

Project 2.1.4 HTML and CSS

Introduction

|  |  |
| --- | --- |
| You’ve learned that data gets transferred from one machine to another by following protocols. But what is the data? Why do webpages look different on different computers? What is a webpage, really?  W3C, the World Wide Web Consortium, is an organization dedicated to producing standards dictating the answers to many of these questions. |  |

Materials

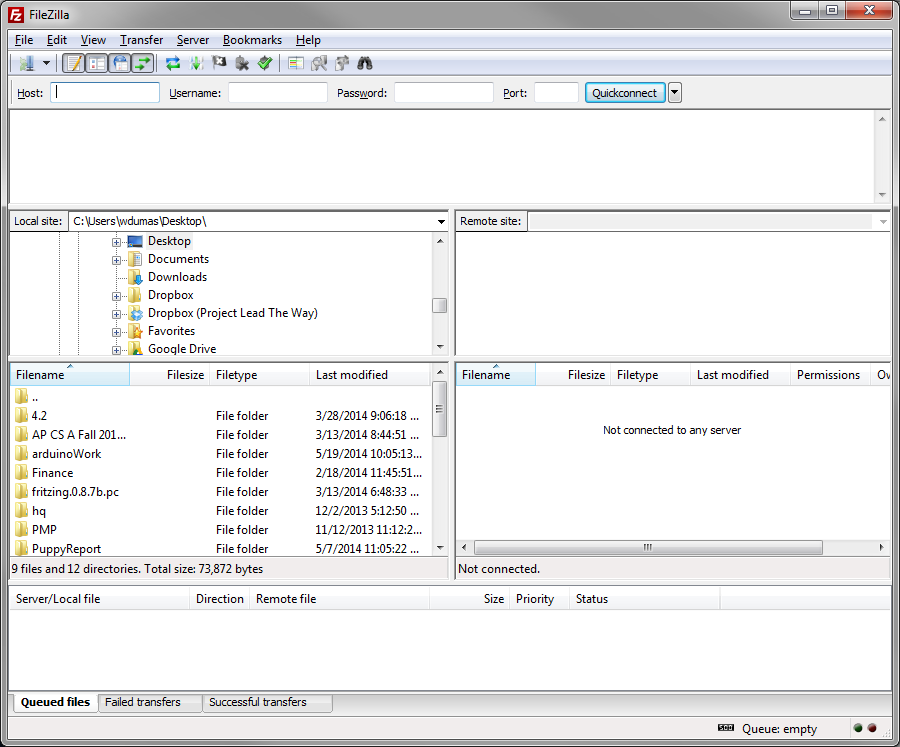
* A writing utensil
* Computer with Internet access and FileZilla installed
* Firefox web browser with FireSSH installed
* AWSS username/password combination and hostname

Procedure

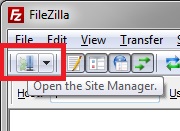
It is easy to create simple web sites with a **WYSIWYG** editor like Adobe DreamWeaver, for example, without having to write code. WYSIWYG stands for “What You See Is What You Get.” Most professional web developers, however, need to work more directly with the underlying code to achieve the exact features that clients want. Today you will learn what is happening behind the scenes when webpages are created and accessed. The webpages you create will be simple; we are not aiming for elegant. However, the *way* the webpages *work* will be exposed to you so you can see what the code does. If you wish to develop more elegant websites using a web design application, your knowledge of HTML and CSS code will give you much more power to customize your site and fix problems.

**Part I: Uploading a Webpage**

1. Meet or greet your partner for this activity and establish team norms. Make sure you have a system in place for version control. You will be uploading files to an **Amazon Web Services system**, which means you will be able to download them as long as you have access to a computer with an **FTP** client. This makes it all the more important that you have a system in place for determining what the most recent version of a particular file is. FTP stands for File Transfer Protocol, a protocol that dictates how files are sent over the Internet. An Amazon Web Services system, AWSS, is a part of a physical server’s hard drive that is dedicated to serving files for a particular domain. Many such virtual servers may coexist on a single machine.
2. For this activity you will use FileZilla, which uses SFTP to transfer files between your local machine and the AWSS. Open the FileZilla application as directed by your instructor. You will see a window like the one shown below.

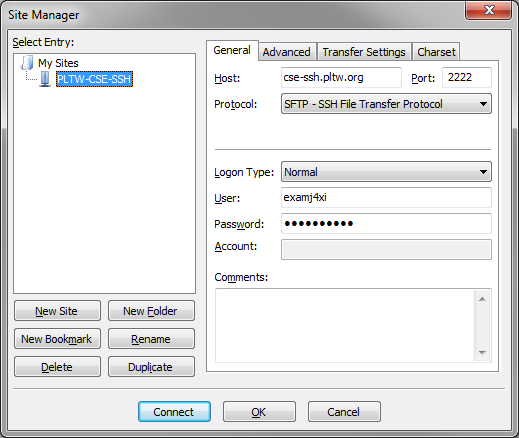


1. Open the Site Manager by clicking the icon shown below.

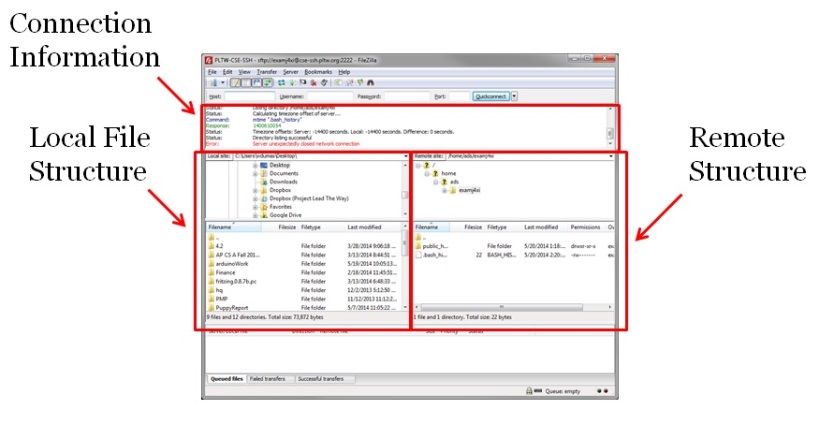


1. Click the New Site button and enter your credentials as directed by your instructor. Note that the sample images shown use a fake student account: examj4xi.

* The User and Password values here should be the same as the ones you use for FireSSH.
* Port should be set to 2222.
* Host should be set to cse-ssh.pltw.org.
* Switch Protocol to SFTP.
* When you have entered all of the values, click the Connect button.



1. The Site Manager window should close, and a connection will be established. There are several different regions of the FileZilla interface. The Connection Information window will communicate the status of your connection to the AWSS. The Local File Structure region shows a tree view of the directory structure on your computer as well as the files and folders in the current working directory on that machine. The Remote File Structure shows the directory structure on the AWSS as a tree as well as the contents of the current working directory. By default the current working directory on the AWSS will be your home directory, though for most purposes you will be transferring files to the public\_html folder within that directory.



1. We’ll now verify that your FTP client is working by uploading a sample webpage written in Hypertext Markup Language (HTML). HTML is the basic language for web content.

Now you’ll need to create the document. Follow these steps:

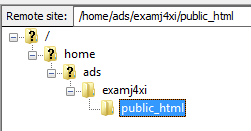
* + Paste the following code into your text editor.
  + Save the file in a location as directed by your instructor with the name index.html. That file named index.html is the file that a web browser will try to open by default if no other name is specified by the client.
  + Replace the name “Jane Jones” with your own name. Save this file in a commonly agreed upon location on your school computer as directed by your teacher.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <!doctype html>  <html lang="en">  <head>  <meta charset="utf-8">  <title>Jane Jones' Website</title>  </head>  <body>  <h1>  This is easy and fun!  </h1>  </body>  </html> |

1. Transfer this file to your home directory on the server as explained below so that you will be able to view it on the Internet from any networked computer in the world.
   * You may need to refresh the file structures in FileZilla by clicking the refresh button shown below:



* + In the FileZilla Local File Structure frame on the left, ensure that your local machine is showing and navigate to the index.html file.
  + Navigate to your public\_html directory on the AWSS by expanding your home directory (clicking the plus sign next to it) and then clicking on public\_html.

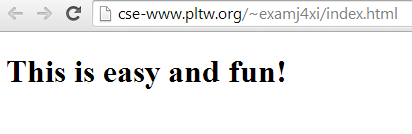


* + Either double-click on the index.html file or click and drag it from its location on your local machine into the public\_html folder on the remote machine.

1. Congratulations! You're published! Anyone in the world can now see your work. Your teacher will provide guidelines about what NOT to publish. Summarize them here.
2. In order to ensure that your upload has worked, use your browser to navigate to your newly published page. The URL path is cse-www.pltw.org followed by a slash and then a ~ and your student AWSS account name, followed by another slash and index.html.

For example:  cse-www.pltw.org/~studentname/index.html

You should see a webpage that looks like the following:



**Part II: Content and Style**

**Note:** If you are unable to use FTP to complete your file uploads, you can simply double-click on the icon for the HTML file on your local machine in order to open it in a browser.

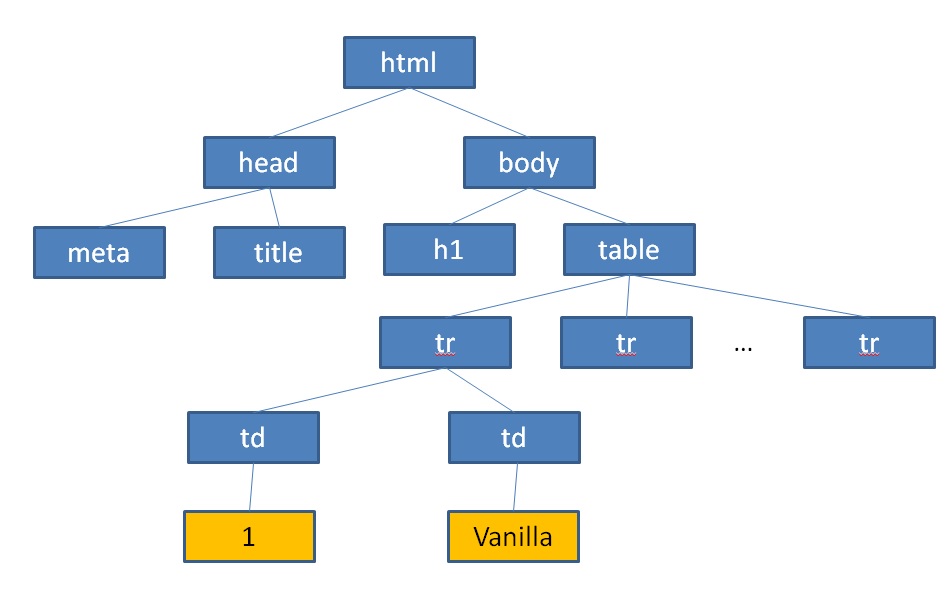
1. The code that you provide in an HTML document specifies what content will be displayed by the web browser. It contains text and possibly references to external files like images or music.

The tags you may use in HTML provide limited stylistic functionality. Copy the following HTML into your text editor and save the file as ice.html. Upload it using FTP. Use what you observe when you view that file as a webpage online along with 2.1 ReferenceCardUNIX-HTML-CSS to explain what each of the following tags do:

|  |  |  |  |
| --- | --- | --- | --- |
| **h1** | **table** | **tr** | **td** |
| Defines HTML headings, most important with #1. | Defines a table element. | Defines a table row element. | Defines a table cell element. |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <!DOCTYPE html>  <html ><head><meta charset=utf-8" /><title>I Scream For Ice Cream!</title></head>  <body>  <h1>Most Popular Ice Cream Flavors</h1>  <table><tr><th>Rank</th><th>Flavor</th></tr>  <tr><td>1</td><td>Vanilla</td></tr>  <tr><td>2</td><td>Strawberry</td></tr>  <tr><td>3</td><td>Chocolate</td></tr>  <tr><td>4</td><td>Cookies and Cream</td></tr>  <tr><td>5</td><td>Mint Chocolate Chip</td></tr></table>  </body>  </html> |

1. The structure of a webpage can be viewed as a tree. The root node is the <html> tag, which has two children: the <head> tag and the <body> tag. The <head> tag generally contains information about the page, whereas the <body> tag contains content. In the tree diagram shown below, is “Vanilla” part of the <head> or part of the <body>?

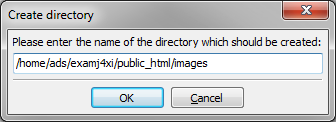
****

1. Having a working knowledge of the directory structure of your website will allow you to organize the files that you use to create it. Find an image of cookies online and save it as “cookies.jpg”. Upload it to your home directory on the AWSS using FTP. You can display the cookie picture on your webpage by adding this line of code within the <head> element.

<img src=”cookies.jpg” />

Save it, upload it, and view it.

1. In FileZilla, right-click on the public\_html folder on the remote server and select **Create directory**. It will bring up a window like the one below. Replace "New directory" with "images".



Move the cookies.jpg file into this new directory by clicking and dragging the image file into the images directory in the Remote File Structure.

What happens when you reload your webpage in the browser?

The image doesn’t load, gives broken image link square

1. The <img> tag has an alt attribute that will display text in the event that a link is broken, or in case it is needed for text-to-speech for use by web surfers with visual impairments. Modify your “img” tag from the previous step to look like this:  <img src=”cookies.jpg” alt=”Picture of Cookies” /> . Save it, upload it using FireFTP, and direct another Firefox tab to the appropriate URL. Because you have viewed this page before, it has been stored in your cache. Your browser will retrieve the old copy from that location instead of the updated copy unless you refresh it. What changes?

The broken picture box is replaced with the text “Picture of Cookies”

1. You have a broken link on your page because the cookies image isn’t in the location that the HTML has recorded. You can fix this by specifying the path to that image resource. There are two methods to specify a path.
   * A **relative** **path** tells the browser to ask for a location starting with the path of the file currently being rendered. Since ice.html is in your home directory, you could modify your tag like this  <img src=”images/cookies.jpg” alt=”Picture of Cookies” />  to specify how to get to the image.
   * An **absolute** **path** to a resource starts from the root of the web server. An image tag like this one will substitute path details to your own directory and allow the image to render in the browser <img src=”/home/examj4xi/public\_html/images/cookies.jpg” alt=”Picture of Cookies” /> . The previous absolute path given as an example above is absolute with respect to the directory structure on the server. Sometimes absolute path refers to the path with respect to the web serving root. In this case the path would read cse-www.pltw.org/~examj4xi/images/cookies.jpg.

Decide with your partner which version to use (absolute or relative). Explain when you would use the other version. Try that solution by modifying the HTML, saving, uploading, and refreshing the page in your browser.

Absolute path is needed if the (working directory/html file) isn’t the highest directory and therefore not on the same path.

1. Create a new file in your text editor called icecream.css. Copy the following Cascading Style Sheet (CSS) code, paste it into your file, and save the file. Style sheets let you apply a variety of visual effects such as font, boldness, center alignment, or background color to your webpage. The standards and specifications for using CSS are maintained by W3C ®, the World Wide Web Consortium.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | body{  font-family: "Marker Felt","Comic Sans MS", fantasy;  color:#003366;  }  h1 {  font-size: 1.3em;  text-align:center;  }  table {  margin-left:auto;  margin-right:auto;  text-align:left;  border-collapse: collapse;  cellspacing: 0 px;  }  td {  border: 1px solid #ffffff;  text-align:center;  background-color:#9FB6CD;  }  th {  text-align:center;  color:#ffffff;  background-color:#003366;  } |

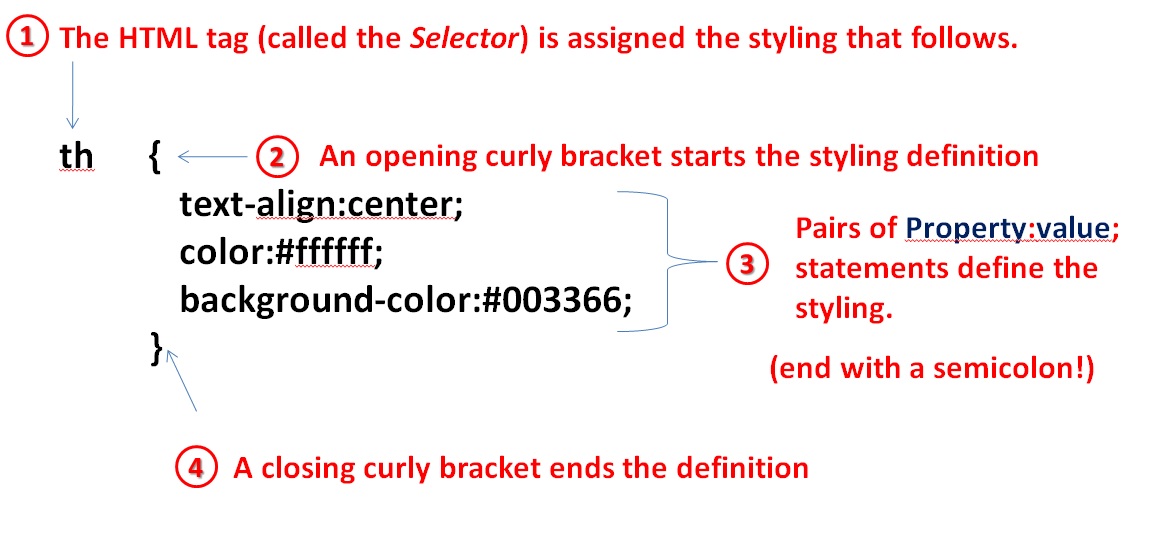
An **external style** sheet is a file that exists independently of your HTML document that contains only style information. In order for this external style sheet to influence the appearance of your website, you will need to reference it within the <head> tag of your HTML document. This type of styling is referred to as a linked style sheet. Paste the following code into your ice.html file within the <head> element and save that file.

 <link href=”icecream.css” rel=”stylesheet” type=”text/css” />

Upload both ice.html and icecream.css to the server via FTP and reload the ice.html page in your browser. What changes in appearance do you notice?

Table is centered and has a dark blue theme

1. The following is a diagram of the syntax of a single style definition in CSS. The source code appears on lines 25-29 of icecream.css shown above.



The **selector** tells CSS which part of the HTML should be stylized. There are several different **properties** of any given selector which you can use CSS to modify. This particular definition modifies the text alignment, color, and background color of cells within a table. A **value** must be specified for each property. What does line 8 in step 16 tell us about how <h1> tags will appear on this webpage?

Anything inside h1 will be centered on the page

1. Linked style sheets are a good way to set up rules for how different parts of your entire webpage will look. There are two other ways of incorporating CSS that allow you to set up rules for specific parts of your page. The “cascading” part of Cascading Style Sheets refers to more general definitions being overridden by definitions that are more specific to the context that you’re focusing on. The external style sheet may be referenced by dozens or even hundreds of pages. You can use an **internal style sheet** within the <head> element of your page to specify styles just for this document. Try pasting the following into your ice.html file within the <head> element.

|  |  |
| --- | --- |
| X1  X2  X3 | <style>  td {color:#FF0000;}  </style> |

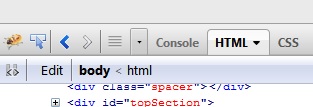
Save the changes, upload using FTP, and refresh your view of the page in your browser. What has changed?

The text showing the results of this person’s opinion in the table has changed to red (table data)

Internal style sheets give you the power to control the contents of a single document. You can set up a rule for an even more specific part of a document using an **inline style**. Inline styles are contained entirely within a single tag. Choose one <tr> tag in your ice.html file to change from this:  <tr>  to this:  <tr style=”color:#00FF00;background-color:sienna;”>  . What changes when you save, upload, and refresh the resulting file?

It changes color of text and background color for the row

1. We will use Firebug to help you understand which HTML and CSS components cause different parts of a page to be rendered differently. Follow these steps and then answer the questions below.
   1. View your page in Firefox and turn on Firebug.
   2. Switch back and forth between the HTML tab and the CSS tabs shown below to answer the questions that follow.
   3. You can mouse over a line of code in Firebug and it will highlight the visual region within the browser window that HTML code is affecting.
   4. Mousing over CSS code will show you a visual representation of the property that is being impacted when possible.





1. What line of code in the HTML links to the external style sheet so that sections 1, 2, and 3 have more visual appeal than they would if this were created using plain HTML? (Hint: Reference Step 16 if needed.)

<link type="text/css" rel="stylesheet" href="icecream.css"></link>

1. Which property in the CSS file dictates the background color of the cells in section 2 shown above?

th {

text-align:center;

color:#ffffff;

background-color:#003366;

}

1. What line of HTML code creates the row of cells in section 3?

<tr><td>3</td><td>Chocolate</td></tr>

1. What is the tag responsible for making the text in section 1 appear larger than the text in the other sections?

<h1></h1>

**Part III: Linux Commands**

1. Like most webpages, your webpage is being served from a machine running an operating system in the UNIX® family. Knowing a handful of UNIX commands can be *very* handy since the majority of processing power runs under UNIX-family operating systems. The purpose of this part of the activity is to show you how to move around on a UNIX server. This is useful now because it allows you to take a look at your webpage *without* using HTTP. It may also be useful to you in the future because it will help you interact with computers when you do not have access to GUI tools, or when the GUI tools that you do have access to don't have sufficient privileges to accomplish what you want.

The web server program on the AWSS is called Apache. Apache configuration files determine which files on the server correspond to which paths specified in a URL. The directory structure on the server is mostly hidden from clients making web requests.

Why wouldn't you want web client software to see the server's entire directory structure?

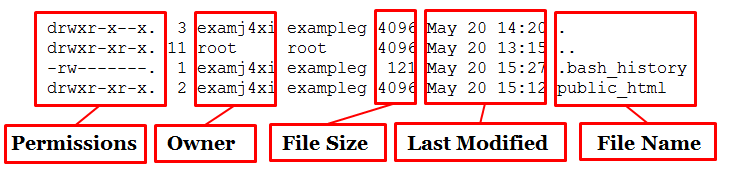
1. Log in to your FireSSH account as in the previous activity and type pwd  in the command line.  pwd  stands for “print working directory.” It shows you the path through the directory structure to your current location. This path leads to your home within the AWSS since you have not yet navigated away from it.

Record the output here.

1. Enter the command to list information about files  ls  next. Write down two of the file names here.
2. Linux commands often offer options to provide additional functionality. You can quickly learn about the options available to various Linux commands using the  man  feature, short for "manual." Linux uses **flags** to provide this additional functionality. When you type a command, you follow it with a space, a hyphen, and then a letter or sequence of letters. One particularly useful set of options for the  ls  command is the  -la  flag combination. The whole command would be entered like this:  ls -la . Type  man ls  at the command line in order to bring up the manual pages for  ls  . The manual is too long to fit on one screen, so the Linux operating system is **paginating** it. You can use Ctrl+f to move a page forward or Ctrl+b to move a page backward. Read the manual page to find out what the  -l  and  -a  flags do. Record your findings below. When you are done, press the q key to quit the paginator.

|  |  |
| --- | --- |
| -l |  |
| -a |  |

1. When you enter ls -la  , you should see a printout like the one shown below in your terminal.



Who are the owners of the files in your home directory?

1. The first two directories listed in your home directory should be  ./  and  ../  . The single dot always refers to the current directory, and the double dot always refers to the parent of the current working directory. The parent is the directory one level above your working directory in the file structure. You will use the  cd  command to navigate the directory structure.  cd  stands for “change directory” and must be followed by the name of a directory accessible from the current working directory or the full path to another directory such as  /home/  . Type  cd ..  to navigate to the parent of your current directory. What command can you use to display the path for the directory you are currently in? Record that path here:
2. Use the appropriate command to list the contents of your current directory. What do you think the file names represent?
3. Spend a moment exploring the directory structure using the commands that you have learned. Then navigate to  /home/ads/ . Each user has a directory here. Write down the name of one account's directory other than your own or that of your teacher.
4. For the sake of convenience, we use a text editor like Notepad++ with a graphical user interface most of the time. There are, however, some situations in which a professional might not have access to such an editor. For example, when a system administrator sets up a server, typically the only text editor available to them initially to modify vital system files is a program called vi. We will use this editor now to examine the contents of your index.html file.

Type  vi index.html  at the command line. vi provides syntax highlighting to help you quickly identify different parts of your program. In HTML, **tags** (defined in the next section) appear in orange. Name one tag that you can see:

To get out of vi, type  :x . This will automatically save any changes that you make. Another popular text-only editor is Emacs. It is installed on the AWSS. You may decide to use it if you wish.

**Part IV: A Static Webpage**

A static webpage is one that does not change in response to user interactions. In a later lesson, we will create webpages that allow for user interaction. For now you will modify the static webpage that you’ve been working with in this activity. Complete 2 - 4 of the following tasks as directed by your teacher:

1. Create a second page that references icecream.css but looks different than ice.html.
2. Use an internal style sheet to change the background of a whole page to an image of your choice.
3. Use an inline style to make one row in a table appear different than others.
4. Step 17 points out properties and values in the CSS. Research five additional properties and their corresponding values and incorporate them into a page through some combination of external, internal, and inline styles.
5. Use the accompanying 2.1 ReferenceCardUNIX-HTML-CSS and the Internet as necessary to help you create your own multi-page website.

Conclusion

1. What sort of visual options are available to you when deciding how you want your website to appear?
2. What reasons can you imagine for W3C requiring that there be only one <head> tag and one <body> tag in a given HTML document?
3. The <img> tag allows you to provide alternate text to help you reach a wider audience. Give two examples of design decisions that would limit the audience of a webpage (i.e., style interfering with content).