Web Development Course

Project 3: Dialogue

Project: https://dialogue-project.glitch.me/
Code: https://glitch.com/edit/#!/dialogue-project?path=index.html%3A102%3A67
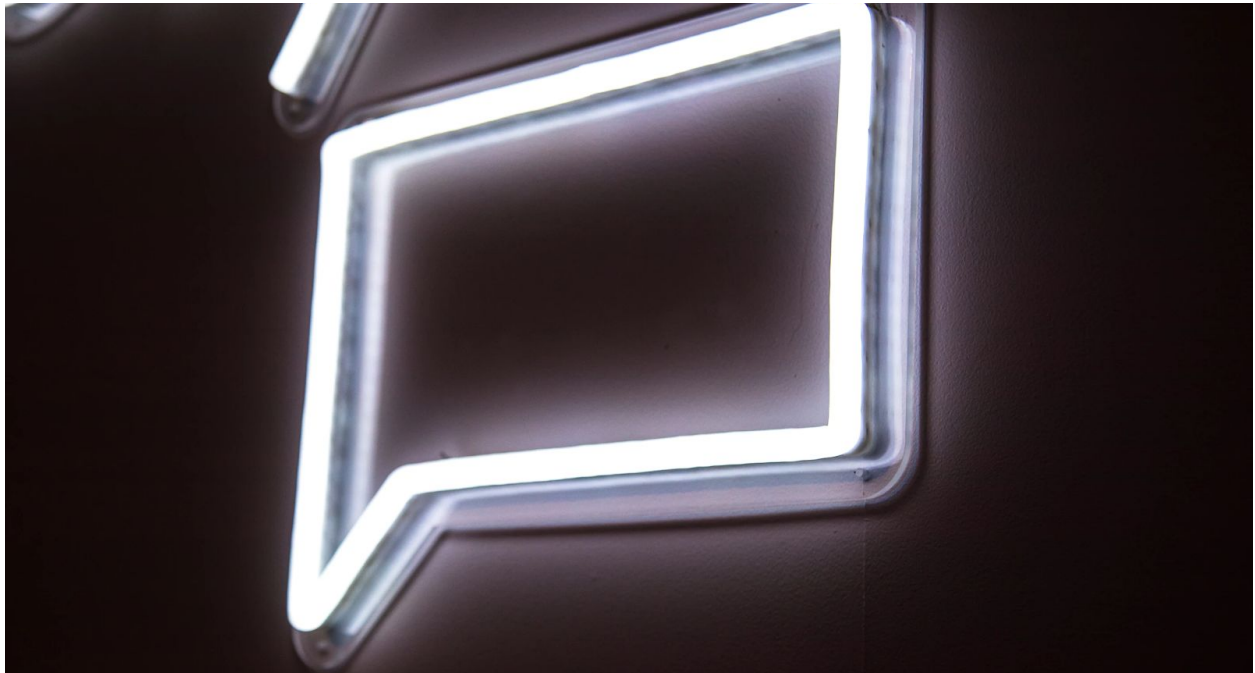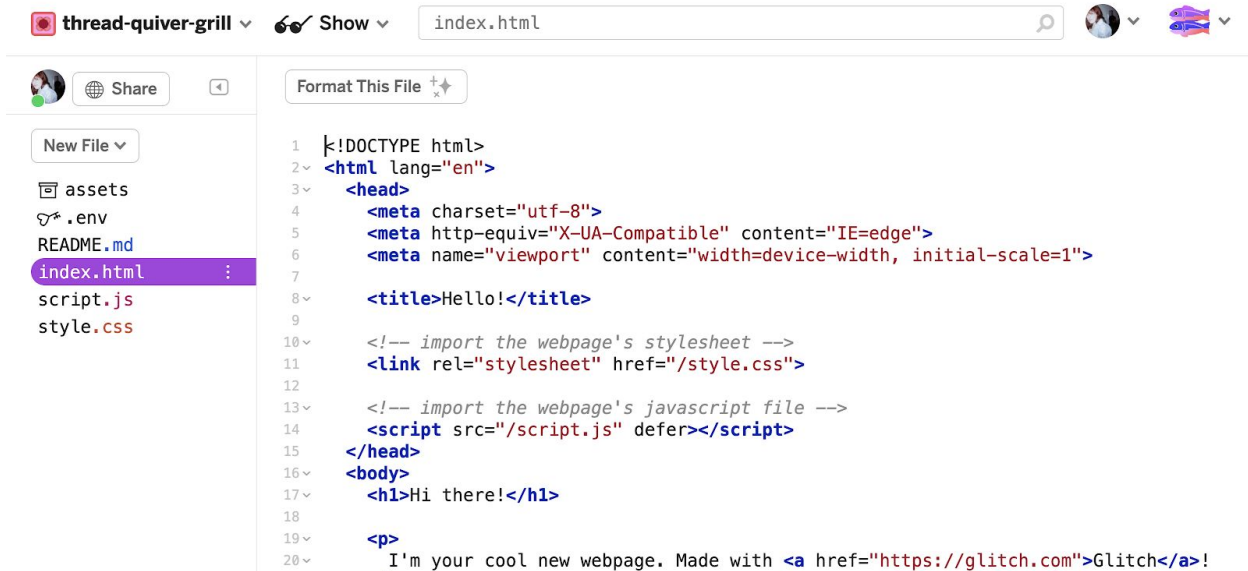
Big Idea: Classes let you style many parts of a webpage at once

Time: 35-45 minutes

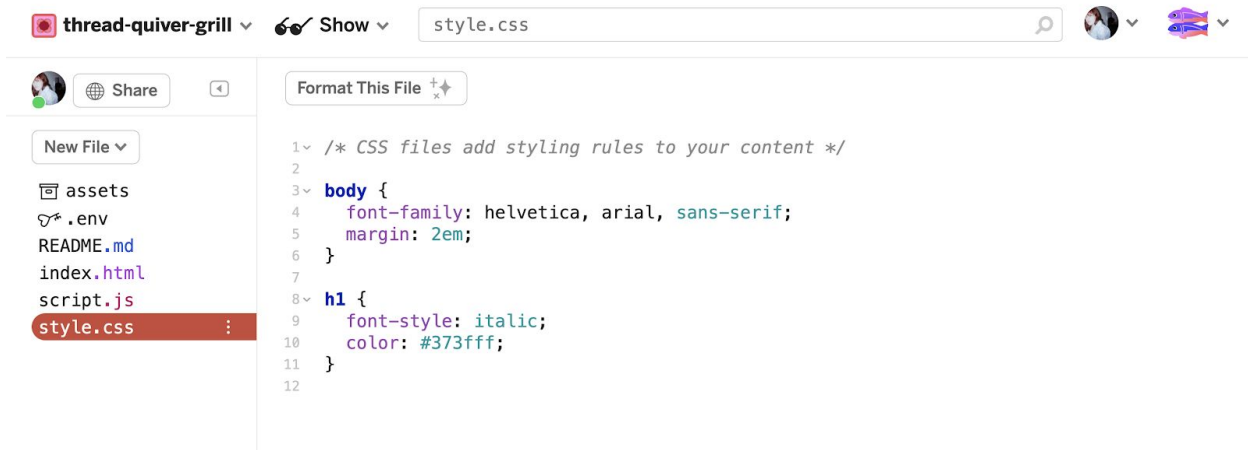|	
10 minutes	Introduce classes and project planning
20-25 minutes	Dialogue project work
5-10 minutes	Sharing and Reflection

Background

Files

Remember that HTML gets added to the **index.html** file - this is the content of the site. Text, photos, and music all goes here.

CSS styles the HTML content (**style.css** file). This is where you can customize things like color, font, and layout!

The HTML page is **linked** to the CSS page with the line of code:

```
<link rel="stylesheet" href="/style.css">
```

This is added by default to the `<head>` of the site. Your students don't have to worry about this yet, but it's interesting to know how it works!

Classes

Classes can style many elements the same way. They are applied directly to HTML elements. Remember how in the last project you edited all text on a page with the code:

```
body {
    font-family: "Times New Roman", Times, serif;
}
```

Well, you can get more specific! If instead wanted to edit all the **paragraph** tags on your site, which look like:

```
<p>Hello there!</p>
```

You could use the CSS:

```
p {
        color: blue;
}
```

But what if you want some paragraphs to be one color, and others to be another color? What if you want different fonts for different lines of dialogue in a story? Meet **classes.**

Classes can be applied to any HTML element to all style them the same way. If I had wanted all my text about tomatoes to be red, I could write code that looks like:
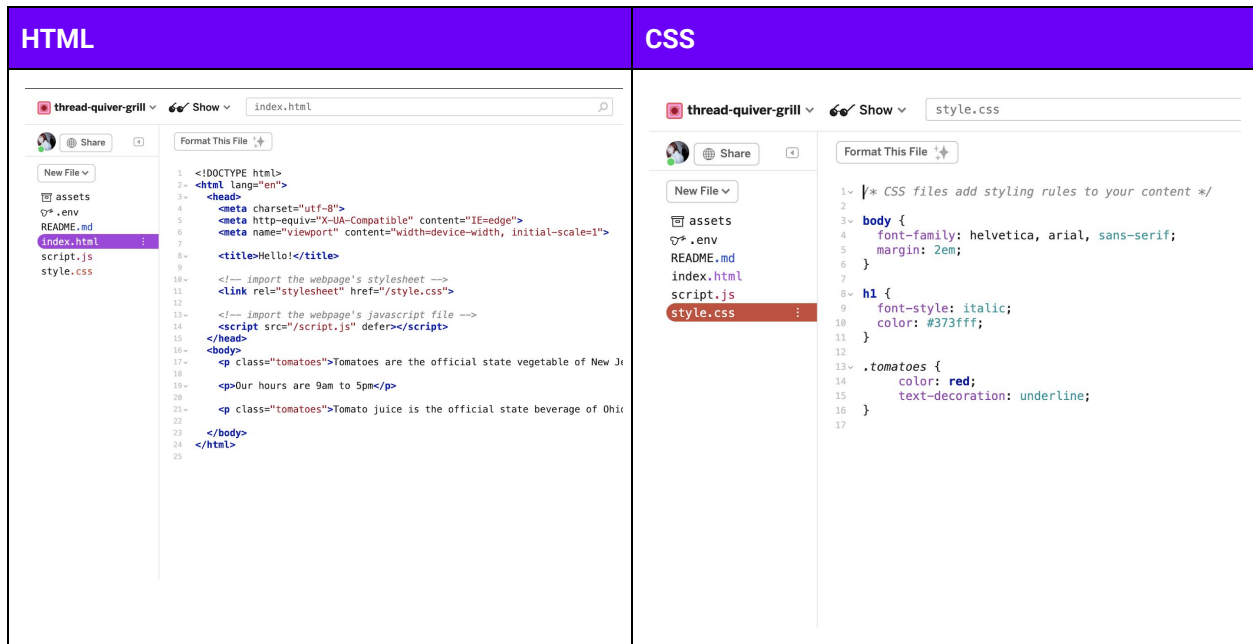
```
<p class="tomatoes">Tomatoes are the official state vegetable of New Jersey.</p>

<p>Our hours are 9am to 5pm</p>

<p class="tomatoes">Tomato juice is the official state beverage of Ohio.</p>
```

Then in CSS, you would write:

```
.tomatoes {
    color: red;
    text-decoration: underline;
}
```

HTML	CSS

Classes are selected in CSS with a period: **.nameOfClass**
These names are arbitrary, above we chose *tomatoes*, but it could have been anything, as long as the word in **HTML** and **CSS** match.

And your webpage looks like this!

> Tomatoes are the official state vegetable of New Jersey.
>
> Our hours are 9am to 5pm
>
> Tomato juice is the official state beverage of Ohio.
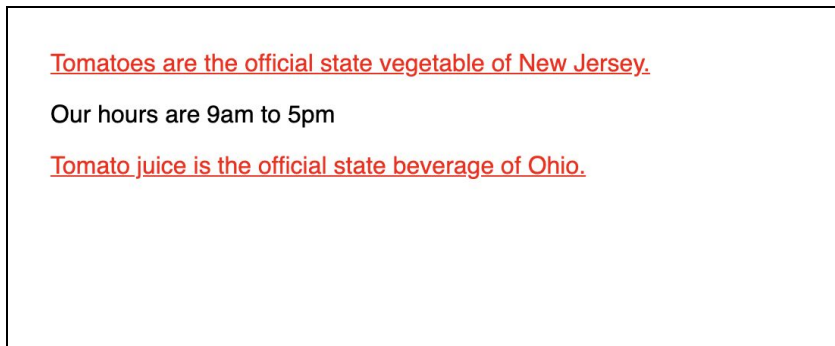
Teacher Reference:
More on HTML classes
Video on classes for HTML and CSS (Timestamp: 0:00 to 4:43)

Dialogue Project

Design a dialogue between multiple people, students can choose a conversation from their favorite book or tv show!

Project Checklist for Students
- Style the text accordingly, customizing it based on who's speaking.
- Add at least 3 pictures that help tell the story.

In addition to styling elements by the **type of element**, you can also use **classes** to style groups of elements the same.

For example, if I wanted all the lines of a **certain character** to be red, I could add to my code:

```
<p class="red">Pizza rain! ...But no pizza clouds...</p>
```

Then in your css, can can add:

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

Now you can add as many classes with <p **class="red"**>Your text here</p> as you want to your code, and it will turn all the elements you set to that class red! This way, you can have many paragraph tags, and only change the ones you want to. You have a lot more control over your project now!

Classes always have a . before them in the CSS file.
HTML: `<p class="character1">Line of Dialogue here</p>`

CSS:
```
.character1 {
     color: blue;
     font-style: italic;
     font-family: Arial, Helvetica, sans-serif;
}
```

Ways to customize text with CSS:
```
letter-spacing: 5px;
font-family: Arial, Helvetica, sans-serif;
font-size: 12px;
text-decoration: underline;
text-shadow: 2px 2px 8px orange;
font-weight: bold;
font-style: italic;
```

Font cheat sheet:
- Georgia, serif
- "Palatino Linotype", "Book Antiqua", Palatino, serif
- "Times New Roman", Times, serif
- Arial, Helvetica, sans-serif
- "Comic Sans MS", cursive, sans-serif
- "Courier New", Courier, monospace
- Papyrus, fantasy;

Discussion Points
- Did classes make things easier to style? Did they save you any time?
- How were you able to represent characters with CSS properties?
- Did you discover any new CSS properties you didn't know about before? Did someone else make something you hadn't seen?
- How did you choose your class names? What choices can you make in your code to make your code easier to understand?