Learning Objective 1

Demonstrate how to import other CSS files into your CSS file

```
@import 'custom.css';
```

Learning Objective 2

Explain how CSS rules are applied based on their order and specificity.

```
Specificity refers to the ordering of CSS properties when there are
multiple CSS rules that apply to the same element.

Specificity is calculated based on the number and type of the selectors
used in your CSS rules

There are three types of selectors that effect specificity, they are: Type
Selectors (which includes div, h1, and psuedo-selectors: ::before,
::after), Class Selectors (which includes classes .hidden, attribute
selectors [type="radio"], and psuedo-classes :hover) and ID Selectors
(which only includes ids #this-id).

Type selectors have the least impact on specificity, class selectors the
middle and ID selectors the most. First count the number of each type of
selector and then enter them into this formula to calculate specificity

<ID Selector Count> * 100 + <Class Selector Count> * 10 + <Type Selector
Count> * 1

The rule with the highest specificity wins and is applied to the styling
of the element.
```

Here's a visual!

Learning Objective 3

Describe and apply element, id, and class selectors

```
div {
color: red;
font-size: 32px;
}

.container {
```

```
color: blue;
}

#main {
color: green;
}
```

Learning Objective 4

Write "combinators" to create compound selector statements to target specific elements

```
There are four types of combinators: descendant space, child >, adjacent
sibling +, and general sibling ~.
```

```
#content div {// this selector will select all divs that are directly
inside #content
}
#content > div {// this selector will select all divs that are descendents
of #content
}
```

Learning Objective 5

Explain and apply pseudo selectors for specific elements in specific states

```
There are psuedo-classes like :active, :first-child, :only-child, :hover
that allow us to write conditional CSS

There are also psuedo-elements like ::first-line, ::before, and ::after
that allow us to write CSS at a more granular level than our DOM Elements.
```

Learning Objective 6

Explain and apply the ::before and ::after pseudo elements

```
::before creates a psuedo-element as the first child of the selected
element. This allows us to apply cosmetic content to an element using the
CSS content property:

::after does the same as ::before, but the psuedo element is appended to
the end of the selected element.
```

```
a::before { // Here a is the selected element, so all links will be prepended
    content: "♥";
}
```

Learning Objective 7

Style content on an HTML page targeting

- Type faces, sizes, styles, and weights
- Text transformation and alignment
- Colors expressed as names, hexadecimal RGB values, and decimal RGB values
- Everything about borders
- Shadows
- Opacity (transparency)
- Covering an element with a background image

Learning Objective 8

Explain the generic font names "serif", "sans-serif", and "monospace" and correctly identify examples of each

```
Generic font names allow you to defer font rendering to the user's web browser. Each browser or operating system may select a font that meets the general characteristics of the generic font name.

- Serif fonts have little visual accents on the ends of the letters
- Sans-Serif fonts are those without Serifs
- Monospaced fonts are those used for rendering code, each character takes up the same width, they have one (mono) spacing.
```

Learning Objective 9

Explain why using Web fonts helps with consistent experience across viewing devices

```
Because web fonts load an entire font alongside your website, the user is guaranteed the same experience regardless of which browser or device they are using.
```

Learning Objective 10

Recall and explain the different absolute and relative length units in CSS

```
- Absolute length units are not relative to anything else and are generally considered to always be the same size: cm, mm, Q, in, pc, pt, px
- Relative length units are relative to something else (the size of a
parent element's font, the size of the viewport, etc). With careful
planning you can make it so elements scale relative to the rest of the
page: em, ex, ch, rem, lh, vw, vh, vmin, vmax
```

Learning Objective 11

Demonstrate how to link a stylesheet into an HTML page

```html
<head>
    <link rel="stylesheet" href="custom.css">
</head>
```

Learning Objective 12

Use the content CSS property to define the content of an element

```
Typically used with psuedo-elements, the content property can be used to
specify the content of an element.
```