|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Intro to CSS** |  |

|  |
| --- |
| **Your Tasks** |
| * Watch the video on CSS * Reference a CSS page from an HTML page * Have Ms. Pluska check off the above tasks * Create a CSS rule-set * Have Ms. Pluska check off the above tasks * Write code to implement the *text-size*, *font-family*, *font-weight,* *font-style,* and *text-align* properties * Receive credit for the group portion of this lab |

* **Watch the video on CSS**

CSS stands for Cascading Style Sheets. And, as the name implies it refers to sheets that can be applied to style our html code. Watch the video below to learn more.

<https://youtu.be/EP9QMdoXvXE>

* **Reference a CSS page from an HTML page**

Before we start styling our pages we must first create a css page. The css page is where you will write all your rules for styling your HTML page. As your website grows it is typical to have several css pages. In order for your HTML page to “talk” to your css styles you must reference your css page from your HTML page.

The reference to your CSS page occurs in the <head></head> section of your webpage. Below is an example of how the *Index.html* page in the *MyWebsite* directory could reference the *MyStyles.css* page in the *Styles* directory.

|  |  |
| --- | --- |
| **MyWebsite** | <!DOCTYPE html>  <html>  <head>  <title>My Website</title>  **<link rel = “stylesheet” type = “text/css” href = “Styles/MyStyles.css” />**  </head>  <body>  </body>  </html> |
| Index.html |
| **Styles** |
| MyStyles.css |

|  |  |
| --- | --- |
| Write code that could be used to reference the MyStyles.css page from the Index.html page | |
| **File Structure** | **Code** |
| |  | | --- | | **MyWebsite** | | Index.html  MyStyles.css | |  |
| |  | | --- | | **MyWebsite** | | Index.html   |  | | --- | | **Styles** | | |  |  | | --- | --- | | **AboutMe** | **OtherPages** | | MyStyles.css | OtherStyles.css | | | |  |
| |  | | --- | | **MyWebsite** | | |  |  | | --- | --- | | **AboutMe** | **Styles** | | Index.html | |  | | --- | | **AboutMe** | | MyStyles.css | | | |  |

* **Have Ms. Pluska check off the above tasks**



Before you continue have Ms. Pluska check off the above tasks

Do not continue until you have Ms. Pluska’s (or her designated TA’s) signature \_\_\_\_\_\_\_\_\_\_\_\_

* **Create a CSS rule set**

The block of code that gives rules for a particular tag is called a rule-set. CSS rule-sets consist of two main parts: the selector and the rules.

### **Selector**

The selectors can be any part of the web page you want to style. One way you can identify parts of the web page is using the names of the element type. Selecting an element type will make all elements of that type have the given styling. The selector name for HTML element types is the name of the tag with the brackets removed. In the below example the selector is h1 and it will style all the h1 elements with the rules inside the curly braces ({ }).

### **Rules**

The rules describe how the elements identified by the selector should change. Each rule consists of a property name and a value, separated by a colon (:). The property name describes what the rule is about, such as color or size, and the value how the property should change. For example, the rule-set below will make all the h1 headers on the page have blue text that is underlined.

|  |
| --- |
| **h1** is the selector  **color** is the property  **blue** is the value  **h1{**  **color**: **blue**;  **text-decoration**: **underline**;  **}** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Consider the html code below. Write rules using the *color* and *text-decoration* property to style each of the following elements: li, h1, h3, p. You may use any of the following values for the *color* and *text-decoration* properties.   |  |  | | --- | --- | | **Property** | **Possible values** | | text-decoration | over-line  line-through  underline  underline overline | | color | blue  black  yellow  purple  green  \*\*These are just a few... There are tons more | |
| <ul>  <li><a href="[#home](view-source:https://hpluska.github.io/APCompSciPrinciples/" \l "home)">home</a></li>  <li><a href="[#tedTalks](view-source:https://hpluska.github.io/APCompSciPrinciples/" \l "tedTalks)">Ted Talks</a></li>  <li><a href="[#knowledgeCelebrations](view-source:https://hpluska.github.io/APCompSciPrinciples/" \l "knowledgeCelebrations)">knowledge celebrations</a></li>  <li><a href="[#labs](view-source:https://hpluska.github.io/APCompSciPrinciples/" \l "labs)">labs</a></li>  </ul>  <h1 id=”home”>home</h1>  <h3>About this course</h3>  <p> Welcome to Computer Science and the endless opportunities it offers! </p>  <h1 id = “tedTalks”>Ted Talks</h1>  <h3>Weekly Talks</h3> |
|  |

* **Have Ms. Pluska check off the above tasks**



Before you continue have Ms. Pluska check off the above tasks

Do not continue until you have Ms. Pluska’s (or her designated TA’s) signature \_\_\_\_\_\_\_\_\_\_\_\_

* **Write code to implement the *text-size*, *font-family*, *font-weight*, *font-style*, and *text-align* properties**

In addition to the color and text-decoration properties, there are other properties that can be applied to style text. Each of these and the corresponding values are summarized below,

|  |  |  |
| --- | --- | --- |
| **Property** | **Possible values** | **Description** |
| text-decoration | over-line  line-through  underline  underline overline | Controls whether the text is underlined, overlined, or lined-through |
| color | Blue, black, yellow, purple, green, etc. | Specifies the color of the text |
| text-size | 1em, 1.5em, 2em, etc | 1em is the default text size. 2em is twices as large, 2.5 is 2.5 times as large, etc. |
| font-family | serif, sans-serif, cursive, fantasy, and monospace, etc | Controls the style of the text |
| font-weight | normal, bold, bolder, lighter | Sets the weight, or boldness, of the font |
| font-style | normal, italic | Specifies the font style for text |
| text-align | left, right, center | Sets the horizontal (side to side) position of the text within each line |

|  |  |
| --- | --- |
| Write a CSS rule-set that could be used to style the code as shown | |
| **Index.html** | **Output** |
| <!DOCTYPE html>  <html>  <head>  <link rel = "stylesheet" type = "text/css" href = "Styles/MyStyles.css" />  </head>  <body>  <h1> My Hobbies </h1>  <ol type="I">  <li>Skiing</li>  <p>Telemark</p>  <p>Cross country</p>    <li>Programming</li>  <p>HTML</p>  <p>CSS</p>  <p>javascript</p>  <li>Mountain Biking</li>  <p>Enduro</p>  <p>Cross Country</p>  <p>Downhill</p>  </ol>  </body>  </html> | This text is blue  This text is bold and red |
| **MyStyles.css** | |
|  | |

* **Receive Credit for the group portion of this lab**



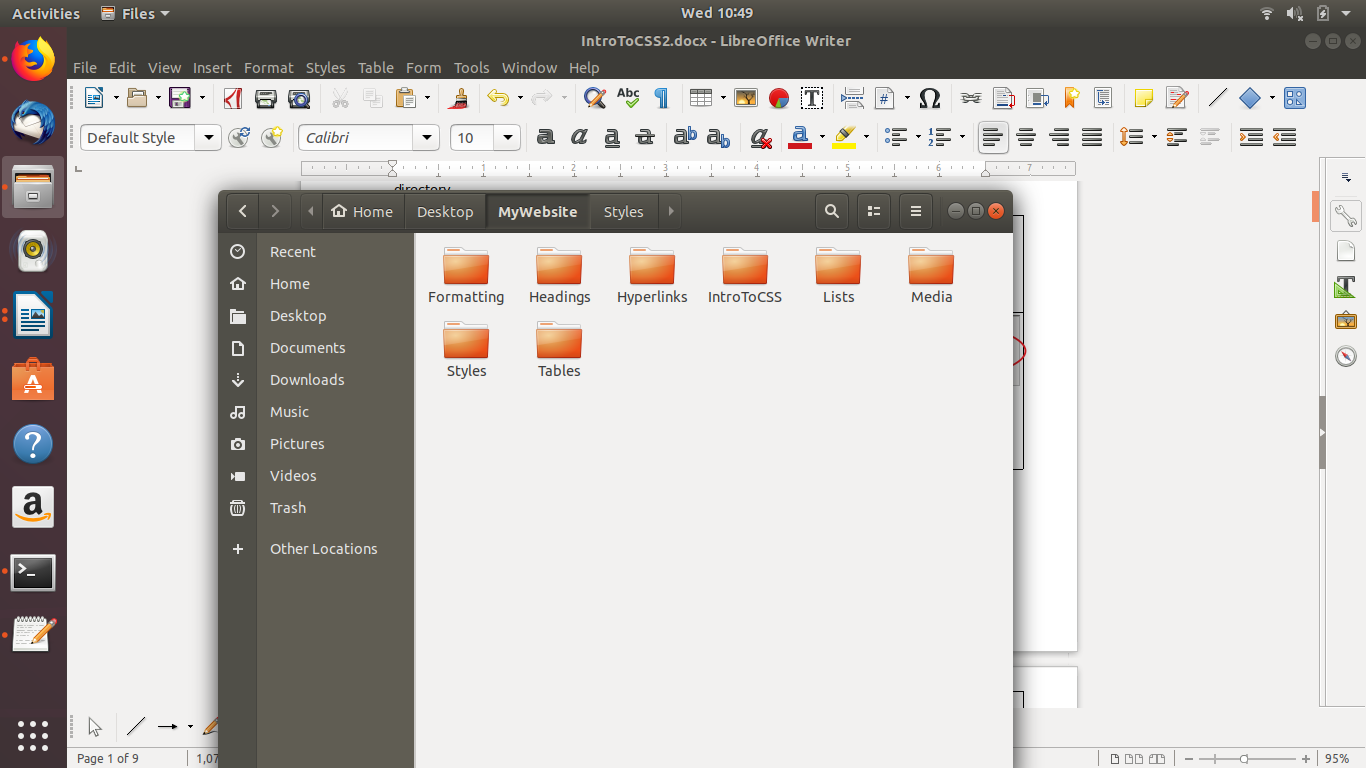
* Indicate the names of all group members.
* Have Ms. Pluska check your Intro to CSS tasks
* Submit your lab to the needs to be graded folder to receive credit for the group portion of this lab.
* Do not submit your lab until you have Ms. Pluska’s (or her designated TA’s) signature

\_\_\_\_\_\_\_

Follow the steps below to create a css page in your website directory,

|  |  |
| --- | --- |
| Navigate to your website directory |  |
| Inside this directory, create another directory called *Styles*  Your main website directory should have the structure like that shown on the right |  |
| Open your text editor – be sure to use the same text editor you used to create your HTML pages - and create a file in the Styles directory you just created called *StyleSheet.css* |  |

Now return to your main website directory and create another directory called IntroToCSS. Inside this directory create an HTML page called Index.html. Your main website directory should now look as follows,



**Be sure to have a separate directory for each assignment up to this point**

All tables contain rows and within those are cells in which we can store data. The table row tag, <tr></tr>, is used to create rows. The table data tag, <td></td>, is used to create cells. Consider the example below. The code below creates a table with one row. Each cell within the row displays a day of the week.

|  |  |
| --- | --- |
| <table>  <tr>  <td>Sunday</td>  <td>Monday</td>  <td>Tuesday</td>  <td>Wednesday</td>  <td>Thursday</td>  <td>Friday</td>  <td>Saturday</td>  </tr>  </table> |  |

|  |
| --- |
| Write code that could be used to create a table with two rows. In the first row, *breakfast*, *lunch*, *dinner* should be stored in separate cells. In the second row, *Eggs*, *PB & J*, *Pizza* should be stored in separate cells. When ran your code should display as follows, |
|  |

* **Create a table heading**

Table data doesn't make much sense without titles to describe what the data represents.

To add titles to rows and columns, you can use the table heading element: <th>.

The table heading element is used just like a table data element, except with a relevant title. And, just like table data, a table heading must be placed within a table row. This is demonstrated below,

|  |  |
| --- | --- |
| <table>  <tr>  <th>Sunday</th>  <th>Monday</th>  <th>Tuesday</th>  <th>Wednesday</th>  <th>Thursday</th>  <th>Friday</th>  <th>Saturday</th>  </tr>  <tr>  <td>Sleep in</td>  <td>Go to school</td>  <td>Go to school</td>  <td>Go to school</td>  <td>Go to school</td>  <td>Go to school</td>  <td>Sleep in</td>  </tr>  </table> |  |

|  |
| --- |
| Write code that could be used to create a table which displays the following months as table headers: January, May, July, September, October, November, December. In a second row, indicate a holiday that occurs in each month. When ran your code should display as follows, |
|  |

* **Have Ms. Pluska check off the above tasks**



Before you continue have Ms. Pluska check off the above tasks

Do not continue until you have Ms. Pluska’s (or her designated TA’s) signature \_\_\_\_\_\_\_\_\_\_\_\_

* **Write code to span rows and columns**

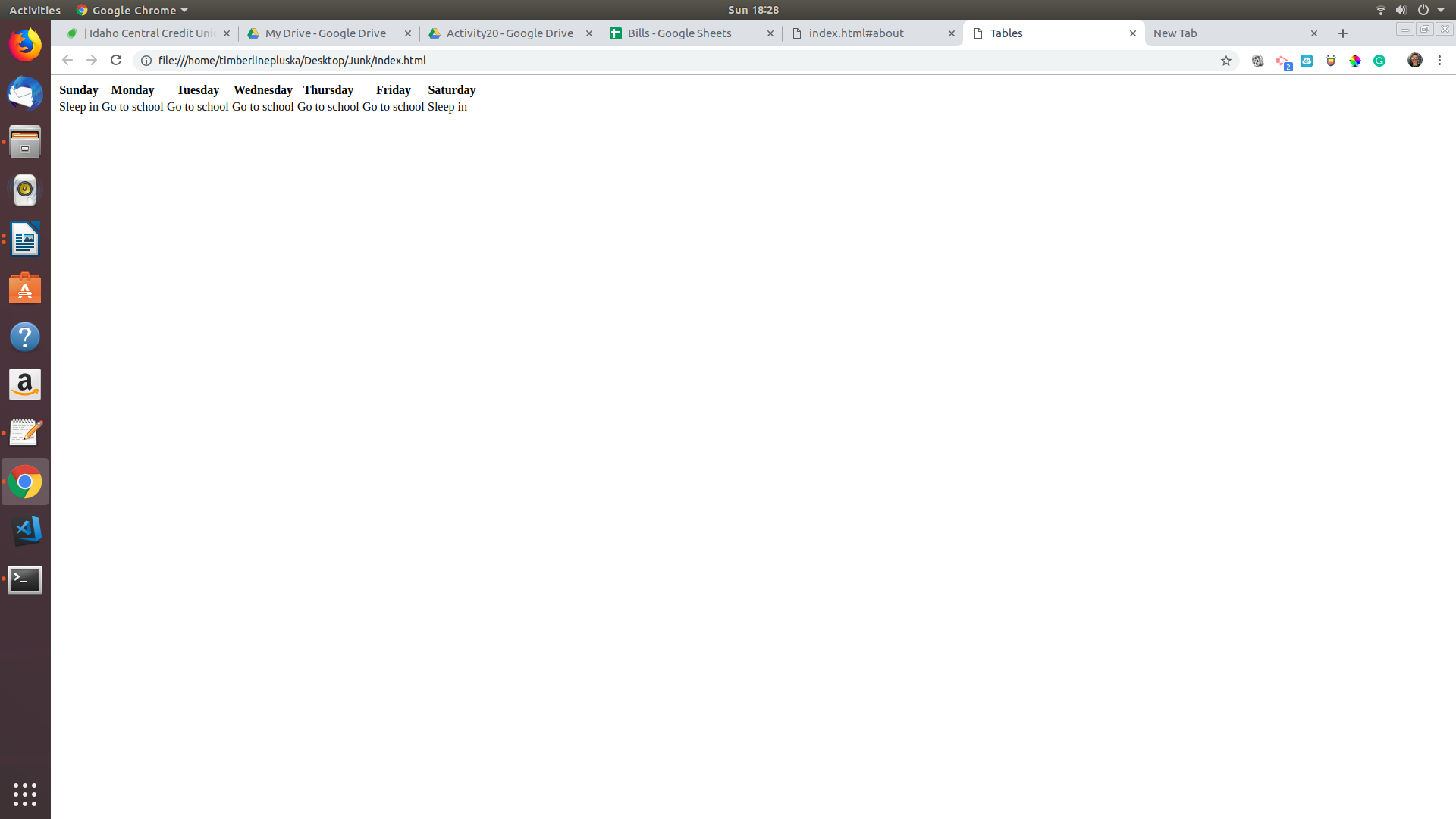
Recall that rows in an HTML table go across. For example, the code below would produce the following output.

|  |  |
| --- | --- |
| <table>  <tr>  <th>Row 1</th>  <th>Row 1</th>  <th>Row 1</th>  <th>Row 1</th>  <th>Row 1</th>  <th>Row 1</th>  <th>Row 1</th>  </tr>  <tr>  <th>Row 2</th>  <th>Row 2</th>  <th>Row 2</th>  <th>Row 2</th>  <th>Row 2</th>  <th>Row 2</th>  <th>Row 2</th>  </tr>  </table> |  |

Columns on the other hand, go down as illustrated in the following example,

|  |  |
| --- | --- |
| <table>  <tr>  <th>Column 1</th>  <th>Column 2</th>  <th>Column 3</th>  <th>Column 4</th>  <th>Column 5</th>  <th>Column 6</th>  <th>Column 7</th>  </tr>  <tr>  <th>Column 1</th>  <th>Column 2</th>  <th>Column 3</th>  <th>Column 4</th>  <th>Column 5</th>  <th>Column 6</th>  <th>Column 7</th>  </tr>  </table> |  |

In a previous example, we wrote code to generate the following output,



In this example we wrote “Go to school” 5 days in a row, or across 5 columns. This practice is redundant and the colspan attribute can be used to consolidate these columns and reduce the amoutn of code we need to write. See below,

|  |
| --- |
| <table>  <tr>  <th>Sunday</th>  <th>Monday</th>  <th>Tuesday</th>  <th>Wednesday</th>  <th>Thursday</th>  <th>Friday</th>  <th>Saturday</th>  </tr>  <tr>  <td>Sleep in</td>  <td colspan = 5>Go to school</td>  <td>Sleep in</td>  </tr>  </table> |
|  |

Although it may not be clear from the output, the cell *Go to school* now spans 5 columns. You can use the bgcolor attribute to change the color of the cell to make this more clear.

|  |
| --- |
| <table>  <tr>  <th>Sunday</th>  <th>Monday</th>  <th>Tuesday</th>  <th>Wednesday</th>  <th>Thursday</th>  <th>Friday</th>  <th>Saturday</th>  </tr>  <tr>  <td>Sleep in</td>  <td colspan=5 bgcolor="yellow">Go to school</td>  <td>Sleep in</td>  </tr>  </table> |
|  |

|  |
| --- |
| Write code that creates a table and displays each month as a header across the top. For each season, use the colspan attribute to indicate the season for each month. Use the bgcolor attribute to color each season (winter = lightblue, spring = green, summer = yellow, fall = orange). When ran your code should display as follows, |
|  |

To illustrate the column attribute, let’s return to our code that generated the days of the week. Depending on the time, you probably do different things on each day. For example, on Monday you may be in school from 8 to 3, whereas on Saturday you may have soccer practice from 1 to 3. The rowspan attribute enables us to indicate these blocks of time without writing unnecessary code.

|  |
| --- |
| <table>  <tr>  <th>Time</th>  <th>Saturday</th>  <th>Sunday</th>  <th>Monday</th>  <th>Tuesday</th>  <th>Wednesday</th>  <th>Thursday</th>  <th>Friday</th>  </tr>  <tr>  <th>8:00 am</th><td></td><td></td><td rowspan=8 bgcolor="yellow" >School</td><td></td><td></td><td></td><td></td>  </tr>  <tr>  <th>9:00 am</th><td></td><td></td><td></td><td></td><td></td><td></td><td></td>  </tr>  <tr>  <th>10:00 am</th><td></td><td></td><td></td><td></td><td></td><td></td><td></td>  </tr>  <tr>  <th>11:00 am</th><td></td><td></td><td></td><td></td><td></td><td></td><td></td>  </tr>  <tr>  <th>12:00 pm</th><td></td><td></td><td></td><td></td><td></td><td></td><td></td>  </tr>  <tr>  <th>1:00 pm</th><td rowspan = 2 bgcolor = "lightblue">Soccer practice</td><td></td><td></td><td></td><td></td><td></td><td></td>  </tr>  <tr>  <th>2:00 pm</th><td></td><td></td><td></td><td></td><td></td><td></td><td></td>  </tr>  <tr>  <th>3:00 pm</th><td></td><td></td><td></td><td></td><td></td><td></td><td></td>  </tr>  </table> |
|  |

|  |  |
| --- | --- |
| Write code that could be used to display the schedule shown. When ran your code should look like the code | |
|  |  |