closing tags. In this particular case, anything found between the tags will be part of the form you are creating. However, bear in mind that you are not limited to a single form per page. You can add any number form tags that the page can accommodate. You can provide, for example a login and a signup form in one page. Just make sure that they are not nested.

The form tag is categorized as an invisible tag, which means it doesn't change the look of a page, except that there are browsers that leave a space after displaying a form.

On its own, the <form>tag is practically useless. You need to use any or all of the three primary attributes to make the form functional. These are action, method, and name

- ***Action Attribute***

  The action attribute is used this way:

  ```
  <form action="http://yourwebsite.com/cgi-bin/formmail.cgi">
  ```

  This code instructs the browser on where the entered data is to be sent. In the example above, it is http://yourwebsite.com/cgi-bin/formmail.cgi, which is tasked to handle the information. One great thing about a form tag is that the script the data is sent to may be located anywhere in cyberspace. This way, you are not constrained to use only those found in your site.

- ***Method Attribute***

Following are the two ways on how you can use the attribute:

```
<form method="get">
```

or

```
<form method="post">
```

GET and POST, as method attributes, pertain to HTTP’s standard methods of transmitting data on the net. GET places the information into the succeeding page’s URL; thus, it becomes visible in the address bar of the browser; for instance, http://theirsite.com/scripts/page.php?page=12&name=jim&agree=yes.

There is both a pro and con to this, however. The good thing is that the URL can be typed easily or linked to through GET, though this is not really critical when using forms. On the other hand, the bad thing is that anyone who sees your browser can view the data; it will likewise appear in the browser’s history. If you’re transmitting critical info from the form, you must not resort to this method.

POST, on the other hand, is a little different. Form data is transmitted through a special data system, instead of encoding it into the URL. This way, it’s not seen in the browser. It is therefore a more secure method than GET. It is still advisable, however, not to send sensitive data without any encryption. A form, by default, will make a submission via GET, unless the method is specified. Specifying the method, whichever you prefer, is highly recommended.

- ***Name Attribute***

  Typically, the form tag is used with the action and method attributes, although you can sometimes see it used with the attribute name. It is used this way:

  ```
  <form name="loginform">
  ```

  With this, the browser will recognize the form currently displayed on the webpage. This is quite useful, especially when you are coding, for example, in JavaScript for data validation.

To sum it up, you must use both the action and method attributes when you define a form. You must only use the name attribute when it is necessary.

Text Input

The most common method to input in a form, the text input looks something like this:

The text input allows users to type in text, and once the form is submitted, the data will be sent to the processing script. All most all of the input options available are don using the same basic tag form:

```
<input>
```

When combined with different attributes, a particular type of form input can be displayed. For the standard text box, for example, the following code is used:

```
<input type="text">
```

Now, this isn’t done yet as you need to use another important attribute to make the textbox work – Name. This attribute obviously allows you to provide a name for the textbox which you can later refer the data to. The name must not have any special characters or spaces. It is also important to note that each form item must bear a unique name. Here is the syntax to use for the name attribute:

```
<input type="text" name="username">
```

In the above example, the textbox is named “username.” Like the form tag and other HTML form tags, Name is also an invisible tag. It doesn’t have any practical use, except when you begin processing forms. When submitting forms using GET, remember that the name of the input will show up in the page’s URL. While it will likewise be sent with the data/information when using POST, it is invisible. For instance:

```
login.php?username=jim
```

Value is another attribute that you can use for the textbox which will allow you to set a default or initial value for the textbox that the user, in turn, can change. An example would be a textbox that is intended to collect an email address. If you want to set the textbox’s initial value to user@domain.com, write a code similar to the one below:

```
<input type="text" name="email" value="user@domain.com">
```

Other attributes are available for use with textboxes. However, since these are not really necessary when creating basic forms, we will cover them in the succeeding chapters.

Other HTML in Forms

Just to put things in perspective, though it should be obvious to you, you can include any other tag in your <form>tags, but it is important to make sure that all your input elements are labeled. This way, users will know that type of information to enter. Other HTML formatting types can be used as well. In particular, using tables can help produce better looking HTML forms.

Buttons

After the user has entered some information into a particular form, you need to have a way to proceed to the succeeding page. Buttons can be used for that purpose. Two basic button options are available – submit and reset. The submit button is more commonly used option. When this is clicked, the browser is told to transmit the date to the pre-defined URL, using the supplied method. The code below will add the most basic form of submit button:

```
<input type="Submit">
```

The submit button comes with an optional attribute that lets you do a bit of customization – the value attribute. This is utilized in exactly the same manner as in a textbox, though when used with buttons, it merely changes the text on the face of the button. Here’s the syntax:

```
<input type="Submit" value="Sign Up">
```

You can include any number of submit buttons you want in one form, but all buttons in the same form must perform the same task. An example would be “Yes” or “I Agree” buttons you find in many websites.

The reset button is another type of button that you can add to a form. When clicked, all it resets the form back to its original state. Typically, all values are cleared. However, it will restore the set initial values in cases where value attributes are set for input options. The syntax is as follows:

```
<input type="Reset">
```

Similar to the submit button, the reset button text can likewise be changed when the value attribute is used.

Chapter Summary

In this chapter, you learned the basic form format in HTML, including its structure, how to reset and submit the form, as well as how you can get basic inputs. In the next chapter, we will cover other available input options.

Chapter 9: More About Forms

This chapter will be a continuation of our discussion on forms. There are many other elements that you can include in HTML forms that were not discussed in the previous chapter. We will cover some of them in the succeeding pages.

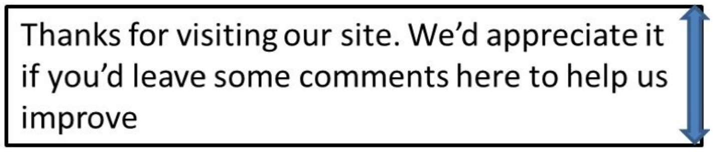
Larger Text Inputs

Textboxes, while they provide a functional way to get text inputs from users, are very limiting. For one, users can only enter a line with a limited capacity. If you need users to input more than just a phrase or a few words, you can instead use textarea (multi-line textbox) which is more flexible. Here is the correct syntax to use:

```
<textarea name="comment">
</textarea>
```

Like in other form elements, a name must be specified for a textarea (“comment” in the example above). Note that unlike the button and standard text box, a closing tag is used in this element. This makes it possible for the webmaster to set a bigger amount of initial text. Thus, users can input more text compared to other form elements. Anything enclosed in the tags will be included in the textbox.

Unlike standard HTML, the new lines included in the code between the tags will be displayed on the page as new lines. For example, if you want users to input their comments, you can make a textbox like the one below:



Thanks for visiting our site. We’d appreciate it if you’d leave some comments here to help us improve

Here’s the code to generate the textbox above:

```
<textarea name="comments">Thanks for visiting our site. We’d appreciate it if you’d leave some comments here to help us improve.</textarea>
```

Functional features of textboxes include scrollbars that will be displayed if the text input does not all fit in the box, as well as automatic text wrapping. If you’re expecting large data input, you need something bigger than the default box. Using the rows and cols attributes, you can achieve this in HTML. The attributes let you set the width or number of columns your box should have as well as the number of rows to appear. Instead of the previous example, you can use this code instead:

```
<textarea name="comments" rows="5" cols="30">Thanks for visiting our site. We’d appreciate it if you’d leave some comments in this box to help us improve.
</textarea>
```

Radio Buttons

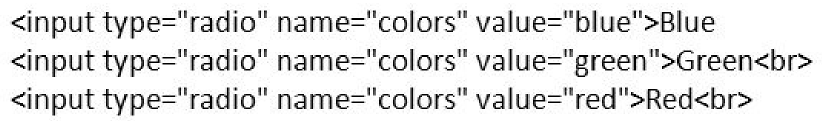
The radio button is a form element that is often used to offer choices, such as in a voting form. The buttons are defined as a group, and only a single button can be selected at a time. Following is an example:

o Blue

- Green

- Red

You can achieve the above example through the following code:



```
<input type="radio" name="colors" value="blue">Blue
<input type="radio" name="colors" value="green">Green<br>
<input type="radio" name="colors" value="red">Red<br>
```

Standard <input> tags are used when inserting radio buttons. The type is set to radio. However, unlike other form elements, naming for radio buttons is not by individual element but by grouping.

When working with textboxes in single form, you may name one textbox box1, and another box2. For radio buttons to work properly, however, all buttons belonging to the same group must bear the same name. The form will then provide the name of the button group when submitting the data, followed by the selected button’s set value, which is the output.

To add multiple sets of radio buttons in a single form, you just give a different name to the other sets. Note that the buttons’ location on the webpage has no bearing to their groupings. They will be grouped according to their name in the same form.

<u>Checkboxes</u>

Checkboxes, unlike radio buttons, are either off or on. Any number of checkboxes can be selected, and there are no groupings. To insert a checkbox, the standard <input> tag is used with both value and name. for example:

```
<input type="checkbox" name="disagree" value="no">
```

When submitting a form with a checkbox, the browser only sends the name and value of the textbox if selected; otherwise the checkbox will be ignored. In some cases, having the checkbox pre-selected can be useful. For this purpose, simply adding a checked attribute to the tag will do, like in the example that follows:

```
<input type="checkbox" name="disagree" value="no" checked>
```

This way, the user is in full control as he can opt to check and uncheck the items even if the boxes are pre-set as checked. In most cases, however, the user will just leave them checked.

Selection Boxes

A selection box lets you provide an options list from which the user can select one. Useful applications include choices for the country of origin, and other cases where you do not want to give the user the freedom to input text, just like the sample below:

Here’s the basic code for declaring a selection box:

```
<select>
</select>
```

You can use the name attribute for this tag, so you can later access the data. Add the options between the opening and closing select tags, in the exact order that you want them to appear in the list. You can add as many list items as you want, and the box size will adjusted by the browser accordingly.

```
<option value="1st">Item 1</option>
```

It doesn’t matter what value you put in. Unlike in a textbox where the value must be related to the option, you can put anything, even if it is not sequential. Just put everything you want to be displayed in the list in-between the opening and closing option tags. The length does not matter as long as it is reasonable; the selection box will resize automatically to accommodate the longest item.

The default display is the first option if no choice is made. in most instances, this is fine. Some boxes will contain a message like “Please select here”, then assign a value that will later be deemed as invalid. Sometimes, in country-selection boxes, for example, you may want to have another default option. You just have to include the “selected” attribute to do this:

```
<option value="USA" selected>United States</option>
```

Hidden Fields

The last element that you can include in your form is a hidden field. Initially, it may seem that this element is useless. What it does is to send form data with a name and value. The user cannot change the value as nothing will appear on the screen. Although it can be read in the source code, nothing will show up on the page. Thus, it must not be used to store information that the user must not see.

However, hidden fields are very useful for many server-side systems that generate HTML frequently. A good example is an ordering system where users can input data in multiple pages. The data is usually put into a database temporarily, with the key to access it stored in a hidden field on the webpage for use in the succeeding stage. This will enable the backend system to know which specific user is expected to submit data from the next page.

You can add a hidden field using the code below:

```
<input type="hidden" name="usercode" value="139494037720">
```

Obviously, hidden fields do not require any extra attributes as no appearance settings require changing for the element. You can include any number of hidden fields. Just make sure that each one bears a different name. Likewise, you can place a hidden field anywhere in the form tag.

Chapter Summary

You have learned the things you need to do to put a form on a page, and how you can include several advanced elements.

Chapter 10: Common HTML Mistakes to Avoid

Although beginners are expected to make mistakes from lack of experience, it is sad that even supposedly jaded HTML programmers sometimes commit costly mistakes in coding primarily because of carelessness. A clean code is important as it will serve you in good stead as a front-end developer, not to mention that it will save you a lot of time you would otherwise spend for debugging and editing.

If you're a skilled programmer, it always helps to review your work as a lot of errors are caused by haste and failing to practice sound coding habits right from the start. Following is a list of the common missteps and mistakes that both new and experienced coders alike often encounter.

1. ***Placing Block Elements in Inside Inline Elements By default, HTML displays an element as inline or as block. Block elements like paragraphs and divs, comprise the document's structure. On the other hand, inline elements dwell inside blocks like span and anchor tags. It is therefore a big mistake to place a block inside an inline element. To illustrate what is sound coding and what is not: Right: <h2><a href="#">Inline element inside Block element</a></h2> Wrong: <a href="#"><h2>Block element inside Inline element</h2></a> 2. Failure to Use The ALT Attribute When Working with Image Tags When using IMG tags, remember that the ALT attribute is an important requisite as it describes the image's context. It allows a user with a slow internet connection or one who is relying on a screen reader to decide for as to whether a particular image is important or not. Likewise, it allows web crawlers to index the contents of your website in a more efficient way. You should use the attribute even if your images serve no crucial purposes and are just for show. Just include an empty one such as ALT ="".***

Right: <img src="userprofilepic.jpeg" alt="User's Profile Picture"/> Wrong: <img src="userprofilepic.jpeg"/> ***3. Using Line Breaks when Showing a List*** If you intend to present a numbered or bulleted list, you should not use any line breaks. Instead, you are better off using ordered list <ol> or unordered list <ul> tags.

Right:

```
<ol>
<li>Larry Bird</li>
<li>Magic Johnson</li>
<li>Michael Jordan</li>
</ol>
```

Wrong:

```
1. Larry Bird<br/>
2. Magic Johnson<br/>
3. Michael Jordan
```

4. Using the <b> and <i> Tags to Bold and Italicize Text Yes, it's true that <i> and <b> are used for italicizing and bolding text, but they're categorized as presentational tags in terms of semantics. You can use the font-style and font-weight properties of CSS. If you really need to apply the styles to your page, use <em> and <strong> instead as they perform the same task, but are correct – semantically speaking.

Right: <strong>This is bold and correct!</strong> Wrong: <b>This is bold but faulty!</b> ***5. Using Multiple Line Breaks*** You should not use the line break tag to create a gap between elements. Split the text into individual paragraphs instead.

Right:

```
<p>Here’s a line</p>
<p>Here’s another line</p>
```
Wrong:

```
Here’s a line
<br/>
<br/>
Here’s another line.
```

6. ***Using Inline Styles This may not be the first time you ever heard about this, but the whole idea behind semantic HTML and CSS is to create a distinction between styling and structure. Thus, it does not make any sense to directly put styling into the HTML code.***

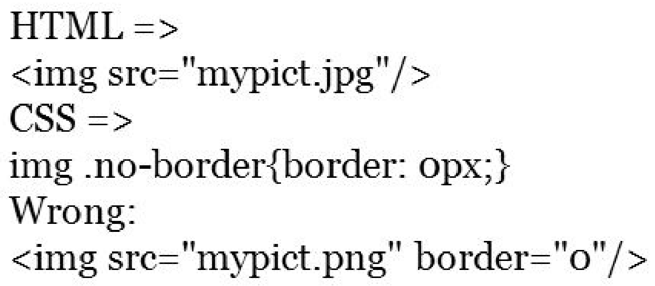
Right:

```
HTML =>
<h2>Correct</h2>
CSS =>
h2 .blue{color: blue;}
```

Wrong:

```
<h2 style="color: blue;">Wrong</h2>
```
7. Adding or Removing the HTML border attribute Instead of modifying the border attribute in HTML, it should be semantically done in CSS as it is presentational.

Right:



```
HTML =>
<img src="mypict.jpg"/>
CSS =>
img .no-border{border: 0px;}
Wrong:
<img src="mypict.png" border="0"/>
```

8. Using <marquee> or <blink> These tags are not part of the official HTML standards. In addition, they are deemed as unimpressive and not aesthetically pleasing. If there is a need to put a spotlight on a particular part of your page, opt for an approach that is not as offensive: Right: Do not use them altogether Wrong: <marquee>Come and see this!</marquee> 9. Utilizing Deprecated Elements Although at present, some modern browsers support a number of old, deprecated HTML attributes and tags, they may no longer do in the future. Avoid using.

10.Forgetting to indicate the DOCTYPE

Doctype indicates the type of HTML you're using. If it is not described there, there is no way of knowing whether your code is valid or not. In addition, your browser will assume for you, which may not result according to what you intended. Usually, it is ignored since it is difficult to recall the long address. To avoid the problem, use a blank template that can help you avoid remembering it, and you will always have it on hand.

Chapter 11: HTML FAQs and Their Answers

Following are some of the most frequently asked questions about HTML and their answers: Q: Is it really necessary to use </TD> and </TR> to end a table? I tried creating a table without using them, and everything went fine. So, what happened?

A: If there's only 1 table on your page, there's no risk of interaction issues. However, if there are images in the table cells, you may encounter some problems. The same is true if there are multiple tables on your page. So, use them.

Q. Should I always use new lines for new commands?

A. The browser doesn't mind, so why should you? You can even write your codes in a single straight line, and the page would look the same in your browser. It is, however, recommended that you do it as it will make your document look more organized.

Q. How can I make text blink?

A. Use

```
<BLINK>text text text</BLINK>
```

Q. Should I capitalize HTML commands?

A. No, the browser doesn’t really care. However, you can do it if you think it can help you quickly find things in your document.

Q. How do I put password protection for WWW pages?

A. There are actually two ways

1. You can do it using JavaScript
2. Have the entire server directory set as password-protected

Q. How can I limit the number of characters in my text box?

A. Insert the command MAXLENGTH="-" inside the textbox command.

Q. What is the difference, if any, between using the word color codes and the hex color codes?

A. It will matter if you're using an older level, antiquated browser such as ver 1.0 or 1.1 as these are not capable of recognizing word color commands. Aside from that, there is practically no difference.

Q. Is it necessary to put the # mark in front of a color hex code?

A. No, it is not part of the current browser requirements. There are, however, some earlier level browsers still being used today and it would be to your advantage if they can also read your webpages with few to no errors at all. To be safe, just include it.

Q. How can I remove the border surrounding an image when I create a link?

A. In the image command just insert the BORDER="0" command in-between the IMG and the SRC.

Q. My entire page turned into a giant link! It is underlined and all blue! What happened?

A. It certainly does not look good. However, it is quite easy to fix. What happened is you may have incorrectly written or forgotten one end anchor link.

Just go to where the blue starts. Chances are, it is a ling – one that is missing the </A>

Q. My frame commands seem to be ok but it doesn’t work. Nothing appears on my page.

A. If they are correct, then you may have failed to put in sufficient commands. Some browsers are a bit more forgiving on missing end tags and your code may work on them. However, to make sure that your code will work on all browsers that are frame-ready, make sure that all frame tags are properly matched, meaning there is an opening and a closing tag.

Q. Why can’t I seem to get my images to display on the page?

A. There are three possibilities:

1. The image may be corrupted. Does the image open in the browser on its own? If so, then it transferred fine. That is not the problem.

2. Is the image name exactly the same as what you call for? It is case-sensitive so check if you need to use capital letters.

3. Do you see the C:\ in the command line? Get rid of it. It is a link to a location in your computer, and not the server.

Q. All the commands work except for the hypertext link. What can be wrong?

A. Check your code for a missing quotation mark. A dot may also be missing. Make sure that the address is right. Let’s put it this way, assuming all the commands are correct, then there is no reason why your code won’t work. Otherwise, there is definitely a problem – somewhere. Don’t worry. You will find it. Just review your work.

Q. When I view my page in the browser, it appears with strange symbols and characters, and several words. What does all that mean?

A. Do you see box-like characters and other similar things? I am quite sure that you missed something – the </TITLE> command. So, just go back to your HTML document and fix it. I’m sure the strange objects will be gone next time you view your page.

Q. Why is it that when I try to view my page on my browser, what I see is my HTML document with all the commands appearing?

A. Most probably, your file was saved as a .txt file, and not with the .html extension.

Q. How do I get the < and the > to show up on the page?

A. You need to include the symbols in an & command. This code: &lt; stands for <. On the other hand, the &gt; stands for >. Never use these commands together with < and >. Put them on the page similar to how it is suggested here, and it will surely work.

Q. Can I use an image as a button for sending a form?

A. Definitely. Once you have decided on the image, use this code:

A. Yes. Once you have an image, use this as your send button code:

```
<INPUT TYPE="image" SRC="imgbutton.gif" BORDER="0">
```

For so long as you make sure the code is inside the opening and closing FORM tags, it will perform the mailto: activity of the form.

Q. Why is it that sometimes, I get the message that the page does not exist when I know that it should be there?

A. It can be because of net congestions. If there are many people trying to attach or download, it may cause the server to reach its maximum limit, and it may be trying to tell you that you cannot connect. However, servers don’t use a lot of error codes. A single code can be used for various errors which can be quite confusing, especially for those who are new to these errors. If you get the message “The server is full” you may be able to come in after several seconds.

However, you can never really be sure with those kinds of error messages.

Q. How can I get my images and pages to the server from my computer hard drive?

A. You can do that via a file transfer protocol (FTP). It would be too lengthy to discuss how it works here so you can just look it up. There are other ways to send files to your web server. Again, you can check online for these methods or contact your web host and ask for assistances. In most cases, they would be more than willing to help.

Q. All my pages that contain images work just fine when accessed from my computer’s hard drive. However, when I transfer everything to my web server, the images do not display. What can be wrong?

A. Your images may have been corrupted in the process of transferring your files to the web server. To avoid this from happening in the future, try sending your image files in raw data or binary format. You can also request for help from your web server provider. There are lots of ways to transfer your files and your server technician should be able to help you out.

Q. I see an image with a square, triangle, and circle on it. What does it mean?

A. It simply means that it is a space reserved for an image that is still being loaded.

Q. What does the image of a square with a “?” (question mark) inside mean?

A. The icon means that an image is being called, but it cannot be found.

35. **Q.** What does the square with the question mark inside it mean?

A. That's an icon meaning that the image that is being called for is nowhere to be found.

Q. How can I remove the horizontal scrollbar that appears at the bottom part of my webpage?

A. The images you included in your page may be too big for the browser to display. Try to replace them with smaller images. The scrollbar should be gone once you do that.

Conclusion

Thank you again for downloading this book!

I hope this book was able to help you to learn how to write programs in HTML, the language of the Internet.

The next step is to apply what you have learned from this book to create fully functional and aesthetically pleasing webpages.

Finally, if you enjoyed this book, then I'd like to ask you for a favor, would you be kind enough to leave a review for this book on Amazon? It'd be greatly appreciated!

Click here to leave a review for this book on Amazon!

Thank you and good luck!

BONUS : FREE HTML and CSS Course!

I know how important it is to implement what you learn, or even learn by taking action. As my “Thank You” for downloading this book I provide you with FREE HTML course where more than 4.5M students are enrolled! With this course you will learn and use more of HTML programming language. You will learn how to create websites by structuring and styling your pages with HTML and CSS

So what are you waiting for? Click here to get started now!

Or go to: http://bit.ly/1FhfSQa

We want you to succeed in your goal to mastering HTML and CSS! Please make use of this course, it will help you a lot!