USING CSS

IN THIS LESSON

External stylesheets

Internal stylesheets

Inline styles

CSS selectors

Class vs id

USING CSS: EXTERNAL STYLESHEETS

Recommended way to use CSS

CSS rulesets written inside text document with **.css** extension

In HTML document, add **<link>** element inside **<head>**:

<link rel="stylesheet" href="css/styles.css" />

For **rel** (relationship) attribute, always use **"stylesheet"**

For **href** attribute, provide path to external stylesheet

This path can be to local file, or to stylesheet hosted elsewhere on the web, often on a CDN (content delivery network):

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" />

USING CSS: INTERNAL STYLESHEETS

Use **<style>** element inside HTML document's **<head>**

Write CSS rulesets inside the **<style>** element:

```
<style>
  p {
      color: red;
  }
</style>
```

USING CSS: INLINE STYLES

Use **style** attribute in an HTML element to directly apply a style

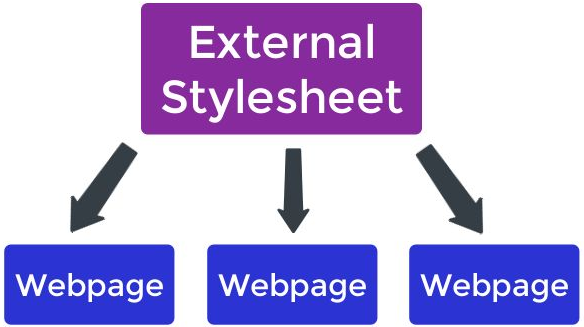
Write CSS declarations inside the **style** attribute's value

```
<p style="color: red;">This text will be red.</p>
```

AVOID INTERNAL STYLESHEETS & INLINE STYLES

Harder to maintain/update/organize

They can only apply to a single webpage; external stylesheet can be applied to unlimited number of webpages



Internal stylesheets/inline styles can be useful during development for testing styles

CSS SELECTORS

Every CSS ruleset has a selector at the beginning

Three primary kinds of selectors: **element/type**, **class**, **id**

CSS SELECTORS: ELEMENT/TYPE

```
p {
    color: red;
}
```

```
<p>This text will be red.</p>
```

This selector targets all paragraph elements in the linked HTML document

CSS SELECTORS: CLASS

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

Create a **class** selector with a **period/full stop**, followed by the **class name**

Name must not start with a number; should be lower case, no spaces, use a hyphen to separate words

This selector targets all elements with a **class** attribute that contains the **class name** (not including the **full stop**):

```
<h1 class="text-primary">This text will be red.<h1>
<p class="text-primary">This text will be red too.</p>
```

CSS relies heavily on **class** selectors; they are the most flexible & reuseable.

CSS SELECTORS: ID

```
#copyright {
   font-size: 12px;
}
```

Create an **id** selector with a **#** followed by the **id name**

Name must not start with a number; should be lower case, no spaces, use a hyphen to separate words

This selector targets the element with an **id** attribute that matches the **id name** from the stylesheet (not including the **#**).

```
<p id="copyright">This text will be sized 12 px.</p>
```

Each **id name** can only be used once per HTML document.
Use **id** selectors sparingly. **Class**es are preferred.

CLASS VS ID

Can set **class** or **id** attribute on any HTML element

Main difference: **class**es can be applied to multiple elements; **id**s should be used as a unique identifier on only one element in the document

Analogy: driver's license - unique id #, can have multiple classes

If you accidentally set the same **id** to more than one element in a document, browser will not warn you and ignore all but the first element when you target that **id**, so be careful

CLASS VS ID

```
.text-primary {
    color: red;
}
.bg-primary {
    background: black;
}
```

Can set multiple **classes** on same element

Then set both on same element to combine the rules, separate with a space:

```
<p class="text-primary  bg-primary">This text will be red with a black background.</p>
```

You cannot do this with **id**s, **id** attribute can only hold one **id**:

~~ ... ~~