

CSS SELECTORS

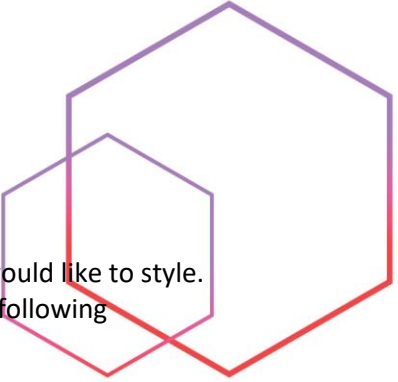
APRIL 2022

YONATAN BENEZRA

- DO NOT DISTRIBUTE –

- DO NOT COPY – ALL RIGHTS RESERVED TO THE
AUTHOR –

CSS Selectors



Selectors are a way for our CSS to understand what element we want to style. In this lesson, we will learn a couple of different ways to select the element we would like to style. The first method we will discuss is to use the element tag name, as shown in the following

Example:

```
div {  
  color: yellow;  
}
```

After entering this code into our style.css, we will see the text of all our divs change to yellow. So how do we select one specific element?

Class and ID Selectors

By giving the element a class or ID, we can easily select a specific element. For instance, if we have a header with the class "title":

```
<h1 class="title"></h1>
```

We can style this header using the class name, as follows:

```
.title {  
  font-family: italic;  
}
```

The above code will change the font of all the elements with the class name "title". Notice the "." before the class name; if it was an ID we would use a "#".

*Note:

it is important to give each element a proper class name or id for cleaner code. In cases where we want to make sure only a specific element will get a certain style, we will use an ID selector.

For example, if we have a div with an id of "my-name":

```
<div id="my-name">Yonatan.</div>
```

Then our CSS will look as follows:

```
#my-name {  
  color: red;
```

```
}
```

Notice we are selecting the ID with a "#" and not with a "."

If we want to change the style for all the elements on the page, we can use the universal selector (*), which selects all HTML elements on the page.

To summarize:

We will assign classes to elements that are in a bigger styling group, and an ID for elements that are to receive more specific styling. Each element can have both an ID attribute and a class (or even more than one class).

Pseudo Selectors

Using the pseudo selector, we can give an HTML element a special state. For example, if we want our buttons to change color when a mouse hovers over them, we can write the following in our CSS:

```
button:hover {  
  background-color: orange;  
  color: white;  
  border: 1px solid black;  
}
```

We can use the Pseudo selectors to style our links as well, as follows:

```
/* unvisited link */  
a:link {  
  color: #FF0000;  
}  
  
/* visited link */  
a:visited {  
  color: #00FF00;  
}  
  
/* mouse over link */  
a:hover {
```

```
color: #FF00FF;
}

/* selected link */
a:active {
  color: #0000FF;
}
```

We can even create a simple tooltip using CSS. To do so, we will first need to learn about the display property:

The display property allows us to hide/show certain elements on the page according to rules that we set.

Let's look at an example. In the bellow image, we set a tooltip using the display property along with pseudo selectors.

```
p {
  display: none;
  background-color: yellow;
  padding: 20px;
}

div:hover p {
  display: block;
}
```

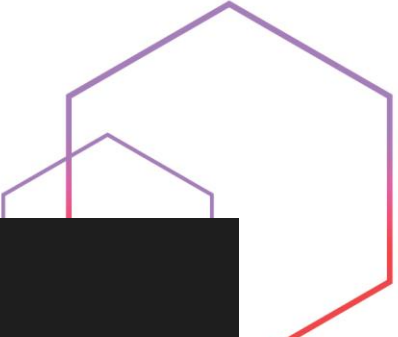
Exercise:

Write the necessary html code to complete the CSS in order to create a tooltip.

Combining Selectors:

We can combine CSS selectors as well. Let's try to color all of the "p" tags inside all divs only, but not ones that are outside of divs.

First, let's write some HTML to meet the requirements:



```
<div>
  <p>I am inside a div</p>
  <p>I am inside a div</p>
</div>
<p>I am outside of a div</p>
```

Let's add the CSS with combined selectors:

```
div p {
  color: orange;
}
```

As you can see, only the "p" tags inside of a div were colored! [Here](#) you can find additional options for combining selectors.

Take a look [here](#) as well to see it happening.

