

# **CREATING YOUR OWN HARD-CODED WEB PAGE**

**[USING A STANDARD TEXT EDITOR]**

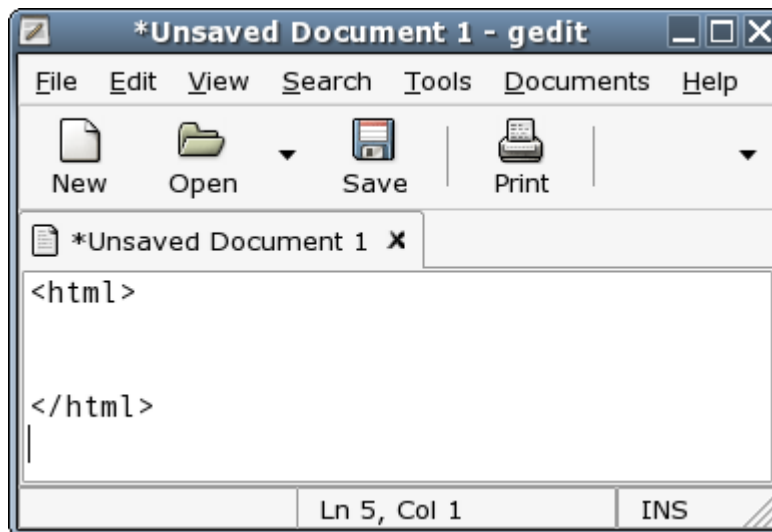
**AUTHOR: RICKY BRYCE**

Disclaimer: Although I believe the information in this document to be accurate, it is your responsibility to verify the accuracy of the information contained in this document before implementing it in any way. By continuing to read this document you agree to hold no one liable who has written or distributed this document... Even for negligence.

## Creating a hard-coded web page in HTML (By Ricky Bryce)

HTML is a mnemonic meaning “**Hyper Text Markup Language**”, and is the language that renders most web pages to a user's web browser. Most HTML tags are opened and closed by using special codes that are enclosed in the “<” and “>” characters on the keyboard.

- 1) **Open the Editor** -- For this example, we are going to write a very simple web page.
  1. **For RedHat/CentOS Linux** – Click Actions|Run Application from the top of your screen, and type “**gedit**” as the application you wish to run.
  2. **For Windows** – Open a text editor such as **notepad** (click Start|Programs|Accessories|Notepad).
- 2) **Declare the format** – Next, declare the format in which the page will be served. In this step, we will open the entire page with the <html> tag, and close the entire page with the </html> tag. Most all tags in html will follow this same “<tag> </tag>” convention. In your text editor, type <html> to open the page, and then type </html> to close the web page as shown. The entire content of the page will be coded between these two tags.




- 3) **Specify the head and body** – Each web page has two sections: a head and a body. The head contains information about the page, such as the title, and the body is the actual content of the page, such as text, pictures, and links. Within our html document, let's specify the head and body as shown.



Congratulations! You have just created a web page! There's just not a lot of content in it right now, but at least you now have the structure of the page in place that we can work from. Within your HTML document, you have both a head (information about the page, such as the title) and a body (which is the actual page)


- 4) **Title the page** – The title is what usually appears in the title bar of the browser window. We know the title is just information about the page, so the <title> tags will appear within the <head> tags. Add the title as shown, but substitute the text between the title tags for the actual title that you want to appear on your own page. The html tags are not case-sensitive, so <title> is the same as <TITLE>, however to keep your code looking nice, you may want to use the same case for all of your html tags. In most cases, white space does not effect the browser's output, so to keep the code easy to read, I have indented the title within the head of the document. The person looking at your web page is not going to see all these codes anyway, but later on, when your page becomes more complicated, you will thank yourself for keeping your code clean.

```
<html>
  <head>
    <title>Writing HTML</title>
  </head>
  <body>
  </body>
</html>
```



- 5) **Add some content** – Finally, we are going to add some content to the page that the user will be able to see. This content will appear within the **BODY** of the document. Let's start off by adding some unformatted text, and later on we will change the way this text appears on the page.

```
<html>
  <head>
    <title>Writing HTML</title>
  </head>
  <body>
    This is my web page
  </body>
</html>
```

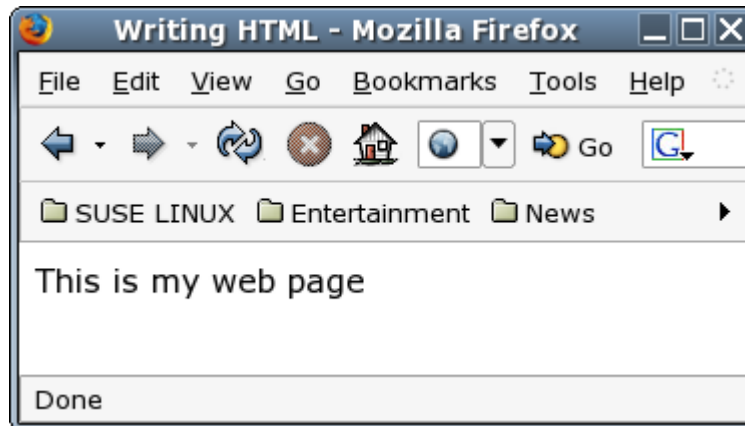


You have a web page now, with content in the body of the page. Next, we need to save this page, and see what the page will render in a web browser.

- 6) **Save the page** – To keep things simple, let's just save this page on your desktop. Most files for web pages end in the “.html” extension. This main home page is called the **index** page. Therefore, our file will be named “index.html”. For viewing on our own computer, the name of the file isn't that important, but when we upload it to the web server, the server will be looking for a file named index.html as a starting point for a user entering your site. To save the page, click File|Save As from the menu bar of your text editor. (The menu bar is right below the title bar at the top of your text editor)
1. If you are using Windows, change the “Save In” Box to “Desktop” if it's not set to desktop already. Change the filename to “index.html”, and change the “Save as type” box to “All files” This is very important for default settings of windows. By default, windows will hide the file extensions of known file types, so if you forget to select “all files” when saving your document, you could end up with a file that is actually named index.html.txt with the .txt extension hidden. This means that you would have saved the file as a text document, and not an html document.
  2. If you are running “**gedit**” you will get the following screen once you click File|Save As. Just name your file “index.html”, make sure your “Save in folder” is set to Desktop as shown, and press the save button.

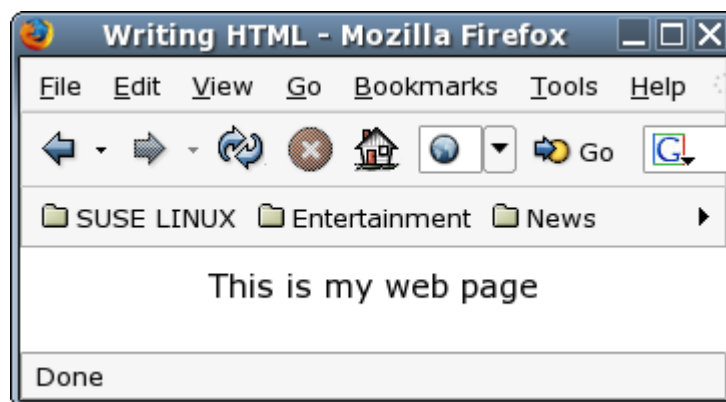


- 7) **View the page** – Next, we need to open our page in a browser window to see what the browser rendered. Minimize all applications you have open (including the text editor). To minimize a window, click the “-” sign in the upper right corner of each window. On your desktop, you will see the file you created, index.html. Double click the index file, and it should open in your web browser. If you are running windows, and the file opened in notepad instead of the web browser, you probably did not set the file type to “all” when saving your document. Delete the file, and try to save it again to your desktop. You will notice your content appears on the page, and the title bar at the top of your browser window reflects the text you placed in the head of your page.



- 8) **Center the text** – Now let's look at formatting this text. Use the <center> tags as shown to center the text horizontally within the browser window. When you are finished, click the save button on your text editor (or click File|Save) from the menu bar. Double click the index file on your desktop again to see how these tags changed the format of your text on the page.

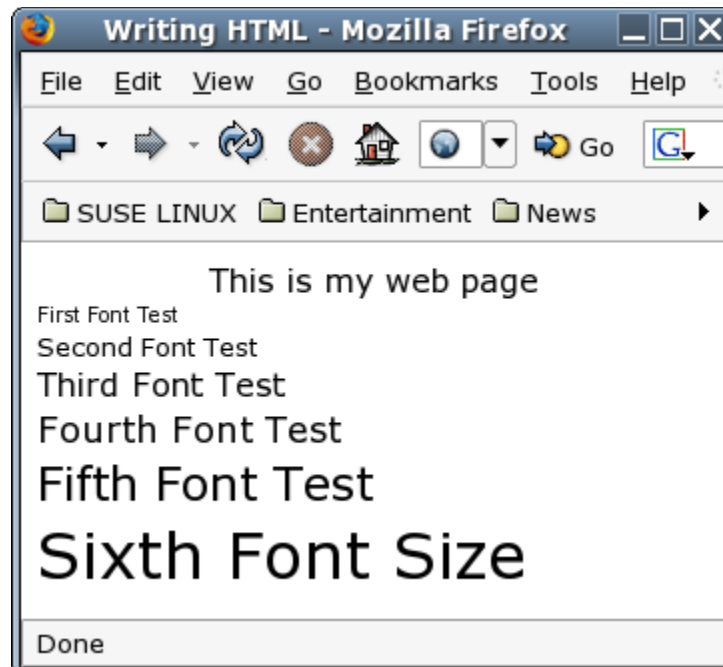
```
<body>  
  <center>This is my web page</center>  
</body>
```



- 9) **Change the font size** – Let's learn about the `<font>` tag. This is a versatile tag that allows you to change the size and color of your text. Change the body of your document as shown. After each line, you will notice a `<br>` tag as well. This is a break, which means the text following the `<br>` tag is to appear on the next line. Note: If you are using notepad in Windows, you will not see the color formatting that we have in gedit under Linux. Your page will still work just fine. After you save your work, open the page in the web browser again, and you will see what the font tag does for you.

```
<body>
  <center>This is my web page</center>

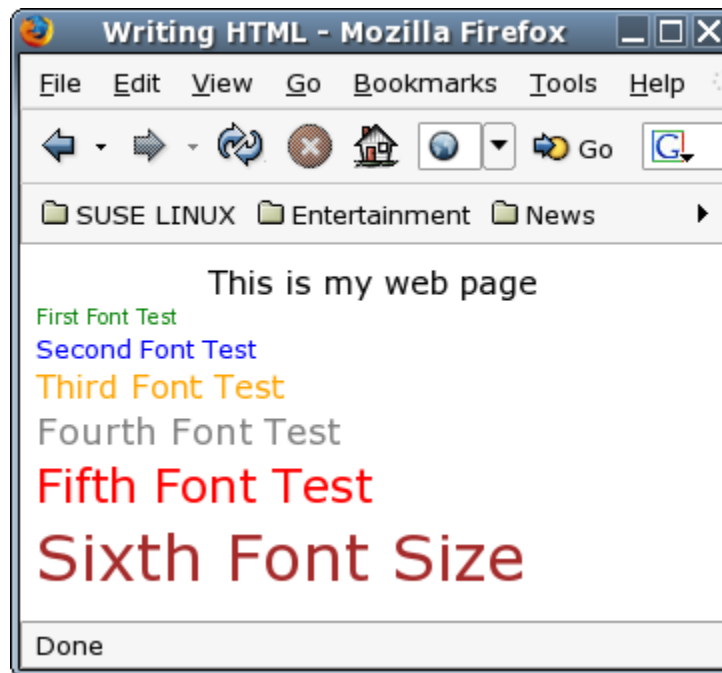
  <font size="1">First Font Test</font><br>
  <font size="2">Second Font Test</font><br>
  <font size="3">Third Font Test</font><br>
  <font size="4">Fourth Font Test</font><br>
  <font size="5">Fifth Font Test</font><br>
  <font size="6"> Sixth Font Size</font><br>
</body>
```



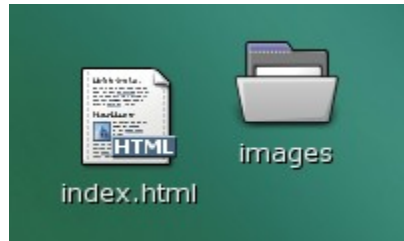
- 10) **Add some color** – The font tag can also be used to color the text. In this example, we will change the color of the text using parameters such as red, green, blue, etc... If your application requires more precision, you can use a color chart, and use special codes. Lets just keep things simple for this exercise. You can change your html document as shown, save your document again, and refresh the web browser. (click the refresh button in the tool bar of your web browser)

```
<body>
  <center>This is my web page</center>

  <font size="1" color="green">First Font Test</font><br>
  <font size="2" color="blue">Second Font Test</font><br>
  <font size="3" color="orange">Third Font Test</font><br>
  <font size="4" color="grey">Fourth Font Test</font><br>
  <font size="5" color="red">Fifth Font Test</font><br>
  <font size="6" color="brown">Sixth Font Size</font><br>
</body>
```



- 11) **Add Pictures** – To keep our directory clean on the server, we are going to create a new folder called “images”. This will be the folder that we will put all of our images in. For this example, and in most real applications, the images directory is in the same location as the **index.html** file. Let's create a folder on the desktop, and put a picture in the folder. For this example, I have a picture called **unix.jpg** that I will place in the “images” folder which is on my desktop.



- 12) The `<img src>` tag is a stand alone tag (like the `<br>` tag). We will use the `<img src>` tag to embed the picture into our web page using the relative path of our picture. Remember the picture is located in the “images” folder, so we will just enter the following text to the bottom of the page (just before we close the `</body>` of the page).

```
<br>
</body>
```

- 13) Save your code, and open the index.html file from your desktop in the web browser. Let's see what this `<img src>` tag does for us.



- 14) You can modify the size of the image with the height and width modifiers of the `<img src>` tag as shown:

```
<br>
</body>
```

- 15) Save your work, and refresh the web browser. You will see the size of the image has changed. Remember when we talked about the `<center>` tags? Try placing the center tags around the `<img src>` tag, and see if that centers your picture on the web page! (remember to save your work after you put the `<center>` tag before the `<img src>` tag, and the `</center>` tag after the `<img src>` tag)

```
<center></center><br>
</body>
```

16) **Add a Hyperlink** – A Hyperlink is a link to another web page. When the user moves his mouse over a Hyperlink, a finger will appear indicating that the user can click on the text, and the browser will be taken to another page. Hyperlinks are set up using the **Anchor** `<a></a>` tags. You can place text, or an image between the `<a></a>` tags. Whatever you put between the anchor tags is what the user must click on to get to the target location of the anchor. In this example, I will add an anchor to my page that allows the user to get to Yahoo.com. Here is the format of the **Anchor** command: Let's add this line just above the `</body>` tag.

```
<a href="http://www.yahoo.com">Yahoo's Page</a><br>
```

17) Notice that within the `<a>` tag is an href parameter. This parameter is the actual page the user is going to be sent to. Notice that the text “Yahoo's Page” is between the `<a></a>` tags. This is the text the user must click on to go to the link. Then we must close the anchor `</a>`, otherwise we would make the rest of the page an anchor as well. You can use the anchor command to link to other pages you create as well. The other pages can be named anything you like as long as they have the .html extension, however, your entrance page **MUST** be named index.html. For example... If I create a second page, and title it ricky.html, I might save it to the same location as the index.html file (in this case, I would save ricky.html to the desktop. Then in the index page, I might have a link of code that looks something like this:

```
<a href="ricky.html">Click here to get to Ricky's other Page</a><br>
```

When the user clicks the link, the ricky.html file will be served to the user. The ricky.html page would be structured the same as index.html (ie head and body), but would have different content in the body. We won't add that line of code for this example, but I wanted to show you how you would use the anchor to link to pages together in a multi-page website.

This is my web page

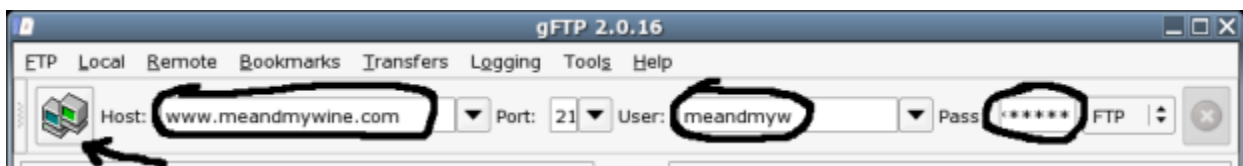
First Font Test  
Second Font Test  
Third Font Test  
Fourth Font Test  
Fifth Font Test  
Sixth Font Size

© 1998

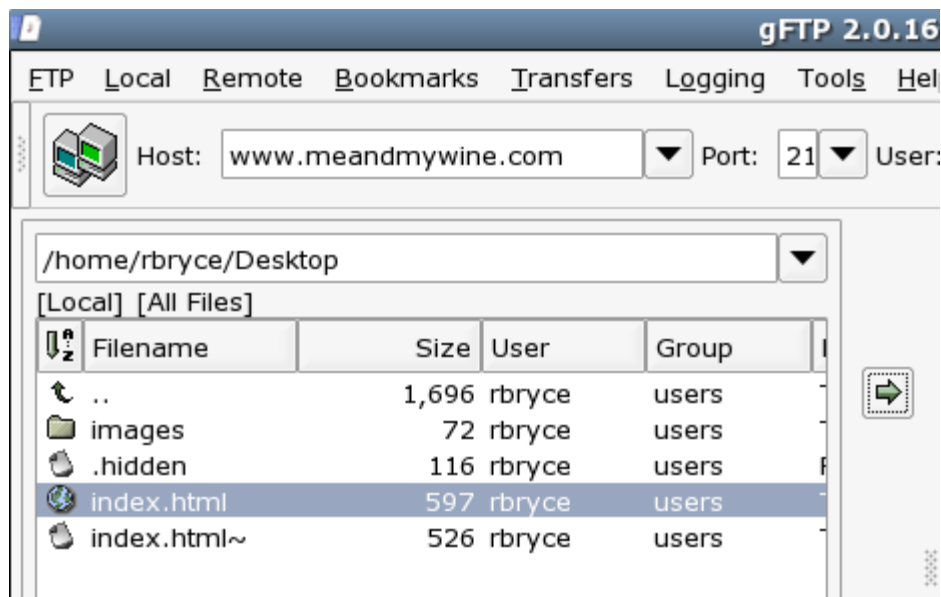
[Yahoo's Page](#)

- 18) **Upload your page and images** – Before you can upload your web page (and the images needed for it), you must first find a web server. There are free services available, but usually they will want to place an ad on your web page to cover costs related to hosting. Usually for under \$40/year, you can have your own domain name, plus a shared server to host the package. The name you choose must follow internet domain standards, and cannot already be registered by someone else. If you don't have a domain name or server yet, please contact me ([ricky@learnautomation.com](mailto:ricky@learnautomation.com)) and I will set you up! If you already have a domain name, and a server, then you're ready to go. All you have to do is upload the page using an FTP program.
1. If you are running Windows, you can use a free ftp client such as **SmartFTP**.
  2. If you run RedHat/CentOS, you can use **gftp** (Just go to Actions | Run Application, or Applications | Run application, and type **gftp** as the program you want to run.

- 19) In your FTP program, enter the name of the site, the username and password, that you were given when you signed up for your web server in step 18. Then press the connect button.

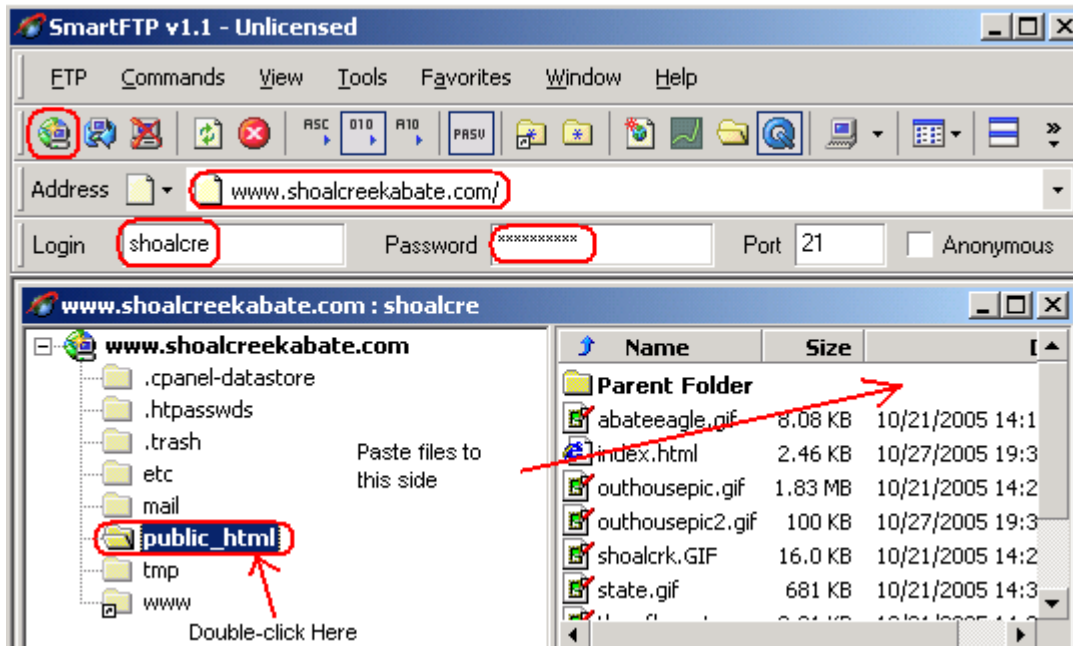


- 20) If you are using GFTP in Linux, Browse to the Desktop on your local system. This will be on the right side. **You will not need to do this step if you are using SmartFTP.** To get to the desktop, the only thing you will probably have to do is scroll down on the left window pane, and double click “Desktop” Your location in the local directory should be “/home/username/Desktop” as shown in the example below.

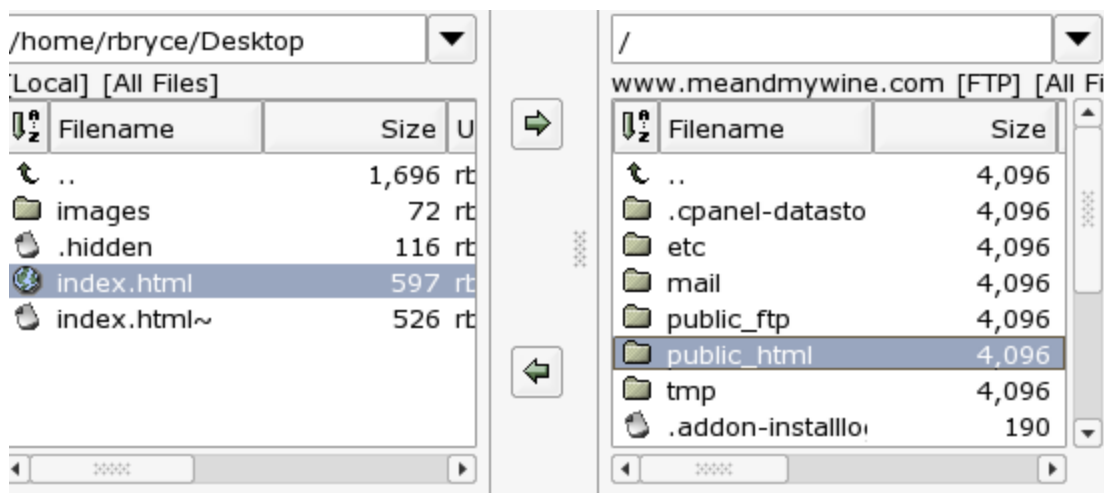


21) **Select the directory on the server, and upload the files** – Next we must select the directory on the web server that we will be uploading the files to. These files **MUST** be placed in the **public\_html** folder.

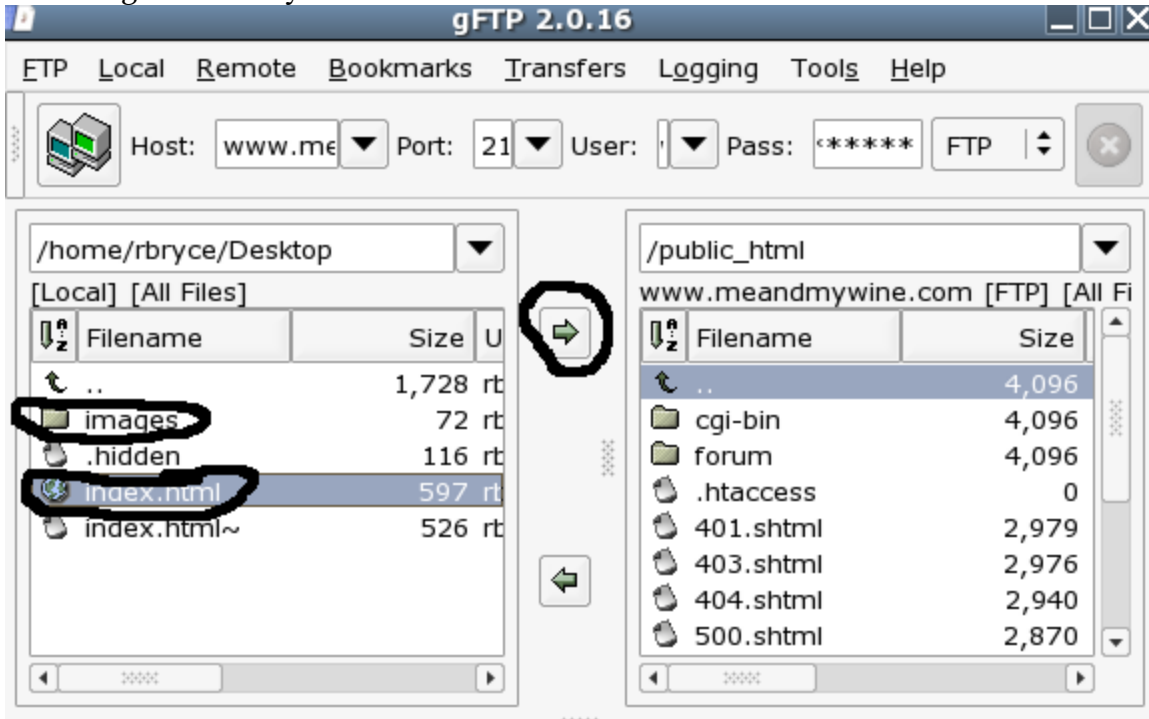
1. **If you are running SmartFTP**, just double click on the public\_html folder on the left side of your screen. You will see the contents of the public\_html folder appear on the right. Just paste the index.html file and the images folder into the right side of your screen.



2. **If you are running GFTP**, Double click the public\_html folder on the remote server.



- Next, click the index.html file in your local directory (left window pane), and click the right arrow (between the two panes) to move the file onto the web server. Do the same with the “images” directory.



- Your web page is now on the server, and you can access the page by opening a web browser, and visit [www.yourdomainname.com](http://www.yourdomainname.com).

**Miscellaneous other tags that might be useful:**

- Here are some other tags you may be interested in:

- `<b>This makes me BOLD when rendered</b>`
- `<i>I am italicized</i>`
- `<u>I will be underlined</u>`

- To change the color on the background, use the **bgcolor** parameter in the `<body>` tag:

```
<body bgcolor="green">
  <center>This is my web page</center>
```

25)

- To use an image as your background, use `background="images/mybackground.jpg"` instead of the `bgcolor` parameter in the `body` tag (your image must be in the `images` folder on the server):

```
<body background="images/mybackground.jpg">
```