Lesson 1 - Introduction

Welcome to the first of seven basic HTML (**H**yper**T**ext **M**arkup **L**anguage) tutorials. By the time you reach the last basic tutorial, you will be able to make a web page that is customized to your own needs. It's not that difficult! Hundreds of thousands do it. Let's get on with the show.

What is HTML?

It's what enriches plain-text information to be the colorful, interactive web pages that you see all across the web. Without HTML the term "surfing the web" would mean "digging through mounds of text".

Hyper	Text	Markup	Language
No longer does the reader have to read things in order. This enables them to skip around berween sections. This "loss of order" is what makes HTML hyper.	This is what you will use to convey your ideas to the reader. It's good ol' English letters.	In a web page, HTML marks up the text. This is what HTML is all about.	No, you can't speak it like English (well, you probably could, but it might get ugly). It's the language used to create web pages.

What isn't HTML?

It's not a computer language. This can't be emphasized enough. If someone asks you what computer languages you know, don't answer "HTML." You will feel really, really stupid. Also, HTML isn't a difficult, cryptic computer language. It is a set of codes that enables rich types of content to be distributed across the World Wide Web.

Why not use a program to make web pages?

With all those programs that let you create web pages without learning HTML, why learn the language anyway? Here are three important reasons why HTML is the best way to go. These such programs are:

- **Inflexible.** These programs rarely use the new updates to HTML that are added so frequently. Therefore, they do not use all of the technology that is available to HTML coders.
- **Costly.** The price of these programs can be up to a few hundred dollars (USD). How much does it cost to code HTML? Nothing.
- **Inefficient.** Most of these programs contribute significant amounts of extra code does not serve any positive purpose. What does this extra code mean? Your web pages could load much slower for your readers.

What you will need

1. A web browser. Anything from Microsoft Internet Explorer and Netscape's Navigator to NCSA Mosaic will be fine. If you're looking at this page, then you obviously have one of these.
2. An ASCII text editor. Don't freak out if you're confused! This includes Windows Notepad, Macintosh SimpleText, or a word processor.

Q & A

Q. *Do I have to be connected to the internet to write HTML?*
A. No. All of the web pages you create will be made while you are offline.

Q. *Can I write HTML even though I have a Mac (or IBM)?*
A. You bet. HTML can be written on any platform because it is just plain text.

Good job! You finished the first tutorial!

Go to **Lesson 2**

Lesson 2 - Working with HTML Documents

Let's get down to business. In this tutorial you will learn how to use a text editor, which is what you will use to write HTML documents. Please read the directions that correspond to your operating system:

- **Windows 95/98/NT** The text editor for Windows is Notepad. You can run it by clicking the *Start* menu, highlighting *Programs*, highlighting *Accessories*, and clicking on *Notepad*.
- **Macintosh**. You will use SimpleText as your text editor. I am told that SimpleText wanders around most Macintosh systems. If you can't locate SimpleText, select *Find File...* from the apple menu and search for "SimpleText".

When you are finished with your HTML document you will want to save it. Just select *Save As...* from the *File* menu. Save the file wherever you like. I suggest making a directory to put all of your web pages or you can easily store them on a floppy disk.

How to name your HTML document
Some disagree, but you should name your HTML documents with the .html extension whenever possible. An HTML document doesn't deserve to have it's extension truncated to *.htm* unless it's absolutely necessary. If your operating system is...

Windows 95/98/NT or Macintosh
Save your files with the *.html* extension. Name your files whatever you want **(avoid the space, quotations, and any other weird characters)** and add the suffix *.html*. Here are some legal examples of filenames: *dogs.html, myfirstpage.html, htmlrules.html*. **I would suggest you stay away from capital letters to reduce confusion.**

Why should I save it with the .html or .htm extension?
All this extension mumbo-jumbo is based on a thing called file association. When you add the .html or .htm suffix to your document, your computer knows it's an HTML document. If you learn about graphics, movies, or sound, then you will see other, different file extensions.

Opening the HTML document into your browser
When you've got an HTML document saved onto your hard drive or floppy disk, follow these instructions to see how it looks in a browser.

1. Go to the *File* menu and select *Open*, *Open File with Browser*, *Open Page*, or anything very close.
2. Some browsers let you select the file right away, but on others you will have to click a *Browse...* or *Choose File...* button. When you get the dialogue box, find your HTML document (whatever you named it) and click the *Open* button. You may have to *OK* or *Open* out of another window of a newer browser.
3. The browser will do the rest and hopefully renders your HTML document so it appears as a web page.

Looking at other people's HTML
I highly recommend that you look at other people's HTML as you surf the web. I'm not talking about the pretty web page rendered by the browser, but the actual HTML source code used to make the page. "Why?", you say? It'll help you understand HTML. If you can read through someone else's HTML and can see what's going on then you know what level you're at. Also, if you see a good idea on someone's web page, you're going to want to know how to do it. I don't mean for you to steal their work, but it'll give you an clue on how that idea works. Here's how you do it:

1. Find a page that you like.
2. Click on the *View* menu, and select *Source*, *Page Source*, *Document Source*, or whatever is closest. (If this doesn't work, try right-clicking in the browser window and selecting *View Source*, or words to that effect.)

3. Another window will pop up with the web page's HTML.

Get ready, for next lesson you write your first web page.

Go to Lesson 3 - Tags

Lesson 3 - Tags

Now that you've played with your text editor and tried saving as *.html* or *.htm*, we're ready to get down to writing. In this lesson you will create you first web page.

What is a tag?
HTML is written in plain english text. It's simple, reads top-to-bottom, and reads left-to-right, but it isn't just regular english text. What makes it different? Tags. You will use tags to set text apart to be big, small, bold, underlined, italicized, and many more.

Tags are much like light switches in the way they act. If you want a sentence to be bold, you place the bold tag at the exact spot where you want the bold lettering to start (like turning on a light switch) and where you want it to stop (like turning off a light switch).

Tag Format
All tags begin with a less-then sign (<) and end with the greater-than sign (>). This will be the format for every future HTML tag. Between the < > is a code which depends on what you want to do. Learning HTML is about learning codes to perform what you want to happen. Here's an example. Let's say we want to write the word "Dog" in bold. The HTML code for bold is B.

<B>Dog</B>

Take a look at what is happening...

<B>	Dog	</B>
This is the beginning bold flag. It, like a light switch, turns on the bold and is called the opening tag.	With the bold tag preceding it, this text is turned bold.	This turns off the bold and is called the closing tag. If there were more text after this closing tag, it would be just normal text.
This entire thing is called an element. It consists of an opening tag, the affected text, and a closing tag.		

More elements for you to use
Here is a table of the B element (used in the example above) and three new ones that can be used just like the bold element:

Code	Effect	Tag in Use	How it looks
B	Bold	<B>Bold</B>	**Bold**
TT	Typewriter	<TT>Typewriter</TT>	`Typewriter`
I	Italics	<I>Italics</I>	*Italics*
U	Underlined	<U>Underlined</U>	Underlined

Q & A

Q. *Do all elements have both an opening and closing tag?*
A. No. Some just have one tag, but you'll learn more about them soon.

Q. *Is the end tag just the start tag with the added slash?*
A. Exactly.

Q. *Will tags show up on the web page?*
A. Nope. Anything between the < > will be hidden from the viewer.

Q. *Should I capitalize tags like you did?*
A. Although browsers don't care, it's a good idea. It helps keep tags separated from text and it's a good habit to start.

Q. *Will text show up on the page even if it doesn't have a tag?*
A. Yes, if you just type normal text into your HTML document, it'll show up like normal text on the web page.

Q. *Can I use two tags at once?*
A. Yeah, but you must remember to close both tags also. This is how it's done...

<B><U>Bold and Underlined</U></B> becomes **Bold and Underlined**
<I><B>Italicized and Bold</B></I> becomes ***Italicized and Bold***
<I><U>Italicized and Underlined</U></I> becomes *Italicized and Bold*

Elements with just one tag

Unlike the four elements you have learned already, these elements have only a single tag. As you will see, they do not need a second tag...

Tag	What It Does
<HR>	This element puts a line across your page. It stands for Horizontal Reference. --- The line directly above this text was created using <HR>
 	This stops text on the current line and begins again the next line. A BReak, in other words. You will probably need this often, for when HTML is saved as text it does not save where you hit the *ENTER* key. This is how it works... **My favorite animal** **is a cat.** The line break above where "My favorite animal" is broken from "is a cat." was caused by .
<P>	This is the equivalent of two tags. While just skips to the next line, <P> skips a line then starts again. The whitespace you see just above this line of text was created from a <P> tag.

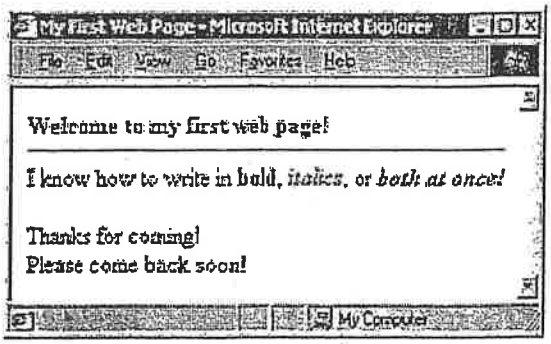
Web page format
Ready? You are about to create your first whole web page. We will be using elements you've learned already **plus** three more that you will learn below. You will see the following three elements on every page you make from now on.

- **All your pages will being with: <HTML>**
  This tells the web browser that it is an HTML document.
- **The next element is: <TITLE> </TITLE>**
  Between these two tags you will be the window title. You can find the window title by looking at the very top of this page. It's the text on a short, wide (probably blue) strip.
- **Here will be: <BODY> </BODY>**
  Between those two tags you will all of the content of the page. This is just about everything except the title (which you put above in the <TITLE> element). It includes text, <HR>, <BR>, <P>, and much more.
- **Lastly, you will end every page with: </HTML>**
  You start and end every page with the HTML tags. This lets the browser know where the end of the web page is.

Writing your first page
Now that you know the structure of a web page, I'll show you a sample one. Here is the source code:

```
<HTML>
<TITLE>My First Web Page</TITLE>

<BODY>

<B>Welcome to my first web page!</B>

<HR>

I know how to write in <B>bold</B>, <I>italics</I>, or <B><I>both at
once!</I></B>

<P>

Thanks for coming!<BR>

Please come back soon!

</BODY>
</HTML>
```

On the right you will find a sized-down browser window with the web page listed above. This is how it looks in Microsoft Internet Explorer.



Go to Lesson 4 - Attributes

Lesson 4 - Attributes

If it weren't for attributes, HTML would be extremely boring. Why? You'll see...

What is an attribute?
Let's look at the <HR> element we learned in the previous lesson with an added attribute:

<HR SIZE="6">

this is how it looks in a web browser:

Wow! It's different! Remember that the <HR> element without an attribute looks like the following horizontal line:

So what happened? The added attribute, **SIZE="6"**, modifies the element it's in, **<HR>**. This is the reason that the first horizontal line has a height of six. If you wanted to you could change the value of the **SIZE** attribute to something other than **6**. Let's say we wanted a horizontal line with a height of 10. You would use: **<HR SIZE="10">**. It would appear like this in the browser...

Q & A

Q. *Do all attributes follow the same format?*
A. Yes. The format is: **ATTRIBUTE="value"**. It's just like: **SIZE="6"**. Attributes are always part of a tag, just like **SIZE="6"** is part of the **<HR>** tag.

Q. *Does the attribute have to be in all uppercase like you have above?*
A. No, it's like tags. The browser doesn't care whether they're uppercase or lowercase, but it's easier to separate attributes if they're in capitals.

Q. *How do I know what an element's attributes are?*
A. That's what HTML references are so useful for.

Changing the size of text
To change font sizes you use the **<FONT>** tag with the **SIZE** element. There are twelve (12) sizes for you to use: **-6 through -1, and +1 through +6**. Minus 6 is the smallest while +6 is the largest. Here's how they look on the page:

<FONT SIZE="-3">Size -3</FONT>

<FONT SIZE="-1">Size -1</FONT>

<FONT SIZE="+1">Size +1</FONT>

<FONT SIZE="+3">Size +3</FONT>

See what's going on? The **SIZE** attribute modifies the **FONT** tag to change the text to the size you choose. Notice that the closing tag for an attributed element need not the actual attribute. For example, to close a **<FONT SIZE="+1">** tag, all you need to do is put **</FONT>** (not **</FONT SIZE>** or **</FONT SIZE="+1">**).

Centering Text
There are two ways to center text. They are:

<CENTER>This text is centered with the CENTER element </CENTER>

<DIV ALIGN="center">This text is centered with the DIV element and the ALIGN attribute</DIV>

I always use the **<CENTER>** element to center text. I just wanted to show you the latter way to give you a little more practice with attributes. The **ALIGN** attribute tells **<DIV>** that it should align to the center.

Aligning text to the right
Here is how you can align text to the right side:

<DIV ALIGN="right">This is on the right</DIV>

Notice this **<DIV>** element can also be used to center text, as shown in the "Centering Text" section above. What enables you to align text to the right is the **ALIGN** attribute. When you set it to equal "right", then it goes to the right. If it's set to equal "center", it goes to the center. Simple, huh?

Now you have a very solid understanding of the format of HTML. From here on it'll be applying knowledge you have already learned. In other words "It's all downhill from here."

Go to Lesson 5 - Making Links to Pages

Lesson 5 - Making Links

In this lesson you will learn how to link to other pages and your email address. These links are what make the **H**yper**T**ext **M**arkup **L**anguage so hyper.

Linking to other web pages

This is how you would link to the main Pine Creek page:

<A HREF="http://academy.d20.co.edu/pchs/index.html">Go to Pine Creek</A>

Let's take a look at what's going on...

A	HREF	http://academy.d20.co.edu/pchs/index.html	Go to Pine Creek	/A
This stands for Anchor. It is the element name for a link.	This stands for HyperText Reference. This is where the link will go when it's clicked.	This is the full address that the viewer will be sent to if the link is clicked.	What you want the user to click is what this is. This is what must be clicked for them to be sent to the link's destination.	The link element is ended.

Here's what the link looks like in a web browser...

Go to Pine Creek

Creating a link so people can e-mail you

This will create a link to your e-mail address so anyone can e-mail you. Here's the format:

<A HREF="mailto:pchsweb@d20.co.edu">Click Here to E-mail Me</A>

It's pretty similar to a link that goes to another page. The main difference is the full address is in a different format. You must use the "mailto:email-address" format to make it work. Here's what it looks like in a web browser...

Click Here to E-mail Me

Try clicking on it. It should pull up a window where you can easily e-mail me. Please do if you wish.

In the next lesson you get to learn how to put pictures on your web page. Ready?

Go to **Lesson 6 - Adding Images**

Lesson 6 - Adding Images

What makes HTML really fun is putting pictures on your web page. But how do you get those beautiful, intricate Images into a simple HTML document? You don't. Images are separate files you include into your web page.

Adding an image to your page
Here's the format...

<IMG SRC="image.gif">

This is what it looks like through the web browser after replacing *image.gif* with *"your image".gif*

IMG	SRC	image.gif
This stands for "image" and tells the browser that an image will go here.	This attribute stands for "source" and it tells the browser where to find the image.	This is the value of the SRC attribute and it tells the browser where the image is. The *.gif* extension tells the browser what type of image it is .

Most browsers can only open .gif or .jpg images, so it is important to make sure those are the extensions of the images that you are going to use.

Linking an Image
Now you know how to make a text link. You know how to place an image on your web page. Let's combine the two. We're going to make an image clickable, just like a link, so the viewer to click on it to get to another page. Here's the HTML for it:

<A HREF="http://academy.d20.co.edu/pchs/index.html"><IMG SRC="pc.gif"></A>

It's just like a link, except you put an image instead of the text. Now, instead of clicking on text to get to *http://academy.d20.co.edu/pchs/index.html*, you click on an image. This is what it looks like in a web browser...

If you click on it, it will send you to the main page of Pine Creek. But what's that colored border around the image? That's what happens when you link an image. Personally, I think it's ugly. To get rid of it you must add a second, **BORDER** attribute to your image. Now the HTML for the image is...

<IMG BORDER="0" SRC="pc.gif">

And now it looks like...

Lesson 7 - Aligning Images

Unlike text, aligning images doesn't require another element. All you must do is add another attribute. Ready? Let's go...

Horizontal alignment
By default an image is aligned to the left. You can use **<CENTER> </CENTER>** (Lesson 4) to center an image. To right-align an image you can use **<DIV ALIGN="right"> </DIV>** or you can do this...

<IMG SRC="image.gif" ALIGN="right">

This is how it is rendered through a web browser using *creation.gif*...

Aligning text with images
You can align text next to images at the top, middle, or bottom with these examples...

<IMG SRC="creation.gif" ALIGN="top">Top of the image.

<IMG SRC="creation.gif" ALIGN="middle">Middle of the image.

<IMG SRC="creation.gif" ALIGN="bottom">Bottom of the image.

And here is how they look through a browser...


Top of the image (**ALIGN="top"**)


Middle of the image (**ALIGN="middle"**)


Bottom of the image (**ALIGN="bottom"**)

Now that you know so many things in HTML you can easily make a nice, organized web page. There is still more that you can do, but the groundwork is set out.

See? Much better. We set the **BORDER** attribute to "0" so it would not have a border. There are two attributes now in the **<IMG>** tag. Does it matter what order the attributes go in? No. the **BORDER** attribute could be after the **SRC** attribute and it would still look the same. Now, when you click on the image above it still takes you to the main page of Pine Creek, but there is no colored border around it.

Now you can add beautiful, intricate images to any of your web pages. In the next lesson you will learn about placing images on your page by changing their alignment.

Go to **Lesson 7 - Aligning Images**