

# COMP 1017

Day 12  
Selectors

In CSS, we use **selectors** to target the HTML in our web pages that we want to style.

So, what kind of selectors exist?

# Selectors

# HTML Elements

```
html { }
```

```
body { }
```

```
h1 { }
```

```
section { }
```

```
ul { }
```

```
li { }
```

ID's

```
#hero-banner { }
```

```
#jumbotron { }
```

**IDs** are very powerful selectors that can only be used once per page.

We won't be using IDs in this course because they can **override** just about anything.



# Classes

```
.container { }
```

```
.container-fluid {  
}
```

```
.row { }
```

```
.banner-image { }
```

We *will* be using all sorts of **classes**, so  
let's talk about them!

# Classes

What is a **class**?

A class is a **bunch of rules** that we can write in our CSS and add to anything within our HTML.

Classes have a dot ( . ), followed by a  
semantic name.

```
.container-960px {  
    width: 960px;  
}
```

In order for classes to work, they need to be in **two places**:

1. your HTML\*
2. your CSS

\* this will be the **value** of the **class** attribute



```
<p class="container-960px">
```

You can add as many classes to your HTML elements as you like!

All you have to do is separate your class names by spaces.

```
<ul>

  <li class="item">One</li>

  <li class="item two">Two</li>

  <li class="item item-link">Three</li>

</ul>
```

```
.item {
    font-size: 18px;
}

.two {
    color : pink;
}

.item-link {
    font-weight : bold;
}
```

# Best Practices

Let's talk about a few do's and don'ts.

**Do** choose class names that are relevant to what you're doing.

Don't just use numbers or letters or something arbitrary.



**Do** use hyphens ( - ) to concatenate words.

Don't use spaces or camelCase.

do keep everything lowercase.

Don't start your name off with a number.

element.class selector

We can define classes inside of our  
CSS ...

```
.red-text {  
    color: red;  
}
```

... but we can also target specific elements **with** specific classes applied to them.



```
p.red-text {  
    font-size: 56px;  
}
```

```
/* This targets paragraph elements  
with the class .red-text applied. */
```

```
/* It will not target other elements  
with .red-text applied. It will also  
not change the rules of the .red-  
text class. */
```

multiple element selector

We can select **multiple elements** at a time, allowing us to apply the same rules to each element.

All it takes is a **comma** ( , ) between each element that we want to select.

```
h1, h2 {  
    color: blue;  
}
```

descendant selector

A descendant selector looks at an element's 'lineage' and place within the DOM.



This allows us to be much more explicit about which elements we want to target.

It is written with a **space** ( ) between each element.

It starts with the **parent element** and works inward.

```
header h2 {  
    color: blue;  
}
```

```
/* This only targets second-  
level headings inside of the  
header. */
```

```
footer a {  
    text-decoration: none;  
}
```

```
/* This only targets hyperlinks  
   inside of the footer. */
```

descendant combinator

A descendant combinator takes all of these things and puts them together.

```
ul.my-things li {  
    color: blue;  
}
```

```
/* This targets list items  
inside of any unordered list  
with the .my-things class. */
```



```
<ul class="my-things">
```

```
  <li>One</li>
```

```
  <li>Two</li>
```

```
</ul>
```

```
<!-- The list items are  
      targeted here. -->
```

```
<ul>
```

```
  <li>One</li>
```

```
  <li>Two</li>
```

```
</ul>
```

```
<!-- These are not. -->
```

```
<ol>
```

```
  <li>One</li>
```

```
  <li class="my-  
things">Two</li>
```

```
</ol>
```

```
<!-- These are also not. -->
```

pseudo-class selectors

A **pseudo-class selector** targets an element when it's in a certain state.

One of the best examples of this is applying rules to **hyperlinks** to change their look or behaviour.

```
a:hover {
```

```
    text-decoration: underline;
```

```
}
```

```
/* When the user moves their mouse over the  
   hyperlink, an underline will appear. */
```