

# Intro to HTML

By Bryan Cairns

## How To use this help file

When using this help file, you can easily cut and paste the scripts from this help file to your text editor (HTML Helper), but be aware that do not change any thing in **red**. But you will want to change the text that is **green**.

## Introduction

This is a tutorial for the Hyper Text Markup Language known as HTML. The HTML programming language is widely used for the construction of internet "web pages".

Before we dive right in, first let us go over some basic terms.  
(If you already know HTML, you will find this section redundant.)

**HTML** - (Hyper Text Markup Language) Used to make web pages. For this tutorial we will be using Microsoft "Note Pad" which comes standard on all computers with Windows 95. But any text editor will work just fine.

**FTP** - (File Transfer Protocol) Used to transfer files to your pages to the internet. Most internet Browsers come FTP compatible, which means that you can upload (send) files to your directory on the internet. If your browser is not FTP compatible, you will need to get an FTP program.


**GIF** - An animated image used on the internet

**JPG** - A compressed image used on the internet

**Link** - A "hot spot" on the web page that loads another page, sends mail, or performs an action. Usually they look like [this](#), but they can also be pictures, that once you click on them, a page loads.



## Active Channel

Add Active Channel 

Active Channels are something that can keep visitors coming back to your site. An Active channel can inform your viewers of updates to your site, and automatically download materials on to their computers for offline browsing. But Active Channels do have one fatal flaw; they can only be used by the IE4 web browser.

Below is a standard CDF file (Active Channel) generated by HTML Helper:

```
<?XML Version="1.0" Encoding="iso-8859-1" ?>
<Channel HREF="HTTP://www.domain.com/channel.html" BASE="HTTP://www.domain.com">
<Title>Title of Channel</Title>
<Abstract>Description of Channel</Abstract>
<Logo HREF="HTTP://www.domain.com/image1.gif" Style="IMAGE" />
<Logo HREF="HTTP://www.domain.com/image2.gif" Style="ICON" />
<Logo HREF="HTTP://www.domain.com/image3.gif" Style="IMAGE-WIDE" />
<LogTarget HREF="channel.html" Method="Post" Scope="Offline">
</LogTarget>
<ITEM HREF="HTTP://www.domain.com/channel.html" PRECACHE="yes">
<Schedule>
<IntervalTime DAY="1" />
</Schedule>
</Channel>
```

I know you are saying to yourself "Hey that looks a lot like HTML."

Well actually it is written in a language called XML, which is just a step more advanced than HTML.

Now let's take a look at the CDF file and its parts.

```
<?XML Version="1.0" Encoding="iso-8859-1" ?>
```

Tells the computer that this is in the XML Language

```
<Channel HREF="HTTP://www.domain.com/channel.html" BASE="HTTP://www.domain.com">
```

Channel is the URL of the channel.  
Base is the URL of your web site.

```
<Title>Title of Channel</Title>
```

The Title of the channel.

```
<Abstract>Description of Channel</Abstract>
```

A Brief description of the channel.

```
<Logo HREF="HTTP://www.domain.com/image1.gif" Style="IMAGE" />
<Logo HREF="HTTP://www.domain.com/image2.gif" Style="ICON" />
<Logo HREF="HTTP://www.domain.com/image3.gif" Style="IMAGE-WIDE" />
```

The three images needed for your active channel, Image, Icon, and Wide.  
These three images are GIF files that must follow certain height and width requirements.

Image	16h x 16w
Icon	32h x 80w
Wide	32h x 194w

If the images you choose do not meet the above height and width requirements, they will not be displayed correctly on the user's computer.

```
<LogTarget HREF="channel.html" Method="Post" Scope="Offline">
</LogTarget>
```

Tells the computer to download the channels web page, for offline viewing.

```
<ITEM HREF="HTTP://www.domain.com/channel.html" PRECACHE="yes">
```

The item command is where the magic happens, with the item command you can tell the viewer's computer to download just about any thing. So long as the URL to the items are correct, the users computer will attempt to download it, when the scheduled update occurs.

```
<Schedule>
```

```
<IntervalTime DAY="1" />
```

```
</Schedule>
```

This tells the computer to update the "subscription" to your site at a scheduled interval.

```
</Channel>
```

This tells the computer that this is the end of the channel.

And now lets look at the code that is placed in your web page by HTML Helper:

```
<A HREF="mychannel.cdf" Onmouseover="window.status=' Add this page to your channels '; return true"
OnMouseOut="window.status=' '; return true"><IMG border ="0" SRC="addchan.gif" align="left"></A>
```

As you can see it is just a standard link, with a mouseover command for looks.

To make an active channel for your web page using HTML Helper is easy, just select the Active Channel option under the "Insert" menu. A dialog box opens, fill in the information, and select make it.

When you select "make it", a few things happen in your Work Directory...

- A file called "mychannel.cdf" is created.
- A file called "addchann.gif" is copied from the HTML Helper Directory.
- The three images you chose (Image, Icon, Wide) are copied.

And the proper link to the CDF (Active Channel) is created in your web page.



## Basic Commands

Here is the basic command that every page must follow, in order for the browser to recognize the page.

```
<html>
<head><title>Title of page here</title>Most forms of Java Script go here</head>
<body>
Content of page here.
Java Script also goes here when called for.
</body>
</html>
```

The brackets `< >` are what hold the command, every command must be "Turned on", and then accordingly "Turned off" when you no longer need it's function. For instance, in the above command statements, notice the first line is `<html>` that tells the browser that the page is beginning. The last line `</html>` tells the browser that the page is ending. If you were to create a page with out turning off the `<html>` command, the browser will continue to load the page indefinitely. The `/` is what tells the browser that the command is being terminated. Also commands must be terminated in the opposite order that they were started. To demonstrate this, look at the above commands. The `<head>` command is turned on then the `<title>` command is turned on. Then the `<title>` command is turned off using `</title>`. And lastly the `<head>` command is turned off using `</head>` .

## Link, Vlink , Alink

```
<body font color="#000000" link="#0000ff" vlink="#bbbbbb" alink="#ff0000">
```

Link, Vlink , Alink, are all used in the first body command. They are used to define the colors of Links (link="#0000ff")  
Visited Links (vlink="#bbbbbb")  
Active Link (alink="#ff0000")

In this example, text color is black, link color is blue, visited link color is light grey, and active link is red.

## Changing the Back Ground Color

```
<BODY bgcolor="#000000">
```

change the `000000` to the HEX value that you want.

## Changing the Back Ground to a Picture

```
<BODY background="image.jpg">
```

Both of the above commands are to be placed in the first `<body>` command, before you would add the link, vlink, alink, commands.



## Form Commands

In this section we will be discussing how to create a form for your web pages. You would add a form to your pages if you wanted to know specific information from those that are inquiring about your page.

The first thing you must tell the computer is that you are starting a form, and what you want done with the form data. The command to alert the computer is:

```
<FORM METHOD="POST" ACTION="mailto:your email address">
```

Notice the command did three things:

- 1.It told the computer a FORM was starting.
- 2.It stated the METHOD of dealing with the form is to POST it.
- 3.And the data should be posted to your email address through the "mailto:" ACTION.

Remember you will need to put your email address immediately after the "mailto:" without a space!! This is where the results of the form will be sent.

## The Text Box

This is a basic long box that allows for one line of text. When placed on a page, your reader will be able to write in information such as their name or their email address.

```
<INPUT TYPE="text" NAME="name" SIZE="30">
```

There are three parts to the command. Here's what they are and what they mean:

**INPUT TYPE** tells the computer that a form item is going to be placed here. Remember above you placed the command to alert the computer that form items will be placed on this page? Well, here's your first form item. This form type is "text".

**NAME** is the name you assign to the box. Remember that this is a form that will be sent to you through the mail. When you receive the mail, it won't be just like the page. Only the text will arrive, so you have to denote what each piece of text will be. When the mail arrives from this text box, it will say: name=(whatever is written in the box)

That way you know this information was written in the box marked "name". Also - remember you don't have to call the box "name." Call it what ever you want. It will arrive to you with that name. If you're using the box to get the reader's name, call it "name." If you're using the box to get the reader's email address, call it "email." etc.

**SIZE** denotes how many characters long this box will be. Make it 60 or 100 if you'd like.

## The Text Area Box

This is a larger box, like the one above, that allows your reader to write something. The difference between the Text Box (above) and the Text Area is that the Text Box only allows for one line. The Text Area, however, is much larger and will allow for as many words as you want.

```
<TEXTAREA NAME="comment" ROWS=6 COLS=40></TEXTAREA>
```

Please note that the TEXTAREA requires a </TEXTAREA> command whereas the TEXT BOX command above

did not.

Just as before...here are its parts and what they mean:

**TEXTAREA** (All one word). This yells to the form command above that here will sit another form item. This one will be a text area box.

**NAME** is the same as before. The information the reader puts in this box will arrive in your email box denoted by what ever name you use. In this case, what is written in this box will arrive in your email box with the words, "comment=" .

**ROWS** tells the computer how many rows of text it will accept.

**COLS** tells the computer how many characters will be in each row.

## The Radio Button

This is a neat little deal that places a circle on the page. That circle is active and a reader can use the mouse to click on it. When the radio button is chosen, it darkens.

```
<INPUT TYPE="radio" NAME="heading of button" VALUE="button name">
```

Here are its four parts and what they mean:

**INPUT** This yells to the computer - "HEY! Here's a form item to deal with!"

**TYPE** This tells the computer what type of form item it will be. In this case - it's a radio button.

**NAME** This names the category the button is in on your form page. Let's say you have six different choices under one heading. Like six ice cream flavors all under the heading, "Favorite Ice Cream." "Ice Cream" would be the category. It's the heading of the group of radio buttons.

**VALUE** is the name assigned to the button. Like in the ice cream example above, you have six buttons each labeled with six different flavors. Well, you would give one the value of vanilla, one chocolate, etc.

Remember that this is going to be sent to you through the mail. You will have to be able to read what the person chose. Say you had a guestbook with a section of radio buttons that ask which page the user is signing in from. Your "NAME" in the command might be "signing\_in\_from." Then each of the radio buttons is assigned the "VALUE" of each of your pages. Let's say a person chooses the radio button assigned to your home page. That button's VALUE might be, "home\_page."

## The Checkbox

The checkbox is an exact clone of the radio button except for two distinct features:

1. The item it places on the page is square and it is marked with an "X" when chosen.
2. You can check as many as you'd like.

```
<INPUT TYPE="checkbox" NAME="Signing_from" VALUE="Joes_page">
```

## The Pop-Up Box

The Pop-Up box, unless clicked on, only shows one item. But, this is your form and you can do anything you wish. Here's a Pop-Up box. You have to click on it to see all the choices. This one is for people to choose their favorite color:

```
<SELECT NAME="Favorite_Color" SIZE="1">
<OPTION SELECTED>Blue
<OPTION>Red
<OPTION>Yellow
<OPTION>Green
<OPTION>Black
<OPTION>Orange
<OPTION>Purple
</SELECT>
```

Although this looks a little bit more involved - it really isn't. It's the same thing again and again. Here are the parts and what they mean:

**SELECT** tells the computer another form is going here. This time its a SELECT or Pop-Up form.

**NAME** Same as above. This is the heading of the form item. It denotes how the results of the reader will arrive at your email box. In this case it will say; "Favorite\_Color=" and then the reader's choice.

**SIZE** denotes the size of the box. Here, 1 means 1 line or item is shown. Try putting two there if you'd like to see what it does. I prefer just one. More than one item tends to defeat the purpose of the Pop-Up Box.

**OPTION SELECTED** denotes which option will appear in the box.

**OPTION** denotes another choice that will be visible when you click on the item.

**/SELECT** finishes off the entire deal.

## Send and Reset Buttons

Well, now that the you have placed all the form items you want on your page, you need a way to have the results sent to your email box (or where ever you said this would go in the original form statement). This may be the easiest of all the items I've shown on this page.

```
<INPUT TYPE="submit"><INPUT TYPE="reset">
```

Finally make sure you end the form section of your page with this:

```
</FORM>
```



## Frames

Frames are a little different than anything you've done in html in that you are not writing a page in the traditional sense. What you are doing is creating a template where more than one page can be displayed in your current page.

```
<HTML>

<TITLE>First Frame Page</TITLE>

<FRAMESET COLS="50%,50%">
<FRAME SRC="PAGEA.html" name="A">
<FRAME SRC="PAGEB.html" name="B">
</FRAMESET>

</HTML>
```

\*Note - When using frames, in order to get your links to load into a specified frame, you must use the target command to tell the browser where the page is to load. That is why in the above example you see the code name="A" after the page address. In other words, you are literally naming your frame sections, and using the target command will affect the frame section you specify. Add the following code to Page\_A.html, and if you are connected to the internet, the Yahoo web page will load in the right side of your frame page.

```
<a href="http://www.yahoo.com" target="B">
```

If you wanted Yahoo's page to load in a new window, you would put `target="_top"` instead of `target="B"`.

What the above code does is simply split the screen into two parts and placed a different page in each part. And links placed on the left hand side load in the right. Here is what the code does:

**FRAMESET** starts any frame page. It alerts the browser that FRAMES are going here.

**COLS** denotes that I want columns. In this case I want two; each 50% of the screen.

Can I do other percentages? Of course you can. Go nuts if you want, just separate the percentages by commas and get it all to add up to 99% or 100%. 99%? Yes. You see, 33%, 33%, 33% adds to 99 and splits the screen three ways. The browser just distributes the final 1% over the three spaces.

Also you can specify a width on one side and a wild card on the other...

```
<HTML>

<TITLE>First Frame Page</TITLE>

<FRAMESET COLS="200,*">
<FRAME SRC="PAGEA.html" name="A">
<FRAME SRC="PAGEB.html" name="B">
</FRAMESET>

</HTML>
```



**FRAME SRC** denotes the source of the frame. Frames read like you do, left to right. The first source offered will be hard left. I only have two frame sections so I need only 2 sources.

**/FRAMESET** ends the whole deal.

## Adding Rows to Frames

You can add rows to frames by using the Rows command.

```
<FRAMESET COLS="50%,50%">
```

```
<FRAMESET ROWS="75%,25%">  
<FRAME SRC="LINKPAGE.html">  
<FRAME SRC="PAGE_B.html">  
</FRAMESET>
```

```
<FRAMESET ROWS="33%,33%,33%">  
<FRAME SRC="LONGPAGE.html">  
<FRAME SRC="ME.html">  
<FRAME SRC="PAGE_A.html">  
</FRAMESET>
```

Here's where the rows come in. I added a new FRAMESET command denoting ROWS. I asked for 75%, 25%. So above what I mean? What that did was break up the first column into two rows. I then offered two SOURCES for the two sections of the first column. The first frame offered a source is always the top one. I then put in a /FRAMESET to denote I was done with breaking up the first column.



## BGSOUND

Internet Explorer `<BGSOUND SRC="sound.wav">`

Netscape 2.0+

`<BGSOUND SRC="sound.wav">`

`<SOUND SRC=" sound.wav ">`

`<EMBED SRC=" sound.wav " autostart="true" autoloading="false" volume="<0-100" WIDTH=0  
HEIGHT=0 AUTOSTART="FALSE" CONTROLS=CONSOLE >`

Both Internet Explorer and Netscape use different codes for embedding sound to play at the start of your page, in the codes above, just replace "sound.wav" with the name of your sound file, the file itself has to be a .wav file or a .mid file. Although it is fun to have a sound at startup, bear in mind that even a small sound file at startup can increase the load time of your page as much as 50%. So if you really want to have a sound file, but want to keep your page fast loading, make a link to your sound file.



## Meta and Head Commands

Remember back at the beginning of this tutorial, this simple code that I showed you?

```
<html>
<head><title>Title of page here</title>Most forms of Java Script go here</head>
<body>
Content of page here.
Java Script also goes here when called for.
</body>
</html>
```

Well in this section we will be focusing mainly on the beginning part called the Head command `<head><title>Name of your site</title></head>`.

See in between the Title commands were it says "Name of your site", place the name of your site in there, and it will appear at the top of your browser when the page is loaded.

In the Head commands we can put another command called the Meta command. The Meta command is one of the most popular commands because it does six main things, granted it is capable of a lot more but these are the most popular.

```
<META NAME="keywords" CONTENT="key,word,key,word">
```

This offers key words to the search engines that use them in their searches.

```
<META NAME="description" CONTENT="Great page! Come see!">
```

This offers a description of the page for search engines that use them.

```
<META NAME="generator" CONTENT="HTML Helper">
```

This tells search engines what program was used to create the document.

```
<META NAME="author" CONTENT="Some Body">
```

This tells search engines who wrote the document.

```
<META NAME="copyright" CONTENT="Copyright © 1998">
```

This tells search engines...blah, blah, blah.

```
<META NAME="expires" CONTENT="15 September 2000">
```

This automatically expires the document in the search engine's database.

You place the Meta commands in the Head commands so they look like this:

```
<html>
<head><title>Title of page here</title>Most forms of Java Script go here</head>
<META NAME="keywords" CONTENT="key,word,key,word">
<META NAME="description" CONTENT="Great page! Come see!">
<META NAME="generator" CONTENT="HTML Helper">
<META NAME="author" CONTENT="Some Body">
<META NAME="copyright" CONTENT="Copyright © 1998">
```

```
<META NAME="expires" CONTENT="15 September 2000">
<body>
Content of page here.
Java Script also goes here when called for.
</body>
</html>
```

Remember that you should keep your page description short and to the point and as for key words you should make as long a list as possible and repeat the words as much as you like. The reason behind this is simple; most search engines will pick the pages with the most amounts of matching keywords, so you should have as many as possible.

## Additional Uses for the Meta Command

To create a new window that automatically pops up after your page is loaded, calls for use of the Meta command.

```
<Meta http-equiv="0; URL="http://www.how.com" Target="resource window">
```

META can be used to automatically play music or even load another page after a few seconds

```
<META HTTP-EQUIV="Refresh" CONTENT="5; URL=tada.wav">
```

The above will wait 5 seconds after the page is loaded then it

will play the tada wave file. I recommend using .au files. This way Netscape Player will load, play it, and close automatically

Also

```
<META HTTP-EQUIV="Refresh" CONTENT="15; URL=nextdoc.html">
```

This will load the next page in 15 seconds; this will automatically jump to nextdoc.html in 15 seconds.



## Preformatted Text

```
<pre>Display Text</pre>
```

The pre command will save you a lot of time if you have a page that is full of text. The pre command tells the browser to print the text to the screen as an exact copy of the code.



## Tables

Tables are a lot like the frame command, except instead of creating areas for other pages to be displayed on the present page, they create a space in the current page that has a fixed boarder around it. In this space links pictures and text can be placed. Tables are mainly used to define a certain area of the page that you want to stand out from the rest.

The only aspect of a table that can not be changed, is the boarder color, it defaults to Grey. Here is the code to create a simple two-cell table.

```
<table border="8" cellpadding="4" cellspacing="4" width="75%" bgcolor="#000000"><tr>
  <td width="25%"><p align="center"><a href="pageA.html">
    <font color="#00FF00" size="4">Display Text</font></a></p>
  </td>
  <td width="25%"><p align="center"><a href="pageB.html">
    <font color="#00FF00" size="4">Display Text</font></a></p>
  </td></tr>
</table>
```

Here is each part and what they do:

Table tells the computer that the table is beginning.

**Border**= this is the thickness of the border surrounding the table

**Cellpadding**= this is the amount of "padding" given to each cell, making sure that its contents do not go past the border.

**Cellspacing**= this tells the computer how far apart the cells are to be.

**Width**= this tells how much of the screen the entire table will take up.

**Bgcolor**= changes the back ground color to the desired color.

**Tr** this tells the computer that the cells are beginning.

**Font color**= changes the color of the fonts to the desired color.

**Td width**= this tells the computer how much of the screen the indivigual cells will occupy.

**/Td** this turns off the indivigual cell.

**/Tr** this turns off the table command.

As with frames, you can make as many sections in varying sizes just as long as the end result is 99% to 100%.



## Mouse Over Command

The mouse over command will display text in the bottom of the browser when the mouse passes over the defined text, link, or picture.

```
<a HREF="pageA.html" Onmouseover="window.status='Watch me Disappear!'; return true"  
Onmouseout="window.status=''; return true">Text, picture, or link</a>
```



## Adding to the Body

These are the things that you really need to know in order to make a professional looking web page. Most of the things covered in this section is to be added in the `<body></body>` commands. If the source code is to be add in another place, it will be specified.

### Adding pictures.

```

```

replace `image.jpg` with the file name and extension, the picture must be a `.gif` or `.jpg` file in order to be recognized by the browser. Notice you do not turn this command off.

### Adding Links

```
<a href="address.html">Display text</a>
```

Were it says "address.html", you place the address of the sight you wish to link to, weather it is another one of your pages or a different sight altogether.

Using a Picture as a Link

```
<a href="address.html"></a>
```

Notice that the only difference is that the image command is in the place of the display text. And in the image command you see `boarder="0"` well if you do not place that in there, your browser will place a boarder around the picture to let you know that it is being used as a link. If you want the boarder around the picture then just take out the `boarder="0"` part and it will show a boarder.

### Using a Link as a Mail Feature

```
<a href="mailto:cairnsb@geocities.com">Display Text</a>
```

As you can see the above is almost the same as the link command, except the `address.html` has been replaced with `mailto:cairnsb@geocities.com` Change `cairnsb@geocities.com` with the mail address you prefer.

### Blinking Text

```
<blink>Display Text</blink>
```

Pretty self-explanatory, this command makes the text flash at a rate of once per second.

### Header Commands

```
<h3>Display Text</h3>
```

Header commands make the text appear bigger or smaller depending on the number you place behind the h. The default size for most browsers is `h4`, but you can use `h1` to `h6`. `H1` is the biggest, and `h6` is the smallest.

### Styled Text

```
<b>Bold</b>
```

```
<u>Underlined</u>
```



<I>Italics</I>

These three commands are helpful if you just want to emphasize some part of your web page. Or really get their attention and use them over the whole page.



## A Basic Page

```
<html>
<head><title>Title of page here</title>Most forms of Java Script go here</head>
<META NAME="keywords" CONTENT="key,word,key,word">
<META NAME="description" CONTENT="Great page! Come see!">
<META NAME="generator" CONTENT="HTML Helper">
<META NAME="author" CONTENT="Some Body">
<META NAME="copyright" CONTENT="Copyright © 1998">
<META NAME="expires" CONTENT="15 September 2000">
<body>
Content of page here.
Java Script also goes here when called for.
</body>
</html>
```



## A Basic Form

```
<html>
<head><title>Basic Form</title></head>
<body bgcolor="#ffffff" text="#000000" link="#8060a0" vlink="#000080">
<center>If you have trouble Contacting us,or we do not<br>
return your reply in three business days,<br>

<a href="mailto:cairnsb@geocities.com">click here</a></center>
<FORM METHOD="POST" ACTION="mailto:cairnsb@geocities.com"><br>
Company Name<INPUT TYPE="text" NAME="company" SIZE="60"><br><br>
Contact Name<INPUT TYPE="text" NAME="contact" SIZE="40"><br><br>
Address<INPUT TYPE="text" NAME="address" SIZE="60"><br><br>
Line 2<INPUT TYPE="text" NAME="line2" SIZE="60"><br><br>
City<INPUT TYPE="text" NAME="city" SIZE="30"><br><br>
State<SELECT NAME="state" SIZE="1">
<OPTION SELECTED>
<OPTION>AL<OPTION>AK<OPTION>AZ<OPTION>AR<OPTION>CA<OPTION>CO<OP
TION>CT
<OPTION>DE<OPTION>DC<OPTION>FL<OPTION>GA<OPTION>HI<OPTION>ID<OPT
ION>IL
<OPTION>IN<OPTION>IA<OPTION>KS<OPTION>KY<OPTION>LA<OPTION>ME<OPT
ION>MD

<OPTION>MA<OPTION>MI<OPTION>MN<OPTION>MS<OPTION>MO<OPTION>MT<
OPTION>NE
<OPTION>NV<OPTION>NH<OPTION>NJ<OPTION>NM<OPTION>NY<OPTION>NC<O
PTION>ND
<OPTION>OH<OPTION>OK<OPTION>RI<OPTION>SD<OPTION>TN<OPTION>TX<OP
TION>UT
<OPTION>VT<OPTION>VA<OPTION>WA<OPTION>WV<OPTION>WI<OPTION>WY
</SELECT><br><br>
Zip<INPUT TYPE="text" NAME="zip" SIZE="10">
Other<INPUT TYPE="text" NAME="other" SIZE="30"><br><br>
<br><br><center>Additonal Comments<br><br>
<TEXTAREA NAME="comment" ROWS=6 COLS=40></TEXTAREA>
<br><br>
<INPUT TYPE="submit"><INPUT TYPE="reset"></FORM>
<center><a href="index.html">Back to Main</a></center>

</body>
</html>
```





## A Basic Frame

```
<HTML>  
  
<TITLE>First Frame Page</TITLE>  
  
<FRAMESET COLS="135,*">  
<FRAME SRC="PAGE_A.html" name="A">  
<FRAME SRC="PAGE_B.html" name="B">  
</FRAMESET>  
  
</HTML>
```



## List Command

```
<list> Topic Here  
  <li> one  
  <li> two  
</list>
```

The list command will place the topic on top of your list items, and the list items will have bullets next to them, this command is very helpful when you want to cover a wide variety of things.



## Break Command

<br>

This is one of the most simple and yet important commands in HTML, this command will act like one carriage return on a type writer, thus moving all of the text behind it down one line to the return position. Several of these can be used one after another to create space in between paragraphs.



## Horizontal Rule

`<hr>`

or

`<HR WIDTH="90%" ALIGN="CENTER" NOSHADE>`

The horizontal rule command creates a small bar across your web page. This is a fast and easy way to separate areas of your page.





## Alignment Commands

`<align="left">`

`<center>Display Text Or Picture Command</center>`

`<align="right">`

The `<align="left">` and `<align="right">` commands are primarily used in the source code of pictures to move them to the right or left of the page. An example of this would be `` And the center commands are used for any thing that you want centered in the page.



## Multi Column Text

```
<TABLE WIDTH=100%>  
<TR><TD WIDTH=50%>  
Left hand column text in here.  
<TD WIDTH=50%>  
Right hand column text in here.  
</TABLE>
```

This is done by use of the [table](#) command that will be discussed in greater detail later. For now just know that the above command creates a two column format on your page, by telling the browser that the table covers 100% of the page width<TABLE WIDTH=100%>, and there are two columns each taking up 50% of the page<TR><TD WIDTH=50%>.



## Font Commands

### Changing Font Size

```
<font size="4">Display Text</font>
```

The font command is similar to the header command, except that there are seven sizes, one is the smallest, and seven is the largest. See example.html

### Changing Font Color

```
<font color="#000000">Display Text</font>
```

Font colors are changed by using a HEX code, which is just a six-digit code that includes numbers and letters, which the browser displays as colors. In the above example, the six zero's are the HEX code for black. Remember that the # sign must always come before the HEX code in order for the browser to know that a HEX code is being used. See colors.html for a small list of color codes.

### Special Characters

< &lt;

> &gt;

& &amp;

æ &aelig;

© &copy;

non-breaking space &nbsp;



