PLTW COMPUTER SCIENCE

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 web pages look different on different computers? What is a web page, really?

The World Wide Web Consortium, W3C, is an organization dedicated to producing standards dictating the answers to many of these questions.

Materials

- Computer with web browser
- Cloud9 credentials and workspace

Resources

2.1.4 sourceFiles.zip

2.1 Reference Card UNIX-HTML-CSS.docx

Procedure

It is easy to create simple websites 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 web pages are created and accessed. The web pages you create will be simple; we are not aiming for elegant. However, the *way* the web pages *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: Publish a Web Page

1. Meet or greet your partner for this activity and establish team norms.

   For this activity, you will try another method of collaboration than pair programming. You will be working in a shared workspace that permits four-hands programming. Like pair programming, the partners will work together to produce a single artifact. Unlike pair programming, in the collaboration method for this activity, both partners can use their own

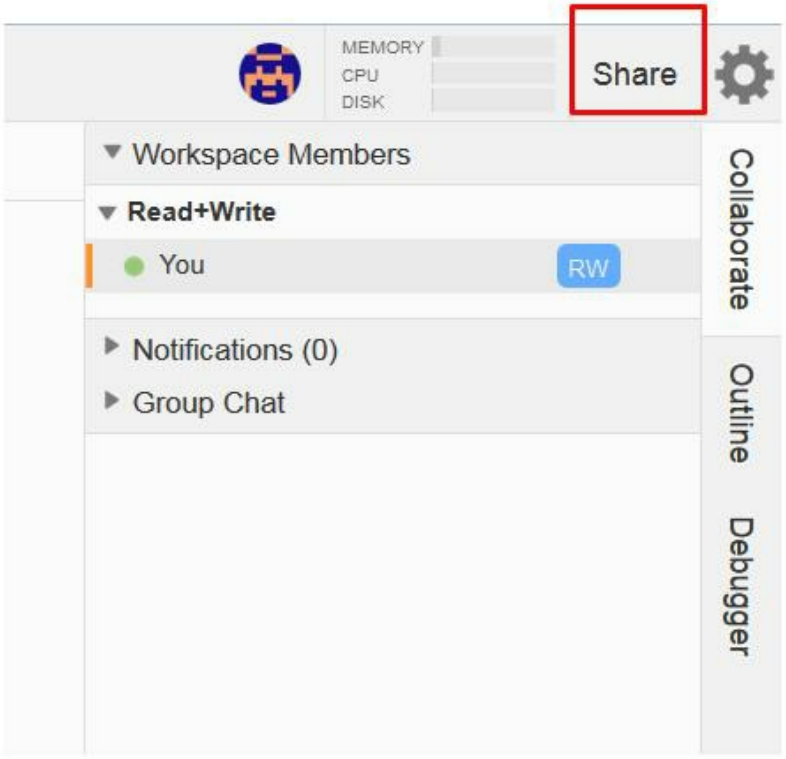
keyboard and mouse to interact with a shared environment. You should still take advantage of being seated together by sketching on paper, gesturing to code on each other's screens, and making eye contact during conversation, but you can use this virtual space to work together as well.

Each workspace is owned by exactly one Cloud9 user, but that user can share the workspace with other users.

2. Log in to Cloud9 at https://c9.io. Each partner should do this.
3. From your dashboard, open the workspace you created in the last activity. Recall that each workspace is its own virtual machine with the Linux operating system. Each partner should do this.

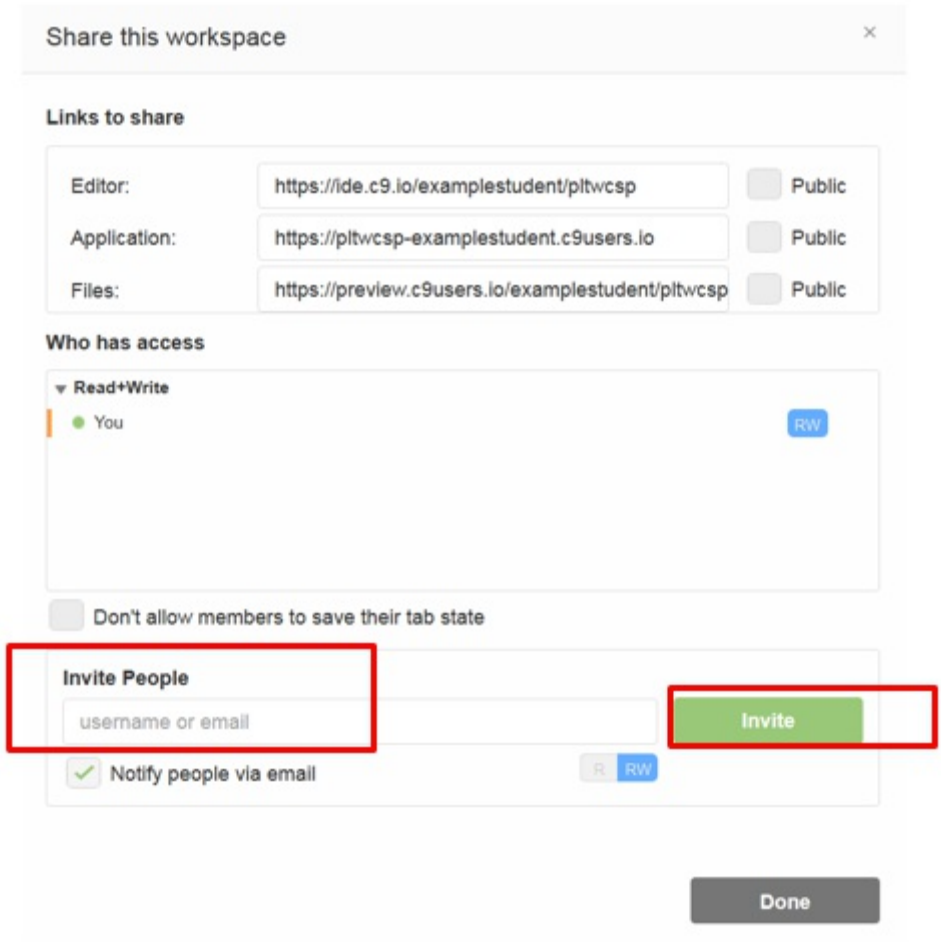
   Each person can develop their own website, in their own workspace. However, the partnership can also develop a single workspace together in a shared environment that gives both partners access. Even if you intend to complete the assignment using two separate workspaces, begin the activity using a single shared workspace, so that you can experience this unique Google-doc-like environment for coding. Pick one person who will technically “own” the workspace that you share.

4. Share the owner's workspace with the other user as follows.
   - In the upper right of the workspace window, select **Share**.



   - Type the Cloud9 username to share the project.
   - A sharing invitation can have permissions to read (R) or to read and write (RW). Leave the permissions on the default RW (

) and select **Invite**.

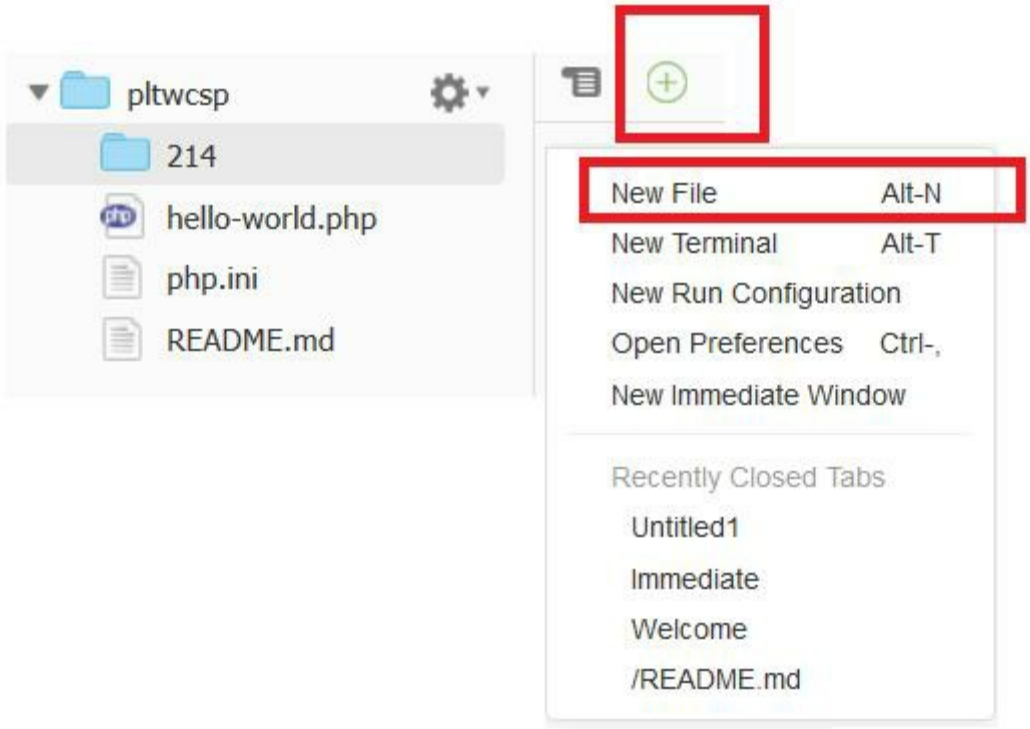


The Cloud9 environment can also be a useful tool for your teacher; the teacher can circulate among students' workspaces using different tabs in their own browser.

- If directed by your teacher, enter your teacher's Cloud9 username and select **Invite**.
- To access a workspace that has been shared with you, go to your dashboard. If you have already entered one of your workspaces, you can access the dashboard by selecting **Cloud9** > **Go to Your Dashboard**. The dashboard will open in another browser tab.
- Select **Shared With Me** from the left-hand navigation bar on the dashboard. Open the workspace that your partner has shared with you. It can take a minute or two for a workspace to appear after it has been shared with you, but you can proceed with these instructions while waiting.

You will create a document containing Hypertext Markup Language (HTML) for a simple web page. HTML is the basic language for web content. You can avoid publishing content if you prefer. To keep content unpublished, do not start the Apache server on Cloud9.

5. Create a new document as follows.
   - Select the circled plus sign to the right of a Cloud9 tab and select **New File**.



- Type or paste the following text into the editor.

```
<!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>
```

  Unlike the *Python* interpreter reading a *Python* file, a browser reading an HTML file does not care much about white space (new lines, tabs, spaces). The indentation shown in the HTML above makes it easier for a human to see how the code is structured, but the indentation does not affect the rendering by a browser.

  Web servers are usually configured to serve the file named `index.html`, if no other name is specified by the client.

- Save the file in your `214` folder with the name `index.html`.

  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.

6. Professional web developers do not develop on a live site. They work on a private site and publish once work is complete.

   To view an unpublished client-side web page (i.e., HTML, CSS, JavaScript, and audiovisual media) as though it had been published, you have two options:

   - Save the file(s) on your local machine and then open the HTML file in a browser. These files can be edited with Notepad++ or another **text editor**, which allows you to edit text characters without adding rich formatting data.
   - Upload to or create the file(s) in Cloud9 and use the preview feature without running the Apache web server. The Cloud9 preview feature **emulates**, or mimics, a wide range of browsers. To view the rendered file, select the file in the Cloud9 environment, select **Preview**, and then select the desktop or mobile browser to be emulated.

   In the next lesson, you will learn to create web pages that include **server-side features** in which code executes on the server. To get comfortable with Cloud9 and Apache, this lesson recommends that you publish content on Cloud9 and view rendered files by loading or refreshing a browser's request to the web server.

Part II: Content and Style

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.

7. The tags you may use in HTML provide limited stylistic functionality. Copy the following HTML into a new Cloud9 file and save the file as `ice.html`.

```
<!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>
```

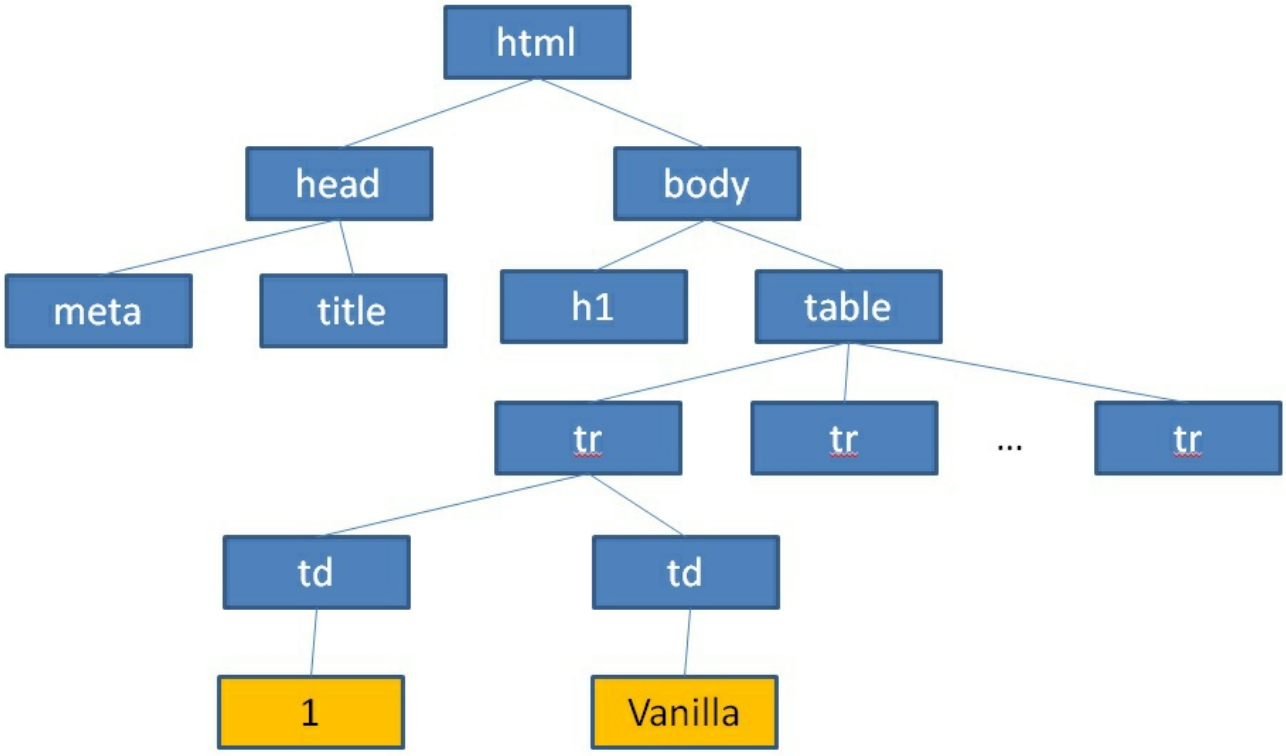
8. Use what you observe when you view that file as a web page online along with the resource *Lesson 2.1 Reference Card UNIX-HTML-CSS* to explain what each of the following

tags do:

`<h1>`
`<table>`
`<tr>`
`<td>`

The structure of a web page 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>`?



9. Add an image to your page as follows.
   - Find an image of cookies online and save it as `cookies.jpg`.
   - Upload it to your Cloud9 workspace by selecting **File** > **Upload Local Files** or by dragging an icon of the file and dropping it onto the Cloud9 file tree viewer.

     Most web servers allow you to upload files using the File Transfer protocol (FTP). What you accomplished in this step using HTTP is equivalent to what is usually accomplished with FTP. Name another real-life example in which two different protocols for computer

or human behavior can accomplish similar effects.

- You can display the cookie picture on your web page by adding this line of code within the `<body>` **element**. An element is one object with the document, usually started and ended by an HTML tag.

  ```
  <img src="cookies.jpg" />
  ```

- Save the modified HTML and refresh the view in the browser. You should see your page with an image of cookies.

A good website has well organized files. Understanding the directory structure of the web server will allow you to create a site with well-organized files.

10. In the Cloud9 workspace, create a directory named `images` within your `214` folder.
11. Drag or otherwise move the cookies.jpg file into the images folder.

    What happens when you reload your web page in the browser?

12. The `img` tag has an `alt` attribute that will display text if the image cannot be loaded or displayed. That `alt` text can be spoken by a screen reader for a web user with a vision impairment.
    - Modify your `img` tag from the previous step to look like this: `<img src="cookies.jpg" alt="Picture of Cookies" />`.
    - Save the new HTML and refresh the browser tab in which you were rendering the page.
    - What changes?

    You have a broken link on your page because the cookies image isn't in the location that the HTML has recorded.

13. 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. Because “images” is a folder within the directory containing `ice.html`, you could refer to the image using `<img src="images/cookies.jpg" alt="Picture of Cookies" />`, where `src` assumes the current document's directory as the place to find the folder `images`.
    - An **absolute path** to a resource starts from the root of the web server and includes the domain name. You can refer to the image as `<img src="https://workspacename-c9username.c9users.io/214/images/cookies.jpg" alt="Picture of Cookies" />`. You will need to use your own Cloud9 workspace name and Cloud9 username in the domain name.

14. Decide with your partner whether to use absolute or relative references to load your image. Explain what you think the advantages might be for each approach.
15. Try the solution of your choice by modifying the HTML, saving, and refreshing the rendered page in your browser.
16. Create a new file in your Cloud9 workspace 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 web page. The standards and specifications for using CSS are maintained by

W3C ®, the World Wide Web Consortium.

```
 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;
}

tr   {
       border: 1px solid #ffffff;
       text-align: center;
       background-color:#9FB6CD;
}

th   {
       text-align: center;
       color: #ffffff;
       background-color: #003366;
}
```

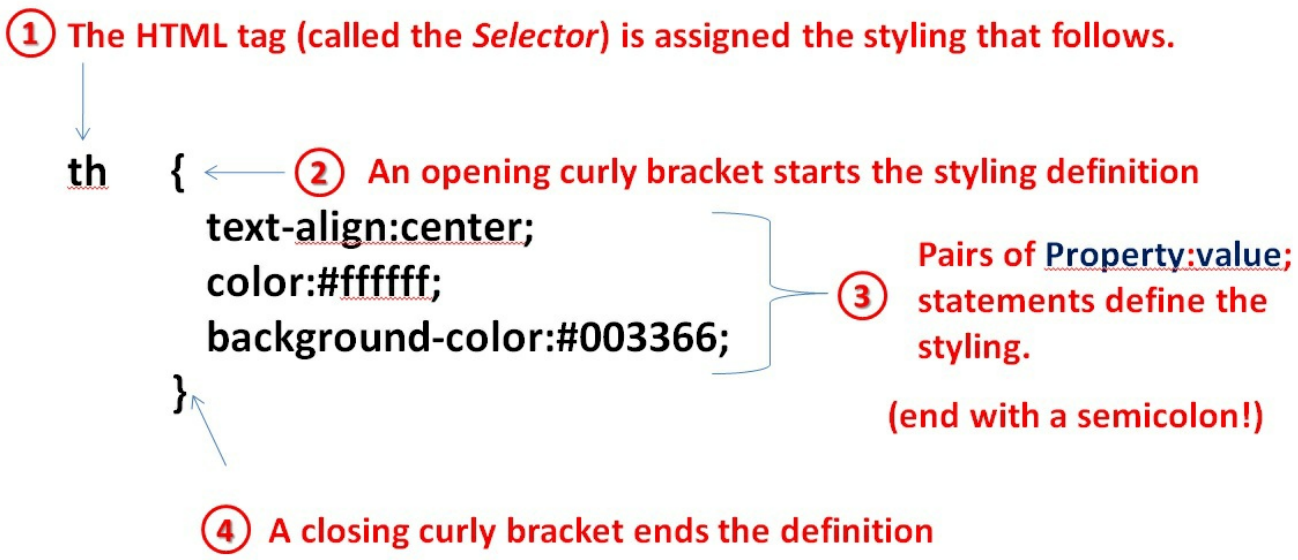
An **external style sheet** is a file that exists independently of your HTML document and contains only style information. For this external style sheet to influence the appearance of your website, you will need to reference it within the `<head>` tag of your HTML document. This type of styling is referred to as a linked style sheet.

17. 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">
```

18. Save both the `ice.html` and `icecream.css` files and reload the `ice.html` page in your browser.
What changes in appearance do you notice?

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 web page?

Linked style sheets are a good way to set up rules for how different parts of your entire web page 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.

19. Paste the following into your `ice.html` file within the `<head>` element.

```
<style>
  td {color: #FF0000;}
 </style>
```

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

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.

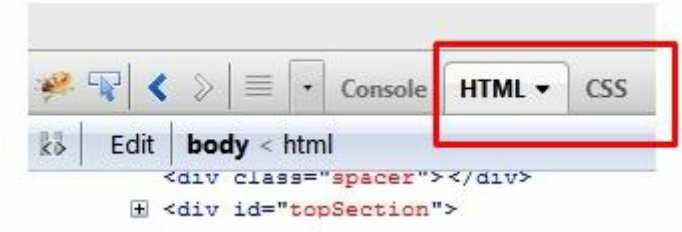
20. Choose one `<td>` tag in your `ice.html` file to change from this: `<td>` to this: `<td style="color:#00FF00; background-color:sienna;">`.

What changes when you save the file and refresh the rendered tab in the browser?

We will use Firebug to help you understand which HTML and CSS components cause

different parts of a page to be rendered differently. We'll use your ice cream page.

21. Follow these steps and then answer the questions below.
    - View your ice cream page in Firefox by typing the address in the location bar. Turn on Firebug.
    - Switch back and forth between the HTML and CSS tabs to answer the questions that follow.



    - You can mouse over a line of code in Firebug and it will highlight the visual region within the browser window that particular HTML code is affecting.

      Mousing over CSS code will show you a visual representation of the property that is being impacted when, if the element is visible.

      Mouse over some CSS to answer the following questions.

    - 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?
    - Which property in the CSS file dictates the background color of the cells in the row containing “Rank” and “Flavor”?
    - What line of HTML code creates the row of cells in section 3?
    - What is the tag responsible for making the text in section 1 appear larger than the text in the other sections?

Part III: Navigate around the server with Linux commands

Your web page is being served from a machine running the Linux operating system, which is similar to all the other operating systems 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.

22. At the Bash command line, type `pwd`, which stands for “print working directory.” It shows you the path through the directory structure to your current location.

    Record the output here.

23. Enter the command to list information about files: `ls`.
24. Compare the output to the file browser that is on the left side of the Cloud9 workspace.

    The Cloud9 file browser only shows files within a specific directory on the Cloud9 machine: the directory you identified in the previous step. The Cloud9 file browser uses icons to indicate whether an item in a directory is another directory, an executable file, or a text file.

    How does the ls command indicate the file type?

    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`.

25. Type `man ls` at the command line to bring up the manual pages for `ls`.

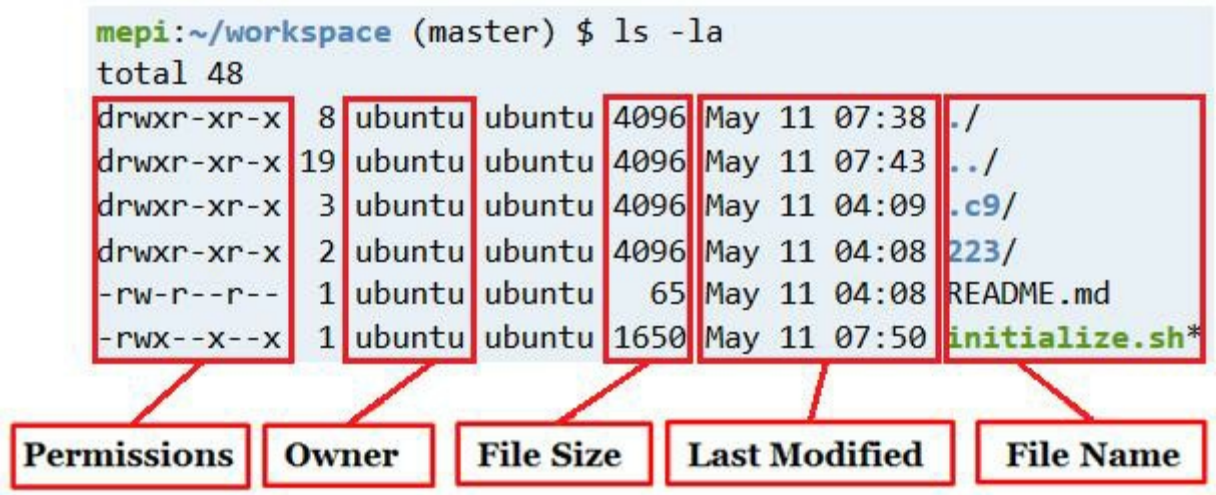
    The manual is too long to fit on one screen, so the Linux operating system **paginates** it. You can use `Ctrl+f` to move a page forward or `Ctrl+b` to move a page backward.

26. Read the manual page to find out what the `-l` and `-a` flags do. Record your findings below.

`-l`	
`-a`	

27. When you are done, press the q key to quit the paginator.

    Now try using both the `l` and `a` flags. When you enter `ls -la`, you should see output similar to that shown below in your terminal.



What user is the owner of the files in your home directory?

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. The command `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/`.

28. To navigate to the parent of your current directory, type `cd ..`

    What command can you use to display the path for the directory you are currently in? Record that path here:

29. Use the appropriate command to list the contents of your current directory.

    What do you think the file names represent?

30. 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.

    Learn how to exit tty editors (optional). We have been using the editor built into Cloud9. On a bare-bones machine, you might have to use the vi or emacs editors. A quick introduction to these editors can be helpful for later use.

31. Select the circle plus sign next to a Cloud9 tab and select **New Terminal**.
32. Type `vi index.html` at the Bash command line prompt.

    The program 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.

33. Name one tag that you can see.
34. Make a change in the editor as follows. The `i` key enters insertion mode, and the escape key

exits insertion mode. To save changes and exit vi, type : `x`.

Another popular editor is emacs; you exit emacs with `Ctrl+x Ctrl+c`. Both editors are installed on the Cloud9 server, but you are likely to prefer the ACE code editor that is built in with the environment. You can close the additional terminal tab you opened for this step.

Part IV: A Static Webpage

A static web page is one that does not change in response to user interactions. However, static web pages can provide user interaction by **hyperlinking** to each other. In the next lesson, you will learn how to use JavaScript to provide interactivity with the user within a single web page.

A hyperlink has the following syntax in HTML.

```
<a href="myfile">Text for the link</a>
```

The quoted filename can be a relative reference to a file on the same server as the HTML file containing this hyperlink, or it can be a full URL for any resource on the Web.

35. Create a second web page that says anything you want, as follows.
    - In the file browser at the left of the Cloud9 interface, right-click the `214/skeleton.html` file and select **Duplicate**. Change the name to something appropriate like `yourtopic.html`.
    - You can edit the file in an editor. You can also see a rendered view by either using the Cloud9 Preview mode to open the file or opening the file in your browser served by Apache as described in step 5. Make changes in the editor and save the modified file. Then refresh one of the rendered views of the file.
    - When you are happy with your new page, modify `ice.html` to link to your new web page.

    A static web page is one that does not change in response to user interactions. In a later lesson, we will create web pages that allow for user interaction. For now, you will modify the static web page that you've been working with in this activity.

36. Complete two to four of the following tasks as directed by your teacher:
    - Create a second page that references `icecream.css` but looks different than `ice.html`.
    - Use an internal style sheet to change the background of a whole page to an image of your choice.
    - Use an inline style to make one row in a table appear different than others.
    - Step 18 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.
    - Use the accompanying *Lesson 2.1 ReferenceCardUNIX-HTML-CSS* and the Internet as necessary to help you create your own multi-page website.

Practice Opportunity for the *Create* Performance Task

The Create Performance Task requires you to demonstrate a program that you created that can be executed. In some sense, HTML and CSS provide instructions to a browser's rendering engine, and those instructions are executed by the rendering engine. However, HTML and CSS are not programming languages, because they do not provide variables nor the `if-then` selection and `for` loop iteration required for most algorithms. With JavaScript learned in the next lesson, you could create an interactive web page for the Create Performance Task. Even with just HTML and CSS, we do use the iterative development and design process and can practice the following component of the Create Performance Task.

- *Describe the incremental and iterative development process you used, focusing on two distinct points in that process. Describe the difficulties and/or opportunities you encountered and how they were resolved or incorporated. (Adapted from Create Performance Task Part 2b.)*

Note: This direction is adapted from the official College Board Create Performance Task but does not duplicate the content of College Board Task or Rubric. The task provided here contains elements that are different than the College Board Performance Task and Rubric. Please reference official College Board materials.

Conclusion Questions

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 web page (i.e., style interfering with content).