

Platform: UNIX/PC/Mac

Level of Difficulty: Beginner/Intermediate

This is a step by step guide for webpage basics. It is divided into two main parts: "What it does" explaining the mechanics behind your actions, and "How to do it" explaining the steps to creating a unique webpage.

Introduction

All students that have an account at Rutgers University can create a webpage. You have several choices in creating a webpage: using a webpage creation program such as Claris Homepage or Netscape Composer which writes the HTML coding for you, or coding directly in HTML. Coding in HTML directly gives you more control and understanding of your webpage. Creating a webpage in HTML is easy if you follow the instructions below for set-up, basic concepts, and commands. Your webpage is an extension of your own personal imagination and style. Anything you want can somehow be placed on a webpage. This guide is designed to be as concise as possible, and therefore it skips over some details about how to use some of the various tags. At the end of this document there is a list of suggested links to web pages that clarify what these tags mean. The reason these tags were included without any explanation was that once you have a basic understanding of how HTML works, you can often figure what a tag does on your own just by experimenting with it. Hopefully this document will give the basic understanding that you need. This guide to HTML also lists all the tags that current browsers are likely to recognize. Also included is all the elements in the official HTML 4.0 recommendations with common attributes, as well as Netscape and Microsoft extensions.

Prerequisites:

In order to understand this document and be able to create your own web page using html, you must have a basic knowledge and understanding of Unix and how a web browser functions.

What is HTML?

Html stands for Hyper Text Markup Language. This is a language that standardizes the structure of documents for retrieval across the Internet. HTML is an abstract description of a page layout that can be read by a number of different computer systems.

Html gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hyperlink at the click of a button.

- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spreadsheets, video clips, and other applications directly in their documents.

Basic Concepts:

Text editors- There are a couple of different UNIX text editors available for use in creating and editing files. These UNIX text editors are applications that can be run either in a telnet session or an X-windows session. The two most popular UNIX text editors here at Rutgers are Emacs and Pico. Using these UNIX text editors allows you to program in any computer language and translate it into code that the computer will be able to read and execute. There are also non-UNIX based text editors that are available as well. On Macintoshes you can use BB Edit and on PC's you can use Wordpad or Notepad. It suggested that you stay away from using MS Word or Word Perfect because of their auto spell check and syntax check functions. These functions create difficulties in having the code being interpreted properly and affect how smoothly the web page will be executed. The difference between UNIX and non-UNIX based text editors is that by using the non-UNIX based text editors you will have to use Fetch or FTP in order to save the file to your Eden account. By using the UNIX based text editors, you can save yourself a couple of extra steps since the files will be directly saved to Eden.

Tags -To denote the various elements in an HTML document, you use tags. HTML tags consist of a left angle bracket (<), a tag name, and a right angle bracket (>). Tags are usually paired (e.g., <H1> and </H1>) to start and end the tag instruction. The end tag looks just like the start tag except a slash (/) would precede the text within the brackets. Not all tags are supported by different web browsers, especially if you have an older version. If a browser does not support a tag, it will simply ignore it. However, any text placed between a pair of unknown tags will still be displayed. The concept of an end tag is an important thing to keep in mind when trouble shooting your code. **Web Browser-** This is the program with which you will view your web page. In the Rutgers Computer labs the program is called Netscape.

Graphics/Backgrounds – Backgrounds, graphics, buttons, and icons can easily be chosen over the web. If you see an image on Netscape you would like to copy and place it on your own website, right click on the image or background and select Save As, and save the image in your public_html directory. You can also find out how to program these things yourself by going to the menu bar and selecting “View” and then selecting “Page Source.” This will show you a copy of the actual code used to program certain backgrounds and graphics.

Personal Pictures- Photos, original drawings and objects can easily be copied using a scanner in any of the computer labs. Follow the directions on how to use the scanner, save your picture and FTP it to your email account. If you are unfamiliar with using FTP, see the document: “Fetch/FTP: File Transfers, From here to there.”

Basic UNIX Commands

In order to have your web page show up properly you will need to be familiar with certain UNIX commands. UNIX has a command line interface, which means that you have to type in commands, rather than use a mouse, in order for UNIX to execute commands. Below is a list of some commands that will come up frequently when trying to place your page on the web. (For a further understanding of UNIX, please refer to the UNIX documentation that we have available in the lab.)

ls -al

lists all files in detail, including size, permissions, last date of modification, etc. This is useful to check space and permissions.

rm <file name>: This command removes/deletes a file that you no longer want. **Be very careful using this command once you have deleted the file there is no way to retrieve it again.**

rm -r <directory name>

If you wish to delete a whole directory and all folders and files within that directory, use the command. **Again be very careful with this command, once you delete a directory it will be gone with no way to retrieve it.**

mkdir <directory name>

This makes a directory of your choice. Basically it creates a folder in the file you are in, this is useful if you want to create another page in your coding or if you have a lot of different graphics you would like to divide by type, name or style.

cd <directory name>

This moves you from folder to folder by changing the directory that you are currently in.

chmod 755 <file name>

This is the command used to set permissions. Permissions are the **r**'s, **x**'s, and **w**'s, to far left of the screen after you do an `ls -al`. the **r** = read, the **w** = write, and **x** = execute. This changes or sets the permissions of all the files in a directory to allow certain people to read, edit or run programs. Usually you will want to set the permissions to 755. This gives you all three permissions, **rwX**, and gives everybody else just **rx** permissions. This way you are the only person who can write to the file. **This is very important because you do not want anybody else writing to your files.** (For a further understanding of permissions please refer to the UNIX documentation that we have available in the lab.)

mv <source file> <destination file>

This command moves a file from one directory to another. Type in the file name first, then where you want the file to be placed. This is useful if you forgot to save directly into your `public_html` directory.

This is also a rename command. By typing in the original name and then the new name you can change what your file is called. You can check this by typing in the command, then **ls** to see if it worked.

Emacs commands

ctrl-x, ctrl-s : this save command for emacs will save all changes in the current window. We suggest you save after ever few lines in case your computer crashes.

ctrl-x, ctrl-c : the quit command for emacs. If you have any changes it will first ask if you want to save changes. Type yes to save changes, otherwise type no.

Before you start:

Bypassing Menus

Menus, also known as lush, is a user friendly interface that is intended help students that are unfamiliar with UNIX accomplish simple tasks like checking their e-mail. Menus can become quite cumbersome, however, when it comes to creating webpages. So, in order to effectively create your webpage you will to bypass Menus and get to tcsh shell (**er#%>**). **If you do not have Menus you can skip this set of instructions.**

How to do it:

Start Telnet session and a web browser on your computer. Remember to click in the window you are using before typing. In the telnet window read the top line of your screen. If you have something called Rutgers Main Menu, follow these instructions to get to an **er#%>** prompt, otherwise go straight to Step 2.

- 1: From the first Menu choose **tools**.
- 2: From the second Menu choose **cmd**.
- 3: When it asks you to enter in a command, type: **tcsh**.

You should now be in a tcsh shell where you will have a prompt that looks something like this:

```
er#%>
```

Creating your public_html folder

What this does: Before you create a webpage, you need a place to store all your information. This is saved in a directory in your email account called public_html. Any pictures, graphics, coding or animations **must be saved here** to be seen on your webpage. By using the webinit command it will set up your eden account with a public_html directory and an index.html file. The index.html file is created by default when using the webinit command. This is a file that has some basic html code in it. It can be thought of as copying a template for a webpage to your account. **If you already have a public_html folder you can skip this set of instructions.**

How to do it:

- 1: At the **er#%>** prompt type: **webinit**
- 2: Then type: **cd public_html**. You are now inside the public_html directory. You can now begin coding or saving images to this directory.

Note* Before you move on, list your files with `ls -al`. If you'll notice, there is already a file in your `public_html` called **index.html**. This information will be important later when we discuss how to figure out what the address to your page is.

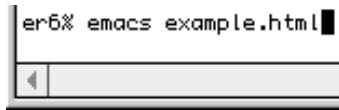
Step 1: Starting to code

What this does: Now you will start to write code in your emacs editor. To create a file in Emacs is very simple. From your prompt (`er#%>`) type in emacs, then a space, then a unique file name. If you type in the name of a file that already exists then you will be writing to that file, which would screw up the file. Then once you have Emacs up and running you will see a lot of blank space. This blank space is known as a buffer, this is the space where you will type you code into.

How to do it:

1: At the `er#%>` prompt type: `emacs <file name>.html`

(It is important to place the extension .html on the end of the file name if you want you page to be seen properly)



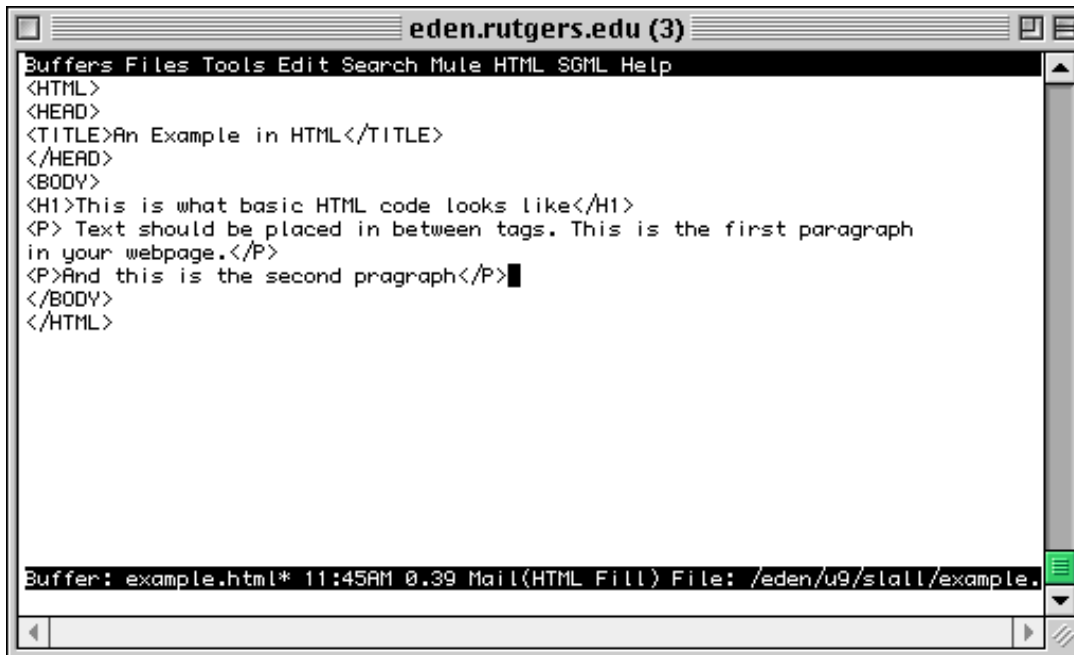
2: Once you have done this you should now be at a buffer in Emacs. This is where you begin to type your code.

All commands in html are in between `< >`, called tags.
To end any tag use `</ >` to close the tag.

GENERAL (all HTML documents should have these)

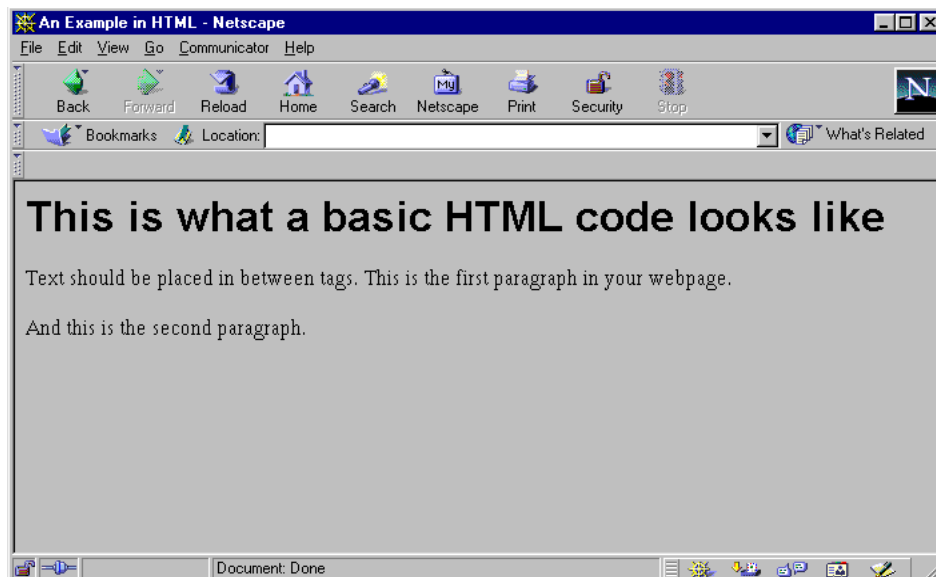
Document Type	<code><HTML></HTML></code>	(beginning and end of file)
Title	<code><TITLE></TITLE></code>	(must be in header)
Header	<code><HEAD></HEAD></code>	(descriptive info, such as title)
Body	<code><BODY></BODY></code>	(bulk of the page)

The following is an example of the basic structure and syntax of what HTML code looks like when programming a webpage on Emacs:

A screenshot of a terminal window titled "eden.rutgers.edu (3)". The window displays raw HTML code. The code includes a title "An Example in HTML", a heading "This is what basic HTML code looks like", and two paragraphs of text. The status bar at the bottom shows the file path: "Buffer: example.html* 11:45AM 0.39 Mail(HTML Fill) File: /eden/u9/stall/example.".

```
<HTML>
<HEAD>
<TITLE>An Example in HTML</TITLE>
</HEAD>
<BODY>
<H1>This is what basic HTML code looks like</H1>
<P> Text should be placed in between tags. This is the first paragraph
in your webpage.</P>
<P>And this is the second paragraph</P>
</BODY>
</HTML>
```

Below is an example of what it translates into on a web browser:



Important things to keep in mind when coding:

1. When coding for a page the very first thing should always be an `<HTML>` tag and the very last thing should always be an `</HTML>` tag. This is so that when the file is read, the computer recognizes what kind of a file it is dealing with and how to interpret it, and so that it knows when to stop reading it.

2. You should always close your tags. When the browser is interpreting the file it takes its instructions very precisely. Generally, a lot of errors occur when you are not careful about closing your tags.

3. You should keep a Netscape window open while you are coding your page in Emacs. The purpose of having both these windows is so you can reload and check your progress or test a potential graphic. By repeatedly reloading your browser after any major change, you can quickly pinpoint mistakes or dislikes in the programming rather than waiting to the end and then trying to find your error.

If you are first starting out a webpage you must set your permissions before you can view your page. After a few lines quit emacs and go to Step 2.

Note: If you see a style you like on someone else's webpage, look at their coding by going to the menu bar at the top of your web browser and selecting View, then selecting Page Source. Good pages will have coding that is easy to read and understand. Copy any graphics and coding needed to achieve the effect you want.

Step 2: Setting Permissions

What this does: This allows certain people access to your page. There are three numbers, 1-7, in a certain order to set these permissions. The norm is 755. The order and numbers are very important:

7- the first digit and refers to yourself. By using 7 you have set the permission for only you to read, edit and execute the file.

5- the second digit and refers to a group. By using 5 you have given permission for only to anyone in the same group as you to read and execute your file, but they can **NOT** write to it.

5- the third digit and refers to the world. Like the second digit you have given permission for the world to read and execute your file but not to edit or write.

How to do this:

1: Exit Emacs


2: At the `er#%>` prompt type: **chmod 755 <file name>**
This changes permissions on all the files.

3: Then type: **cd**

This will move you up and out of your `public_html` directory and back into your home directory.

4: Type: **chmod 755 public_html**

This changes your permission on the whole folder.



If you type in `ls -al` which lists all your files in long format you can check to see if you set your permissions correctly. (permissions are the r's, w's, and x's, to the far left side of the window) If you did then they should look like the example above.

To get to your page on a web browser you have enter in your address in the location bar. To figure what your address is you need to figure out what the path is to page. Your account is on eden so the first part would be `www.eden.rutgers.edu` then a slash (/). Once the browser is looking on eden it then needs to find where your home directory is so the next part of the would begin with a tilde (~) and then your user name. Now the next part of the path depends on where you placed your file with your html code. There are two options.

1. If you started your webpage in your `index.html` file then you do not have to add anymore to the path. The way your eden account is configured allows for it to look in your `index.html` file by default. So if you ever start an `index.html` file and put your webpage code in it, all you have to type into the location bar is:

`http://www.eden.rutgers.edu/~<YOUR USERNAME>`

*Since this feature allows for a simpler web address that is easier to remember, it is suggested that you place any personal homepage that you create in your `index.html` file.

2. If you started your web page in a file other than `index.html` then you would have to add the name of the file to the path. For example, if you started a file in Emacs called `examples.html` then you have to place a slash after your username portion of the path and add `examples.html`.

How to do it:

If your code is in an **index.html** file:

1: Go to the location bar at the top of your Netscape window.

2: Edit the address to read: **http://www.eden.rutgers.edu/~<YourUsername>**

3: Hit enter

- If a blank page shows up saying error, try the address again. There maybe a typo or error.
- If the page shows up saying FORBIDDEN, you have not set your permissions correctly.

OR

If your code is in a file with a unique name, for example, example.html:

1: Go to the location bar at the top of your Netscape window.

2: Edit the address to read:

http://www.eden.rutgers.edu/~<YourUsername>/example.html

3: Hit enter

- If a blank page shows up saying error, try the address again. There maybe a typo or error.
- If the page shows up saying FORBIDDEN, you have not set your permissions correctly.

Step 3: Reviewing your webpage

What this is: When you have finished a few lines in your coding you might want to check your progress. This is a simple process of saving in emacs and reloading in Netscape.

How to do this:

1: Save in emacs

2: Go to the Netscape browser and hit the **reload** button located at the top of your Netscape window. Every time you reload you should see the changes that you've made in your code. If not, wait a moment and click on reload again- sometimes a lag occurs, slowing the saving process.

Step 4: Re-editing your webpage

What this is: This is the step in determining what to keep and not to keep. Try changing the order or colors to get the effect you want. Use some or all of the options at the end of this document for different styles.

SYMBOLS USED

URL URL of an external file (or just file name if in the same directory)

? Arbitrary number (i.e. <H?> means <H1>, <H2>, <H3>, etc.)

% Arbitrary percentage (i.e. <HR WIDTH="% "> means <HR WIDTH="50% ">, etc.)

*** Arbitrary text (i.e. ALT="***" means fill in with text)

\$\$\$\$\$ Arbitrary hex (i.e. BGCOLOR="#\$\$\$\$\$" means BGCOLOR="#00FF1C", etc.)

::: Arbitrary date (i.e. DATETIME=":::" means "1994-11-05T08:15:30" etc.)
@ Email address (i.e. "mailto:@" means "mailto:kevin@werbach.com" etc.)
,,, Comma-delimited (i.e. COORDS=",,," means COORDS="0,0,50,50", etc.)
| Alternatives (i.e. ALIGN=LEFT|RIGHT|CENTER means pick one of these)

COMPATIBILITY (remember, HTML is evolving and browser implementations vary)

(no notation) In the HTML 3.2 spec.; should work on all browsers
4.0 Introduced in HTML 4.0 recommendation
N1 Netscape extension introduced with Navigator version 1.0 or 1.1
N2 Netscape extension introduced with Navigator version 2.0
N3 Netscape extension introduced with Navigator version 3.0
N4 Netscape extension introduced with Navigator/Communicator version 4.0
MS Microsoft Internet Explorer extension
* Netscape extension now included in the HTML 4.0 specification

HTML tags:

Colors: Colors are set by in alphanumeric order. The computer doesn't have every color imaginable programmed and saved into its memory. The computer starts of with certain primary colors: red, green, and blue (**RGB**). When you try and set the font color you will have to enter a number sequence that corresponds to a particular color. These numbers tell the computer how much red, green, and blue are in a color tone to get a certain hue. Here is an example:

```
<Font color=" #D50047">
```

This will set your font color to a shade of red. End it by using the following tag:

```
</font color>
```

The background color is specified within the <body> tag at the beginning of your document. So, if you write in your HTML doc <body bgcolor=yellow> You will create with great efficacy a yellow background. "So," you may be thinking, "which colors can I use? If I type: <body bgcolor=periwinkle> "will the browser know I'm looking for a light bluish gray?" The short answer is: No. Only a limited number of colors can be specified by name (periwinkle isn't one of them), and different browsers recognize different sets of colors. So if you want to be precise about your background you'll want to replace the name of the color, with a hexadecimal code. So your HTML would look something like this: <body bgcolor=#ffffff> Notice all those "f"s in the code? Well, that's how one says "white" in hexadecimal.

Here are some sample colors for you to experiment with. There are many more that are available but for right now you can use these colors to get some practice coding with.

Hex Code

#FFFFFF #FF9966

#FFFFCC #FF9933

BACKGROUNDS AND COLORS

Tiled Bkground <BODY BACKGROUND="URL">
MS Watermark <BODY BGPROPERTIES="FIXED">
Bkground Color <BODY BGCOLOR="#\$\$\$\$\$\$"> (order is red/green/blue)
Text Color <BODY TEXT="#\$\$\$\$\$\$">
Link Color <BODY LINK="#\$\$\$\$\$\$">
Visited Link <BODY VLINK="#\$\$\$\$\$\$">
Active Link <BODY ALINK="#\$\$\$\$\$\$">
(More info at <<http://webach.com/web/wwwhelp.html#color>>)

Background: The background is the easiest way to change the overall look. You can make it a solid color or a patterned image.

Color- coding: <body bgcolor="#0055FF">

This will change the background color to a shade of blue.

Image-coding: <body background= "imageFileNameHere">

Text: Text can change in size and color.

Color- or

Size- <H1> is the largest size. The larger the number after <H>, the smaller the text.

Paragraph

Paragraph tags are the most user-friendly of all the HTML tags. Inserted at the beginning of each paragraph, the <p> tag skips a line before each block of text. There's no need to close off the <p> tag with a </p> ; it's implied when you go on to the next tag.

Hard Breaks/ Paragraphs:

 is a hard break. This is equivalent to hitting the return key once in a word processing document.

<p> is the beginning of a paragraph. The text will be spaced a line away from the last object on the page.

DIVIDERS

Paragraph <P></P> (closing tag often unnecessary)
Align Text <P ALIGN=LEFT|CENTER|RIGHT></P>
Justify Text <P ALIGN=JUSTIFY></P>
Line Break
 (a single carriage return)

Clear Textwrap	<BR CLEAR=LEFT RIGHT ALL>	
Horizontal Rule	<HR>	(a horizontal line)
Alignment	<HR ALIGN=LEFT RIGHT CENTER>	
Thickness	<HR SIZE=?>	(in pixels)
Width	<HR WIDTH=?>	(in pixels)
Width Percent	<HR WIDTH="% ">	(as a percentage of page width)
Solid Line	<HR NOSHADE>	(without the 3D cutout look)
No Break	<NOBR></NOBR>	(prevents line breaks)
Word Break	<WBR>	(where to break a line if needed)

Other text formatting

You can use one, two or all three of these options on text. Bold (**bold**), italic (*italic*), and underlined (underlined) text are used to show emphasis. Happily, they're three of the easiest tags to master. For example, to create bold text, just sandwich the word(s) you want to appear in bold with the tags and .

Bold- Text will be bold

Underline- <U> Text here will be underline </U>

Italics: <I> Text here will be in italics </I>

Note that any text placed outside of the open and close tags will not be affected.

PRESENTATION FORMATTING (author specifies text appearance)

	Bold		
	Italic	<I></I>	
4.0*	Underline	<U></U>	(not widely implemented)
	Strikeout	<STRIKE></STRIKE>	(not widely implemented)
4.0*	Strikeout	<S></S>	(not widely implemented)
	Subscript		
	Superscript		
	Typewriter	<TT></TT>	(displays in a monospaced font)
	Preformatted	<PRE></PRE>	(display text spacing as-is)
	Width	<PRE WIDTH=?></PRE>	(in characters)
	Center	<CENTER></CENTER>	(for both text and images)
N1	Blinking	<BLINK></BLINK>	(the most derided tag ever)
	Font Size		(ranges from 1-7)
	Change Font Size		
	Font Color		
4.0*	Select Font		
N4	Point size		
N4	Weight		
4.0*	Base Font Size	<BASEFONT SIZE=?>	(from 1-7; default is 3)
MS	Marquee	<MARQUEE></MARQUEE>	

STRUCTURAL DEFINITION (appearance controlled by the browser's preferences)

Heading	<H?></H?>	(the spec. defines 6 levels)
Align Heading	<H? ALIGN=LEFT CENTER RIGHT></H?>	
Division	<DIV></DIV>	
Align Division	<DIV ALIGN=LEFT RIGHT CENTER JUSTIFY></DIV>	
4.0 Defined Content		
Block Quote	<BLOCKQUOTE></BLOCKQUOTE>	(usually indented)
4.0 Quote	<Q></Q>	(for short quotations)
4.0 Citation	<Q CITE="URL"></Q>	
Emphasis		(usually displayed as italic)
Strong Emphasis		(usually displayed as bold)
Citation	<CITE></CITE>	(usually italics)
Code	<CODE></CODE>	(for source code listings)
Sample Output	<SAMP></SAMP>	
Keyboard Input	<KBD></KBD>	
Variable	<VAR></VAR>	
Definition	<DFN></DFN>	(not widely implemented)
Author's Address	<ADDRESS></ADDRESS>	
Large Font Size	<BIG></BIG>	
Small Font Size	<SMALL></SMALL>	
4.0 Insert	<INS></INS>	(marks additions in a new version)
4.0 Time of Change	<INS DATETIME=":::"></INS>	
4.0 Comments	<INS CITE="URL"></INS>	
4.0 Delete		(marks deletions in a new version)
4.0 Time of Change	<DEL DATETIME=":::">	
4.0 Comments	<DEL CITE="URL">	
4.0 Acronym	<ACRONYM></ACRONYM>	
4.0 Abbreviation	<ABBR></ABBR>	

LINKS

Without links, the World Wide Web would not be all that useful. After all, links are what make the web a Web. You can add links (or "hyperlinks," if you want to be formal about it) that connect your page to any other accessible site on the Net. Likewise, any other site can link back to you. When you want to build a link, the first thing you need is the URL of the page you're linking to. Then decide which word (or words) on your page will act as the link. The link will be the hot spot, where people can click if they want to go somewhere else. Links are generally underlined, and appear on the browser in a different color than the rest of the text (the default color for links is blue, but this can be altered). To create a link, you should sandwich your selected words between the "anchor" tags, and , replacing the ? with the correct URL. The "href" may seem cryptic, but it's just an abbreviation for Hypertext REFerence. Translation: It tells the browser what file it's referencing and where to find it.

Links:

To another of your own pages:

`<a href= http://www.eden.rutger.edu/~<username>/<fileName>> LinkName `

To somewhere else on the web:

`<a href= http://<WebAddress>> LinkName `

example:

` Rutgers Webpage`

Linking through pictures:

`<a href= http://<WebAddress>> <img src =<filename>>`

example:

` `

Email Address: If you wish to add a link to send you email-

`<a href= <emailAddress>>text here `

Note that the URL is enclosed in quotations marks, and that there are no spaces between any of the characters, from href to the closing >. And always remember to close off a link with the tag, or your entire page will become one long messy link.

LINKS AND GRAPHICS

Link Something ``

Link to Location `` (if in another document)

`` (if in current document)

4.0* Target Window ``

4.0* Action on Click `` (Javascript)

4.0* Mouseover Action `` (Javascript)

4.0* Mouse out Action `` (Javascript)

Link to Email ``

Specify subject `` (use a real ?)

Define Location ``

Display Image ``

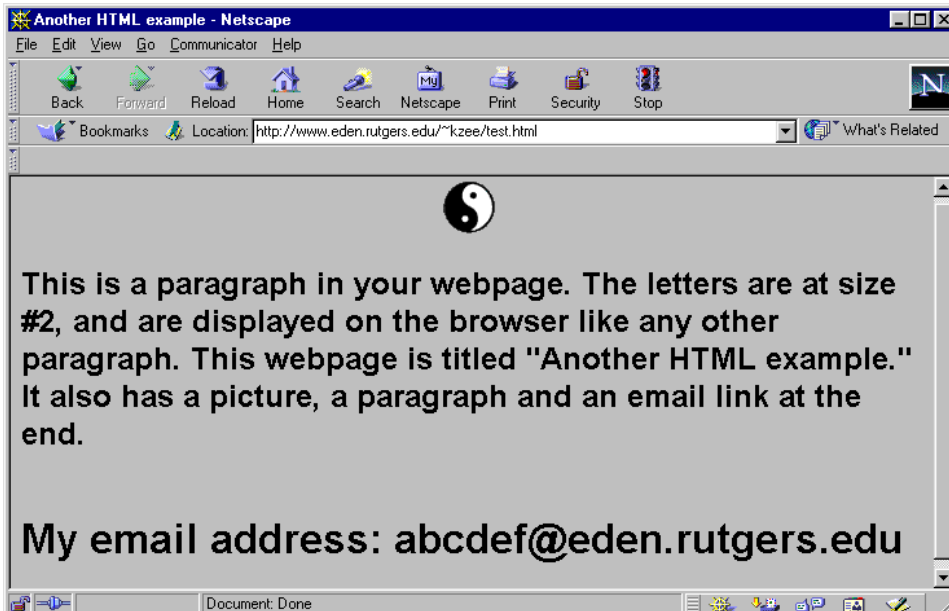
Alignment ``

N1 Alignment `<IMG SRC="URL" ALIGN=TEXTTOP|ABSMIDDLE|
BASELINE|ABSBOTTOM>`

Another more complicated HTML example:

![Screenshot of a Netscape browser window showing the raw HTML code of a webpage. The title bar reads 'eden.rutgers.edu (3)'. The menu bar includes 'Buffers Files Tools Edit Search Mule HTML SGML Help'. The code is as follows: <pre><HTML>
<HEAD>
<TITLE>Another HTML example </TITLE>
</HEAD>
<BODY BACKGROUND=](picture.jpeg)

Viewed from a browser:



Where to get more help

In Unix you can get a list of commands by using the Man Pages. At the `er#%>` prompt type in: `man <command name>` Many webpages have been created to aid beginner HTML programmers. Simply go to a search engine on your web browser and search any keyword you need help with such as html, tables, graphic, or webpage. If you are in a Campus Computing Center, feel free to contact the consultant on duty. They will be able to help you in finding aid in manuals or help on the web.

Links to other helpful pages:

<http://werbach.com/web/wwwhelp.html#color>

<http://www.eden.rutgers.edu/~nathanl/Resources.html>

<http://werbach.com/barebones>

<http://www.webmonkey.com> (Enter "HTML Tutorial" for your search)