Write your name below and indicate your role,

Project Manager (PM), Recorder (R)

Name _____ Role _____

Name _____ Role _____

Selectors

Your Tasks

- ☐ Write code to style a class attribute
- ☐ Style multiple class attributes associated with a single HTML tag
- ☐ Have Ms. Pluska check off the above tasks
- ☐ Write code to style an id attribute
- ☐ Differientate between the use of class and id attributes
- ☐ Have Ms. Pluska check off the above tasks
- ☐ Interpret the specificity of css rules
- ☐ Write code to *chain* selectors
- ☐ Write code to selected nested elements
- ☐ Receive credit for the group portion of this lab

☐ Write code to style a class attribute

CSS is not limited to selecting elements by tag name. HTML elements can have more than just a tag name; they can also have attributes. One common attribute is the *class* attribute. It's also possible to select an element by its *class* attribute. For example, consider the following HTML:

```
<p class="brand">Sole Shoe Company</p>
```

The paragraph element in the example above has a *class* attribute within the `<p>` tag. The *class* attribute is set to *"brand"*. To select this element using CSS, we could use the following CSS selector:

```
.brand {

}
```

To select an HTML element by its class using CSS, a period (.) must be prepended to the class's name. In the example above case, the class is *brand*, so the CSS selector for it is *.brand*.

Consider the code snippet below. Write a style rule for each of the following class attributes: *title, author, publish-time*

In your rules you can choose from the following properties: *text-decoration, color, text-align, font-size, font-weight, font-style*

Index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Vacation World</title>
  <link href="style.css" type="text/css"
rel="stylesheet">
</head>
<body>
  <img
src="https://s3.amazonaws.com/codecademy-
content/courses/freelance-1/unit-2/
explorer.jpeg" />
  <h1 class="title" >Top Vacation
Spots</h1>
  <h5 class=”author”>By: Stacy Gray</h5>
  <h6 class="publish-time">Published: 2
Days Ago</h6>

...

</body>
</html>
```

Styles.css

☐ Style multiple class attributes associated with a single HTML tag

In the previous example we selected elements using only one class name per element. If every HTML element had a single class, all the style information for each element would require a new class.

Luckily, it is possible to add more than one class name to an HTML element’s *class* attribute.

For instance, perhaps there’s a heading element that needs to be green and bold. You could write two CSS rules like so:

```
.green {
     color: green;
}
.bold {
     font-weight: bold;
}
```

Then, you could include both of these classes on one HTML element like this

```
<h1 class="green bold" >Top Vacation Spots</h1>
```

We can add multiple classes to an HTML element’s class attribute by separating them with a space. This enables us to mix and match CSS classes to create many unique styles without writing a custom class for every style combination needed.

(a) In the Index.html page, rewrite the `<h1>` tag shown below so that it contains the following class attributes: *title cursive uppercase*

```
<h1>Top Vacation Spots</h1>
```

(b) In the Styles.css page, add a class selector for the *title* attribute that will make all the letters in the `<h1>` tag uppercased. Do this by writing a class rule named .uppercase. Then, write inside the curly braces:

```
text-transform: uppercase;
```

(b) Write two more class rules – one for *title* and the other for *cursive*. The *title* rule should make the text bolder the *cursive* rule should make the text italic

Index.html	**Styles.css**

`<!DOCTYPE html>`
`<html>`
`<head>`
`  <title>Vacation World</title>`
`  <link href="style.css"`
`type="text/css" rel="stylesheet">`
`</head>`
`<body>`
 	
`</body>`
`</html>`	

☐ 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 style an id attribute

If an HTML element needs to be styled uniquely (no matter what classes are applied to the element), we can add an id to the element. To add an id to an element, the element needs an *id* attribute:

```
<h1 id="large-title"> ... </h1>
```

To select an id element, CSS prepends the id name with a hashtag (#). For instance, if we wanted to select the HTML element in the example above, it would look like this:

```
#large-title {

}
```

(a) In the Index.html page, rewrite the `<h2>` tag shown below so that it contains the following id attribute: *attractions*

```
<h2>top attractions</h2>
```

(b) In the Styles.css page, add an id selector for the *attractions* attribute that will make the first letter of each word in the `<h2>` tag uppercased. Do this by writing an id rule named *attractions.* Then, write inside the curly braces:

```
text-transform: capitalize;
```

(c) Inside the attractions rule set create two more id rules – one to make the color of the text blue and another to make the text underlined.

Index.html	**Styles.css**

`<!DOCTYPE html>`
`<html>`
`<head>`
`  <title>Vacation World</title>`
`  <link href="style.css"`
`type="text/css" rel="stylesheet">`
`</head>`
`<body>`
 	
`</body>`
`</html>`	

☐ Differientate between the use of class and id attributes

CSS can select HTML elements by their tag, class, and id. CSS classes and ids have different purposes, which can affect which one you use to style HTML elements.

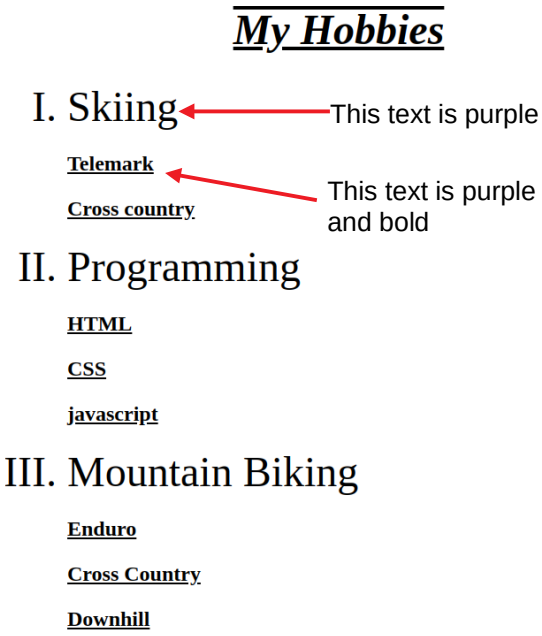
CSS classes are meant to be reused over many elements. By writing CSS classes, you can style elements in a variety of ways by mixing classes on HTML elements.

For instance, imagine a page with two headlines. One headline needs to be bold and blue, and the other needs to be bold and green. Instead of writing separate CSS rules for each headline that repeat each other's code, it's better to write a `.bold` CSS rule, a `.green` CSS rule, and a `.blue` CSS rule. Then you can give one headline the `bold green` classes, and the other the `bold blue` classes.

While classes are meant to be used many times, an id is meant to style only one element. And, ids override the styles of tags and classes. Since ids override class and tag styles, they should be used sparingly and only on elements that need to always appear the same.

Consider the output below. Complete the code in the Index.html. Then write appropriate rule-sets that could be used to produce the output shown. Be mindful of when to use class attributes and id attributes.

Output



Index.html

```
<!DOCTYPE html>
<html>
  <head>
<link rel = "stylesheet" type = "text/css"
href = "MyStyles.css" />
  </head>
  <body>









  </body>
</html>
```

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 _____

☐ Interpret the specificity of css rules

Specificity is the order by which the browser decides which CSS styles will be displayed. A best practice in CSS is to style elements while using the lowest degree of specificity, so that if an element needs a new style, it is easy to override.

ids are the most specific selector in CSS, followed by classes, and finally, tags. For example, consider the following HTML and CSS:

```
<h1 class="headline">Breaking News</h1>
```

```
h1 {
     color: red;
}

.headline {
     color: firebrick;
}
```

Breaking News would appear as *firebrick,* because classes override selectors. In otherwords they have a higher specificity.

In the example code above, the color of the heading would be set to firebrick, as the class selector is more specific than the tag selector. If an id attribute (and selector) were added to the code above, the styles within the id selector's body would override all other styles for the heading. The only way to override an id is to add another id with additional styling.

Over time, as files grow with code, many elements may have IDs, which can make CSS difficult to edit, since a new, more specific style must be created to change the style of an element.

To make styles easy to edit, it's best to style with a tag selector, if possible. If not, add a class selector. If that is not specific enough, then consider using an id selector.

Predict the output of the following code. For colored text indicate the color in parentheses after the text.

Index.html

```
<h1  id="article-title" class = "green
all-caps”>
<h2 class="green">1. Florence, Italy</h2>
    <h5 class="all-caps">Top
Attractions</h5>
    <ul>
      <li>Museums</li>
      <li>Bike Tours</li>
      <li>Historical Monuments</li>
    </ul>

  <h2 class="green">2. Beijing,
China</h2>
    <h5 class="all-caps">Top
Attractions</h5>
    <ul>
      <li>Biking</li>
      <li>Historical Sites</li>
      <li>Restaurants and Dining</li>
    </ul>
```

Styles.css

```
#article-title{

      color: blue;
      text-decoration: underline overline;
}

h1 {
      color: blue;
}

.green{
      color: green;
}

.blue{

      color: blue;
}

.all-caps{

      text-transform: uppercase;
}
```

Output

☐ 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 chain selectors

When writing CSS rules, it's possible to require an HTML element to have two or more CSS selectors at the same time.

This is done by combining multiple selectors, which we will refer to as chaining. For instance, if there was a *.special* class for `<h1>` elements, the CSS would look like:

```
h1.special {

}
```

The code above would select only the `<h1>` elements that have a class of *special*. If a `<p>` element also had a class of *special*, the rule in the example would not style the paragraph.

Predict the output of the following code. For colored text indicate the color in parentheses after the text. Do the same for bolded text.

Index.html

```
<h1  id="article-title" class = "green all-
caps special”>
<h2 class="green attractions">1. Florence,
Italy</h2>
    <h5 class="all-caps">Top Attractions</h5>
    <ul>
      <li class=“attractions”>Museums</li>
      <li class=“attractions”>Bike Tours</li>
      <li class=“attractions”>Historical
Monuments</li>
    </ul>

<h2 class="green attractions">2. Beijing,
China</h2>
    <h5 class="all-caps">Top Attractions</h5>
    <ul>
      <li class=“attractions”>Biking</li>
      <li class=“attractions”>Historical
Sites</li>
      <li class=“attractions”>Restaurants and
Dining</li>
    </ul>
```

Styles.css

```
#article-title{
      color: blue;
      text-decoration: underline
overline;
}

h1.special {
      color: blue;
}

.green{
      color: green;
}

.blue{
      color: blue;
}

.all-caps{
      text-transform: uppercase;
}

li.attractions{
      font-weight: bold;
}
```

Output

☐ Write code to select nested elements

In addition to chaining selectors to select elements, CSS also supports selecting elements that are nested within other HTML elements. For instance, consider the following HTML:

```
<ul class='main-list'>
     <li> ... </li>
     <li> ... </li>
     <li> ... </li>
</ul>
```

The nested `<li>` lements are selected with the following CSS:

```
.main-list li {

}
```

In the example above, *.main-list* selects the *.main-list* element (the unordered list element). The nested `<li>` are selected by adding *li* to the selector, separated by a space, resulting in *.main-list li* as the final selector (note the space in the selector).

For the list below, write a css rule that could be used to style the list as shown. In addition to the css properties you learned previously, you can also use the properties below,

Property	Values	Description
float	left, right, none	Places elements side by side
list-style-type	none, disc, circle, square, decimal, etc	Removes or changes the default bullet in a list
padding	.5 em, 1em, 1.5em, etc	Adds space around the elements content

Index.html

```
<ul class = "nav">
  <li><a class="active"
href="#main">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
```

Output

Home News Contact About

Styles.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

_____