

Class 3 - Introduction to CSS

Agenda:

- Rules of CSS
- Ways to Add CSS
- CSS Selectors
- Properties of CSS (color, background, fonts, and texts)

We have already discussed in the previous class a brief overview of CSS, now let's see how to use CSS and Rules to Use CSS.

CSS Rules

CSS is a rule-based language — you define the rules by specifying groups of styles that should be applied to particular elements or groups of elements on your web page.

For example, you can decide to have the main heading on your page to be shown as large red text. The following code shows a very simple CSS rule that would achieve the styling described above:

Steps to add style tag:

- Add `<style>` and `</style>` tag inside the head tag of your HTML file. Inside this **style** tag, you can use all the CSS stylings.
- Select the element that you need to add CSS.
- For example, h1 and add curly braces to it. Inside this, you can define the CSS properties as shown below:

```
<style>
h1 {
  color: red;
  font-size: 5em;
}
</style>
```

In the above example, the CSS rule opens with a **selector**. This *selects* the HTML element that we are going to style. In this case, we are styling level one headings (**h1**).

- We then have a set of curly braces `{ }`.

- Inside the braces will be one or more **declarations**, which take the form of **property** and **value** pairs. We specify the property (`color` in the above example) before the colon, and we specify the value of the property after the colon (`red` in this example).
- This example contains two declarations, one for `color` and the other for `font-size` . Each pair specifies a property of the element(s) we are selecting (`h1` in this case), then a value that we'd like to give the property.

Various ways of Using CSS They are:

1. External
2. Internal
3. Inline

Using CSS in three different ways:

- Before Moving to Different ways of using CSS , Have you ever Wondered that when you do not use CSS even then there is some default styling is set for your elements , Your button looks a certain way , Input boxes look a certain way , Ever Wondered How these Designs are applied?
- The Answer is User-Agent StyleSheet

User Agent Style Sheet

The user agent stylesheet in CSS is a set of default styles that web browsers apply to HTML documents when no specific styles are provided by the author. Each web browser has its own user agent stylesheet, and it serves as a baseline style for rendering HTML elements. The term "user agent" refers to the web browser that acts on behalf of the user when interacting with web content.

Now Let's Move Forward and See how to Apply your Own CSS styling and Different ways of applying them!

1. External CSS

- You can create a separate file for CSS having the extension **".css"**.
- Here you do not need to use any HTML tag. You can directly write your CSS and properties.

Example

```
1 | h3{
2 |     color: green;
3 | }
```

To reflect these CSS into your HTML file, you need to link that CSS file to the HTML file.

- Use the "link" tag to do this as shown below.

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

- You can add the file location in the **href** to add the CSS file
- The code Will look like this ->

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Example</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

<div>
  <h1>Welcome to Scaler</h1>

  <h2>I hope you are Doing good</h2>
  <h3>See you Again!</h3>

</div>

</body>
</html>
```

You will notice the color of h3 will become green

2. Internal CSS

- When you write CSS in the same HTML file. (using the style tag)
- basic example of using internal CSS within an HTML document:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<link rel="stylesheet" href="style.css">
<title>CSS Example</title>
<style>
    /* Internal CSS */

    h1 {
        color: tomato;
    }

</style>
</head>
<body>

<div>
    <h1>Welcome to Scaler</h1>

    <h2>I hope you are Doing good</h2>
    <h3>See you Again!</h3>

</div>

</body>
</html>

```

In this example, the CSS styles are defined within a `<style>` tag in the `<head>` section of the HTML document. These styles apply to the elements within the HTML document. `h1` sets the color of heading text.

Refer to this for more adv information on style tag:-

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/style>

3. Inline CSS

- It is writing CSS for a particular element. (using style attribute.)

Example

Inline CSS involves styling individual HTML elements using the `style` attribute directly within the element tag. Here's a basic example:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title> CSS Example</title>
    <style>
        /* Internal CSS */

        h1 {
            color: #333;
        }

```

```

    </style>
</head>
<body>

<div>
  <h1>Welcome to Scaler</h1>

  <!-- <Inline CSS> -->
  <h2 style='color:blue'>I hope You are doing Good</h2>

  <h3> See you again </h3>

</div>

</body>
</html>

```

In this example, the styles are directly applied to each HTML element using the `style` attribute within the opening tag. This allows for individual styling of elements without the need for a separate CSS file or `<style>` tag in the `<head>` section.

- Here we are providing the CSS to that specific element "h2" only, known as Inline CSS. Always try to add a style tag inside the **head tag**.

These are the three ways that can be used to apply CSS to your HTML file.

Little on the Precedence

CSS precedence determines which styles will be applied to an HTML element when conflicting styles are defined in multiple places. Here's the precedence order:

1. **Inline CSS:** Styles defined directly within the HTML element using the `style` attribute have the highest precedence. They override any other styles applied to the same element.
2. **Internal CSS:** Styles defined within a `<style>` tag in the `<head>` section of the HTML document come next in precedence. They apply to all elements of the same type throughout the document, unless overridden by inline styles.
3. **External CSS:** Styles defined in an external CSS file linked to the HTML document using the `<link>` tag have the lowest precedence. They apply globally to all elements of the specified type, unless overridden by inline or internal styles.

In cases of conflicting styles (where the same property is defined in multiple places), the style with the highest precedence will be applied. If two conflicting styles have the same precedence, the one defined later in the document will override the earlier one.

It's important to use CSS precedence carefully to ensure that styles are applied as intended and to avoid unexpected behavior.

- As we have seen in the previous example, **h1** was the selector. But there are more ways of selecting elements.
- Have a brief discussion about how the students can select an element as the previous method that we learned till now.

Breif Explanation on Class and ID Selectors and how to use them

In CSS (Cascading Style Sheets), a class selector is a way to select and style HTML elements based on the presence of a specific class attribute. Class selectors are denoted by a period (.) followed by the class name, and they allow you to apply styles to multiple elements across your HTML document that share the same class.

Syntax:

```
.className {  
    /* styles go here */  
}
```

Real-life Analogy: Imagine a classroom with students of different interests, and each interest group has a specific color associated with it. Students interested in science wear blue badges, students interested in literature wear red badges, and so on.

Example: Consider the following HTML snippet:

All h1 here represents students

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>ClassRoom</title>  
    <link rel="stylesheet" href="styles.css">  
</head>  
<body>  
    <h1 class="science">Student-1 - Science</h1>  
    <h1 class="science">Student-2 - Science</h1>  
    <h1 class="literature">Student-3 - Litetrature</h1>  
    <h1 class="literatre">Student-4 - Litetrature</h1>  
    <h1 class="Science">Student-5 - Science</h1>  
  
</body>  
</html>
```

Now, let's create a CSS file (`styles.css`) to apply styles using the class selector:

```
.science {  
  color: blue;  
  font-weight: bold;  
}  
  
.literature {  
  color: red;  
  font-weight: bold;  
}
```

In this example, the class selector `.science` and `.literature` are used to select elements with the specific interest and apply a style that changes the text color to blue and red based on the students interest.

There can be multiple students with interest in science so by creating a science class we can assign them similar properties as we gave them batches similarly it will work for literature students as well

So class is just a way of grouping elements where they should behave and have similar properties.

In this analogy:

- The HTML document represents the classroom.
- Each student corresponds to an HTML element (h1).
- The different interests represent classes.
- The color-coded badges are like class properties.

Now let's look at what is the purpose of id Selector

In CSS (Cascading Style Sheets), an ID selector is used to select and style a specific HTML element based on its unique identifier. The syntax for an ID selector is to use the "#" symbol followed by the unique ID of the HTML element.

Real-life Analogy:

Imagine a library where each book has a unique identification number. In this analogy:

- The library is like an HTML document.
- Each book is like an HTML element.
- The unique identification number assigned to each book is like the HTML element's ID.

Now, consider that you want to highlight or mark a specific book you want to read. You would use its unique identification number to locate it, here you cannot and should not

provide the same highlight color to different books that won't be good because then you won't be able to identify it uniquely

In CSS terms:

- The unique identification number of the book is similar to the ID of an HTML element.
- The process of styling or marking that specific book is like applying styles to the HTML element using the ID selector in CSS.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ID Selector Book Example</title>
  <style>
    #uniqueElement {
      color: blue;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <h1 id='uniqueElement'> The Great Gatsby </h1>
  <h1> Harry Potter </h1>
  <h1> A Tale of two Cities </h1>
  <h1> Fourth Wing </h1>
</body>
</html>
```

In this example, the ID selector `#uniqueElement` is used to style the text within the `<h1>` element. The text inside the h1 will appear in blue and bold.

Just as you use the library book card to identify and handle a specific book, you use the ID selector in CSS to identify and style a specific HTML element.

Question

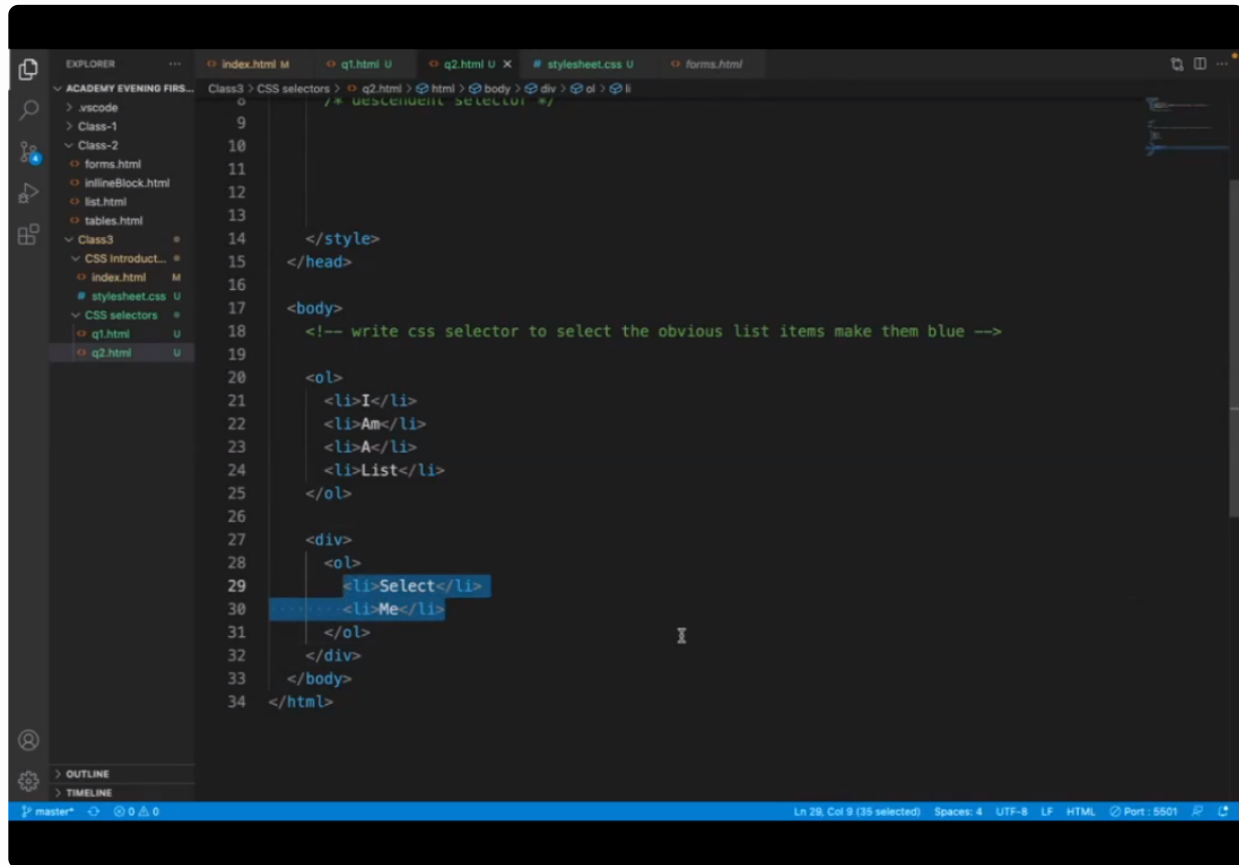
Which CSS selector is used to select all elements with the class "highlight"?

Choices

- ☐ `#highlight`
- ☐ `highlight`
- ☐ `*highlight*`
- ☒ `.highlight`

Descendent selectors

- Anything down the order can be termed as a descendent. For example, a child is the descendent of their Father as well as their GrandFather.
- Now, you need to select the obvious list as shown in the image and make it blue color.



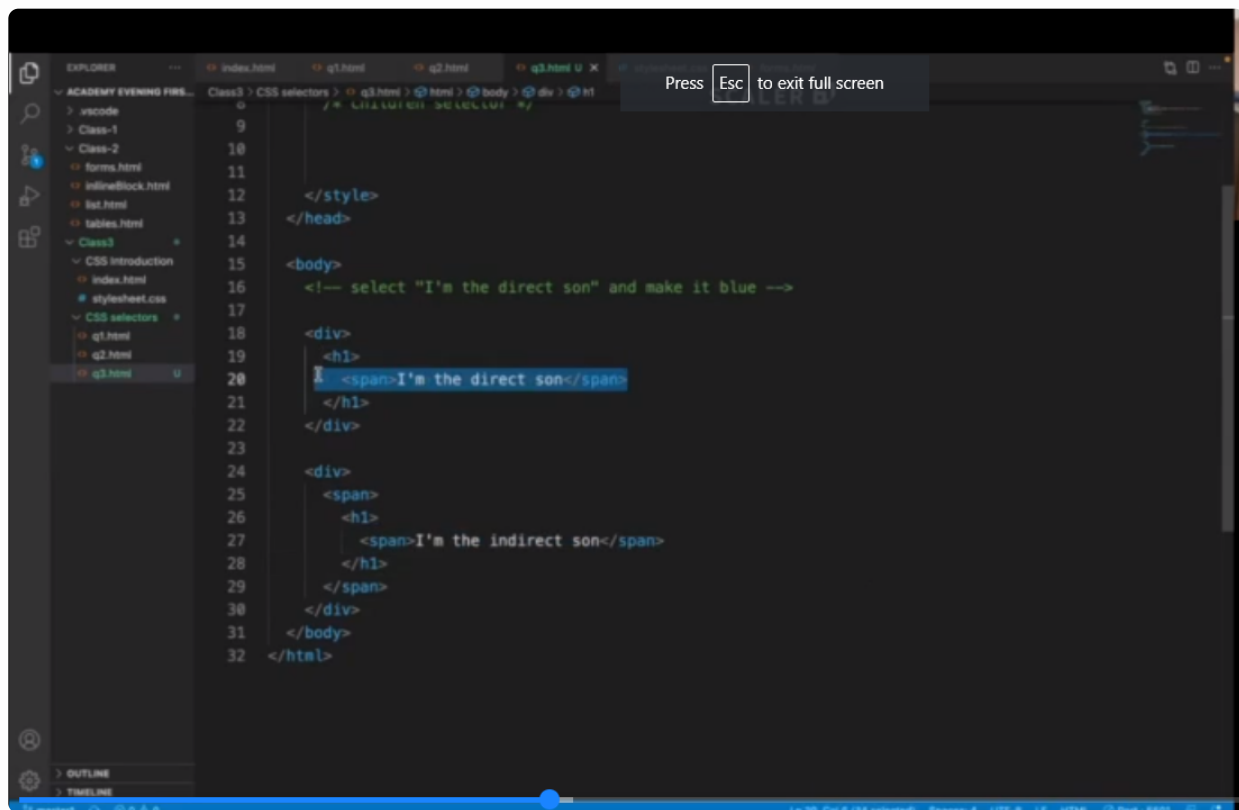
- For that first discuss the "**descent selector**".
- When you select `ol li` as the selector then all the elements will get blue. So you need to select the specific parents that are "div" as shown in the example.
- We will write the selector as:

```
1  div li{
2    color: blue;
3  }
```

Now moving to Children Selectors.

Children selectors

Let us take an example as shown below.



There are two span tags. Here we need to make the text "I am the direct son" blue using CSS.

- Use the greater than ">" symbol to use direct children.

```
1 | div > h1 > span{
2 |     color: blue;
3 |
4 | }
```

- It means that you are directing to apply CSS to the span element that is directly children to the **h1**.

Question

What does the CSS selector "ul > li" target?

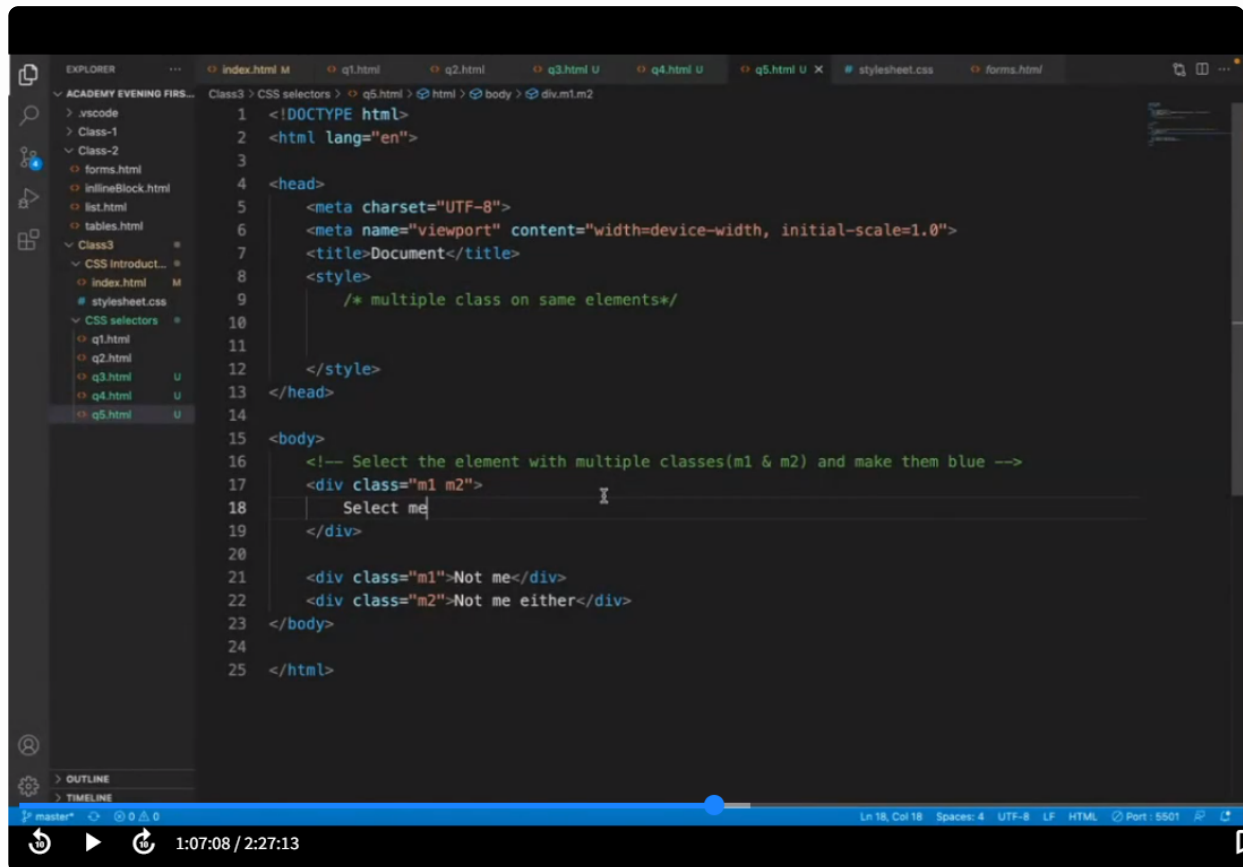
Choices

- ☐ All elements inside a
- ☒ All elements inside a

- ☐ Only the first inside any unordered list ()
- ☐ All list items regardless of their parent.

Question on multiple classes

Suppose a situation when there are two or more than two classes for a single element as shown below and you have to make some specific elements blue.



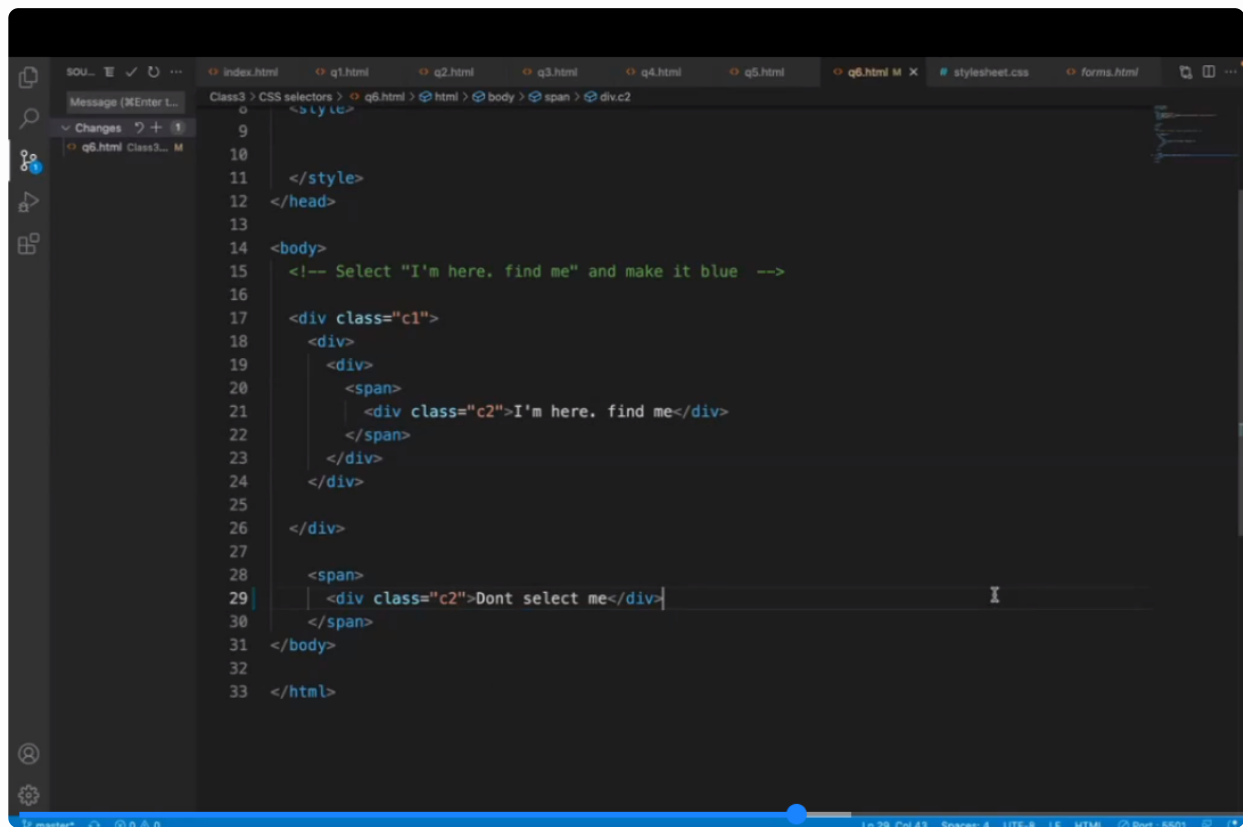
- Here we need to select multiple classes **m1** and **m2** and make them blue.
- Solution:

```
1 .m1.m2{    <!-- it is multiple selector -->
2     color : blue;
3 }
```

Here you will use multiple classes as shown in the example to apply the CSS to some specific elements in such cases.

Question on a combination of all the previously discussed selectors.

You have to make the text "**I'm here, find me**" blue as shown in the question image below. Feel free to use any selector or combination of selectors you want.



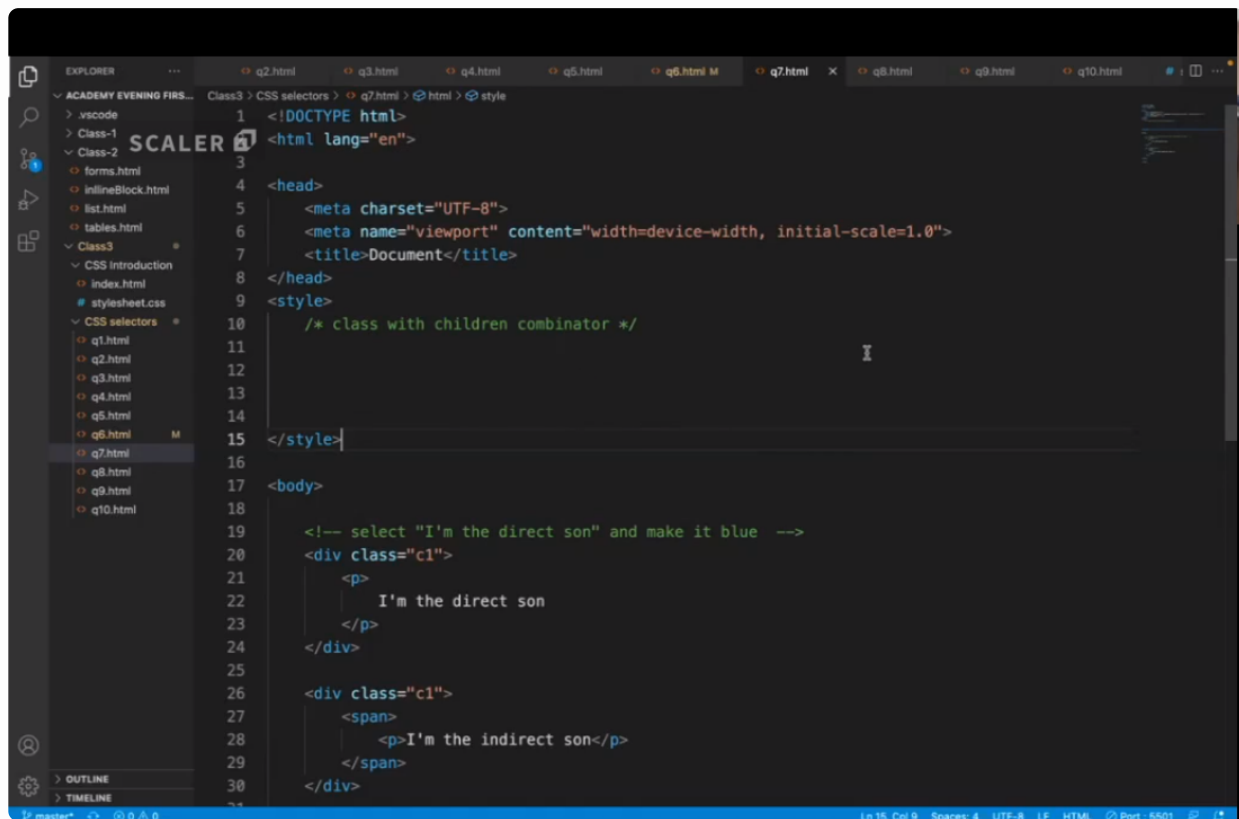
Give 3-4 minutes to the students to try then move ahead with the solution.

- **Solution:**

```
1 | .c1 .c2{
2 |     color: blue;
3 | }
```

Question on Class with Children Combinator

In this question you need to make the select and make the text "**I'm a direct son**" blue.



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <style>
10  /* class with children combinator */
11
12
13
14 </style>
15
16 <body>
17
18   <!-- select "I'm the direct son" and make it blue -->
19   <div class="c1">
20     <p>
21       I'm the direct son
22     </p>
23   </div>
24
25   <div class="c1">
26     <span>
27       <p>I'm the indirect son</p>
28     </span>
29   </div>
30
```

- **Solution:**

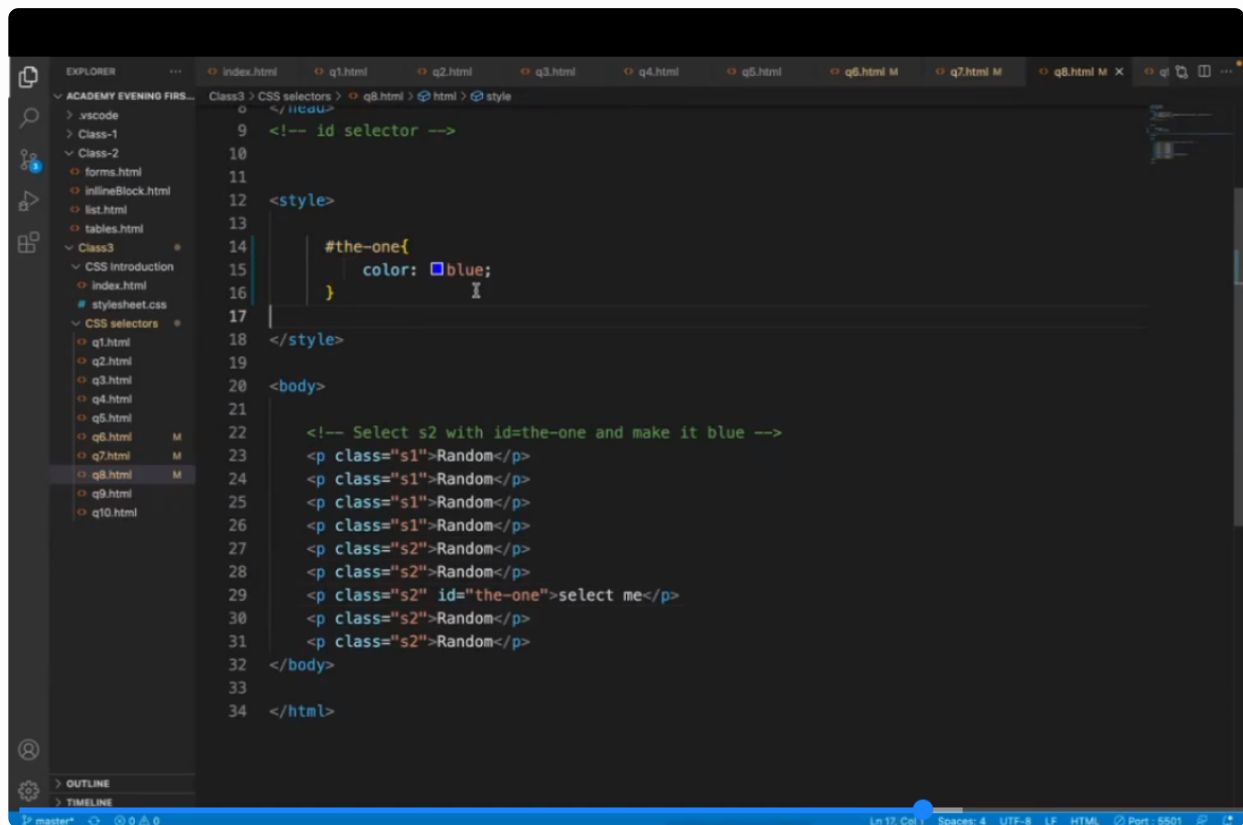
```
1 .c1>p{
2     color: blue;
3 }
```

Question on id selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element.
- We use "#" to address the id.

Ask students the difference between class and id of CSS. **Ans:** Elements can have the same classes but they can not have the same id.

Example

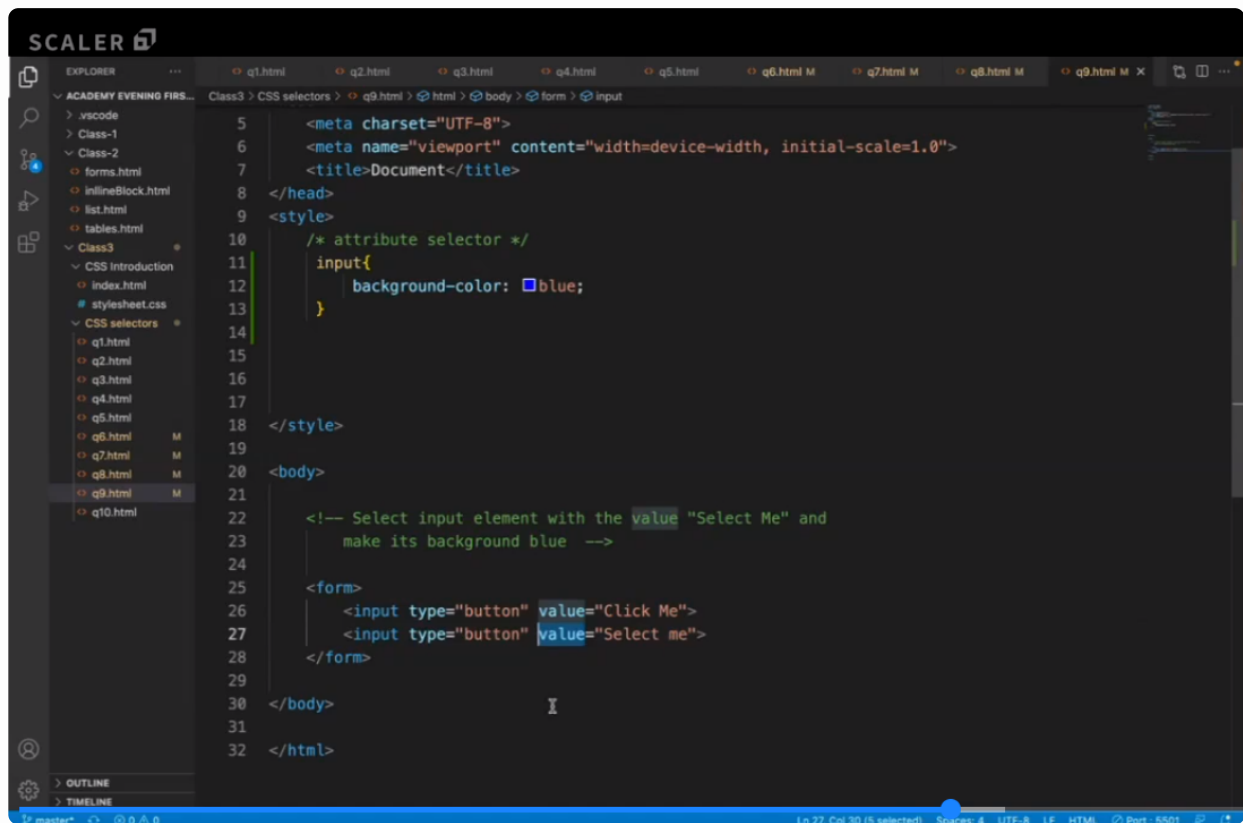


- You need to select **s2** with the "id=the-one" and make blue
- Solution:

```
1 | #the-one{
2 |     color: blue;
3 | }
```

Question on attribute selector

- CSS attribute selectors are used to select and style HTML elements with the specified attributes and values.
- Let us take an example as shown below. Here we need to select the button element and make the color blue.



- **Solution:**

```
1 | input[value="Select me"]{  
2 |     color: blue;  
3 | }
```

Here you can write the element like "input" and start the bracket "[]". Inside the bracket, you can write the attribute that you are selecting to apply CSS.

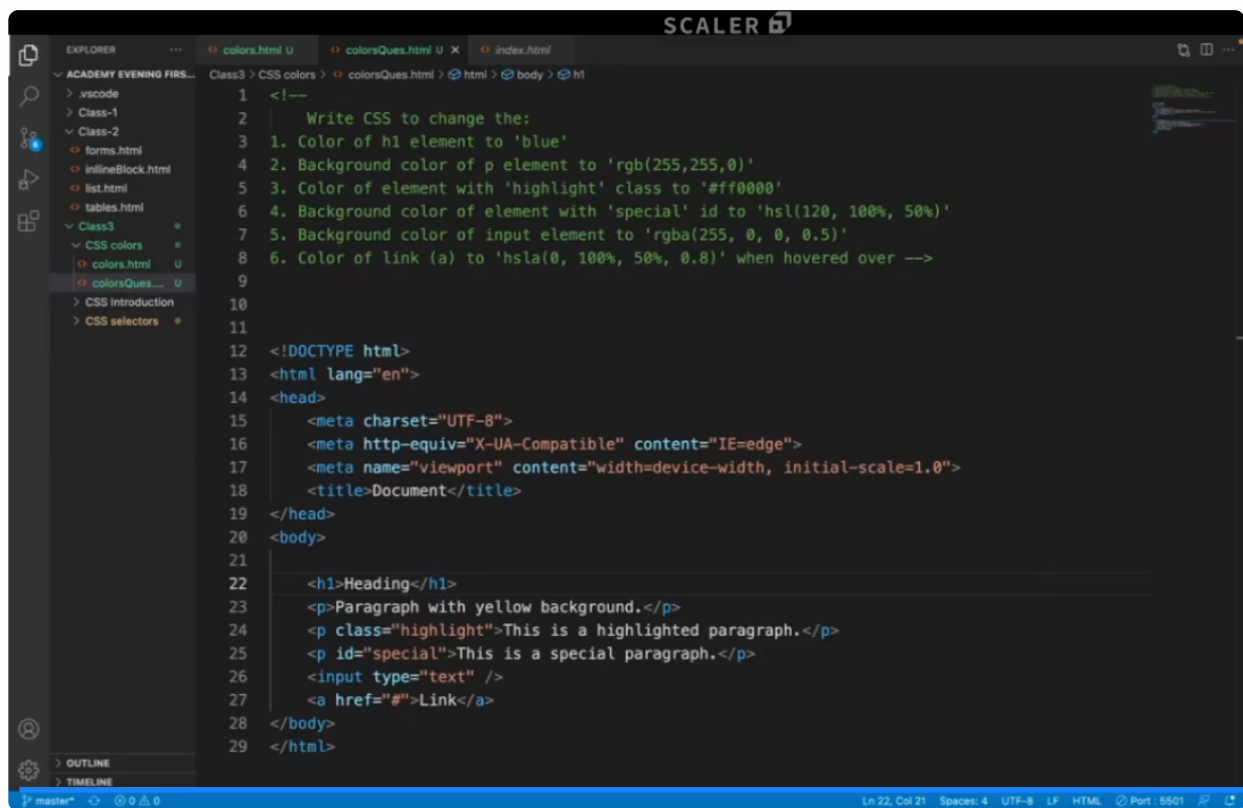
- That's how the attribute selector works.

Links to further explore css selectors

1. <https://css-tricks.com/css-selectors/>
2. https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_selectors

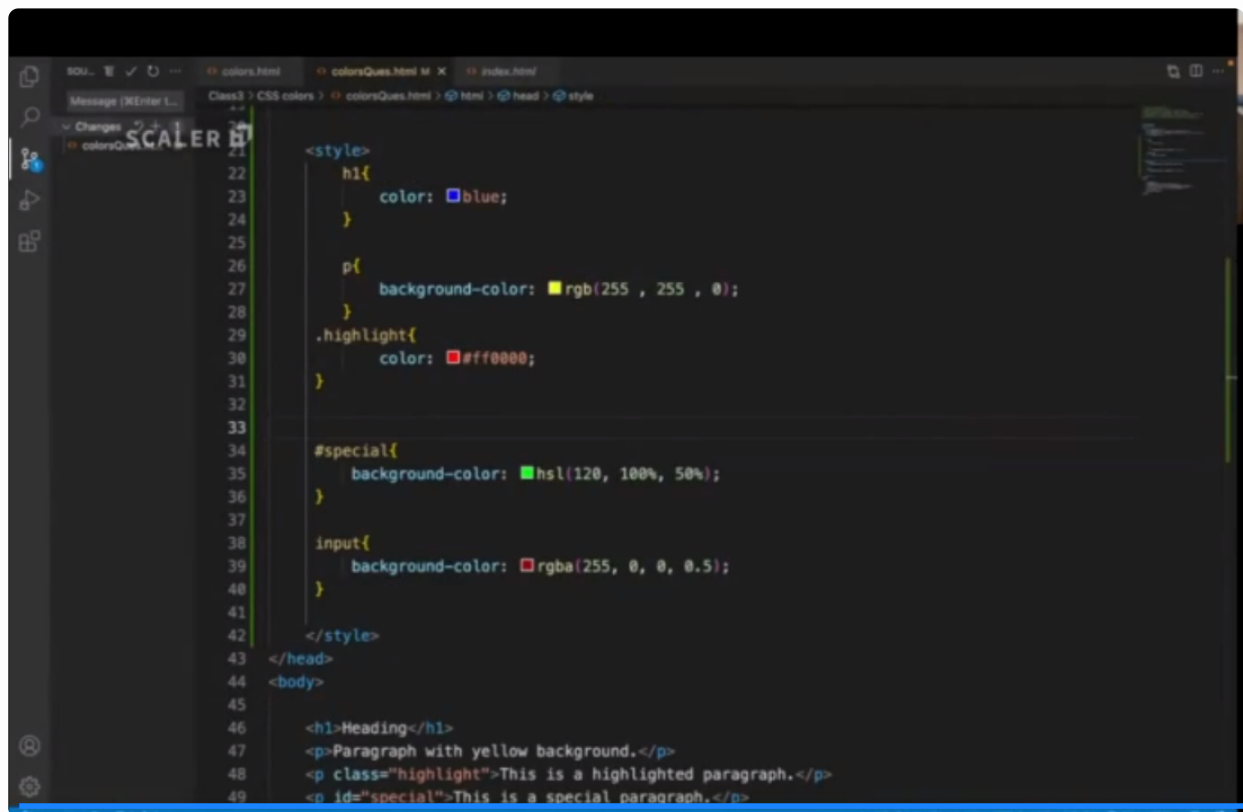
Final Question of the session

Now, let us summarize the session and use all the previous selectors and color properties to solve this problem shown below:



```
1  <!--
2      Write CSS to change the:
3      1. Color of h1 element to 'blue'
4      2. Background color of p element to 'rgb(255,255,0)'
5      3. Color of element with 'highlight' class to '#ff0000'
6      4. Background color of element with 'special' id to 'hsl(120, 100%, 50%)'
7      5. Background color of input element to 'rgba(255, 0, 0, 0.5)'
8      6. Color of link (a) to 'hsla(0, 100%, 50%, 0.8)' when hovered over -->
9
10
11
12  <!DOCTYPE html>
13  <html lang="en">
14  <head>
15      <meta charset="UTF-8">
16      <meta http-equiv="X-UA-Compatible" content="IE=edge">
17      <meta name="viewport" content="width=device-width, initial-scale=1.0">
18      <title>Document</title>
19  </head>
20  <body>
21
22      <h1>Heading</h1>
23      <p>Paragraph with yellow background.</p>
24      <p class="highlight">This is a highlighted paragraph.</p>
25      <p id="special">This is a special paragraph.</p>
26      <input type="text" />
27      <a href="#">Link</a>
28  </body>
29  </html>
```

- **Solution:** First, we will add a style tag and then write all the CSS provided in the question like this:



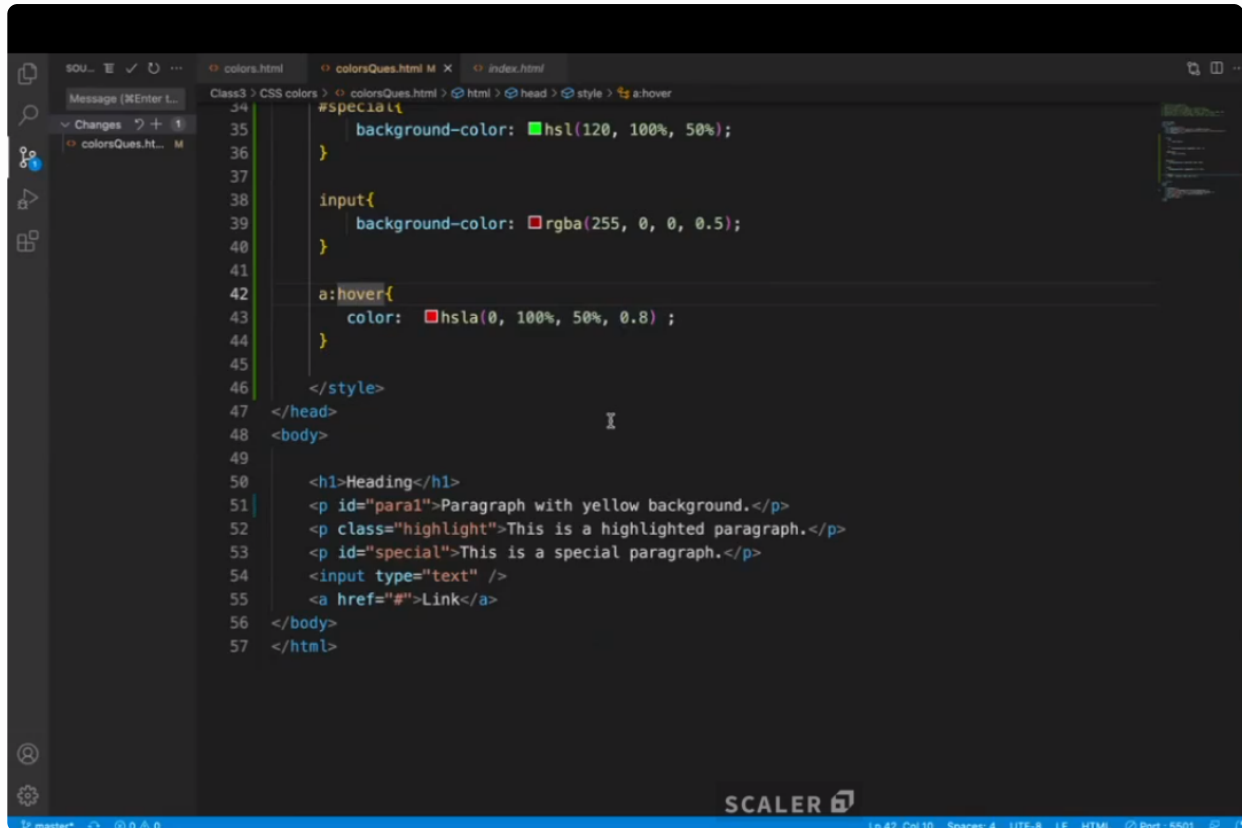
```
21  <style>
22      h1{
23          color: blue;
24      }
25
26      p{
27          background-color: rgb(255, 255, 0);
28      }
29      .highlight{
30          color: #ff0000;
31      }
32
33
34      #special{
35          background-color: hsl(120, 100%, 50%);
36      }
37
38      input{
39          background-color: rgba(255, 0, 0, 0.5);
40      }
41
42  </style>
43  </head>
44  <body>
45
46      <h1>Heading</h1>
47      <p>Paragraph with yellow background.</p>
48      <p class="highlight">This is a highlighted paragraph.</p>
49      <p id="special">This is a special paragraph.</p>
```

Coming to the point 6 of the question. Let us first see what is pseudo selector.

Pseudo selector

- When you hover over any element and it changes its behavior, it is known as a Pseudo selector.
- For example, if you go on a sign-in button it becomes popped up.

```
1 a:hover{
2     color : hsla(100%, 15%, 25%, 0.6)
3 }
```



Background color in CSS.

- It works the same as the color.
- You can add CSS code for background color as :

```
1 h1{
2     background-color : blue;
3 }
```

Example:

❗ Image Not Showing
Possible Reasons

- The image was uploaded to a note which you don't have access to
- The note which the image was originally uploaded to has been deleted

[Learn More →](#)

Here you will select the class namely "background" and apply CSS to them. It will be reflected to all the **h1s** inside that div tag.

`color` is used to change the text color but `background-color` is used to change the color area behind that text.

- You can also use an image as the text background. To do this, you need to provide the image URL in the CSS as shown below:

```
1 | h1{
2 |     background-image : url(" url_of_that_specific_image.jpg ");
3 | }
```

You can set the background size as:

```
1 | h1{
2 |     background-image : url(" url_of_that_specific_image.jpg ");
3 |     background-size : 400px;
4 | }
```

- In the CSS background property, when it finds any empty space, it starts to repeat its element. You can use the property `background : no-repeat;` to avoid it.
- -You can customize the repetition of the BG image using the X and Y axis.
- You can customize various aspects of the font such as size, look, and many dynamics of it. Let us see them one by one.
- First, create a file as shown below:

❗ Image Not Showing

Possible Reasons

- The image was uploaded to a note which you don't have access to
- The note which the image was originally uploaded to has been deleted

[Learn More →](#)

Then we will apply CSS to the fonts. First, let us see font family.

Font family

- The different types of font available in the CSS are termed as font family.
- The font-family property can hold several font names as a "**fallback**" system. If the browser does not support the first font, it tries the next font.
- Font family are like "times", "courier", "arial". If you want to use a single font from this, you can just remove the other font styles from the font family.
- Syntax:

```
1 | .heading_1{  
2 |     font-family: your_desired_font  
3 | }
```

Font weight

- This font property is used to decide the intensity of the font.
- It's value ranges from lightest **100** to boldest **800**.
- Syntax:

```
1 | font-weight: 800;
```

Font size

- It refers to the size of the text.
- Syntax:

```
1 | font-size: xx-larger;
```

```
1 | font-size: 50%;
```

- There is also a font-style that you can choose to customize your font like bold, italic, etc.

Refer to this resource for further reading on font styling:-

<https://developer.mozilla.org/en-US/docs/Web/CSS/font>

Google font

- If you are not unable to get the font requirement from the above font properties, then you can use the Google font option.
- You just need to go to the Google search and search "**google font**". Refer website: `https://fonts.google.com/`.
- Here you can find and choose the font. Click over the font and you will get a lot of options to customize that font.
- Click on "**select font_style**" and select the link there.
- Then paste it on your HTML file head tag as shown below.

❗ Image Not Showing

Possible Reasons

- The image was uploaded to a note which you don't have access to
- The note which the image was originally uploaded to has been deleted

[Learn More →](#)

- Now copy the font family from the Google font website and you can use it in the CSS.
- Since you have imported the Google fonts using CDN till now. Now you can just copy the font family from Google font and paste it at the font family section in the link of the head tag.
- Font and text can be seen as similar but they are not.
- Font focus on the look and styling but Text is the internal working of your text such as spacing, line width, etc.
- First, create a file of HTML is shown below

❗ Image Not Showing

Possible Reasons

- The image was uploaded to a note which you don't have access to
- The note which the image was originally uploaded to has been deleted

[Learn More →](#)

First, let us see the text alignment property

Text Alignment

- It is used to make the alignment of your text like from the left, from the right, or in the center.
- Syntax:

```
1 | .heading_1{  
2 |     text-align : left;  
3 | }  
4 |
```

Text Decoration

- It is used to make your text attractive using some properties.
- These are oval-line, line-through, underline, etc.
- Syntax:

```
1 | .heading_1{  
2 |     text-decoration : oval-line;  
3 | }
```

Word Spacing

- It defines how many spaces are there between any two consecutive words.
- It is defined using the pixel values.
- Syntax:

```
1 | .heading_1{  
2 |     text-decoration : oval-line;  
3 |     word-spacing : 100px;  
4 | }
```

Line Height

- It defines the height between two consecutive lines.
- Syntax:

```
1 | .heading_1{  
2 |     text-decoration : oval-line;  
3 |     word-spacing : 100px;  
4 |     line-height : 100px;  
5 | }
```

These are all text and font properties.

That's all for today's session.

The doubt session begins!

Further Reading - Comprehensive Guide to CSS Selectors

Introduction

CSS (Cascading Style Sheets) selectors play a crucial role in styling web documents. They enable developers to target specific HTML elements and apply styles to them. In this guide, we'll explore various CSS selectors, accompanied by practical examples.

1. Universal Selector

The universal selector `*` selects all elements on a webpage. It can be used to apply a style globally.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    * {
      margin: 0;
      padding: 0;
    }
  </style>
  <title>Universal Selector Example</title>
</head>
<body>
  <!-- All elements will have no margin or padding -->
  <div>Hello, World!</div>
</body>
</html>
```

2. Type Selector

The type selector selects elements based on their tag names. For example, `p` targets all `<p>` elements.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
  p {
    color: blue;
  }
</style>
<title>Type Selector Example</title>
</head>
<body>
  <p>This is a paragraph with blue text.</p>
  <div>This won't be affected.</div>
</body>
</html>

```

3. Class Selector

The class selector targets elements with a specific class attribute.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    .highlight {
      background-color: yellow;
    }
  </style>
  <title>Class Selector Example</title>
</head>
<body>
  <p class="highlight">This paragraph has a yellow background.</p>
  <p>This paragraph won't be highlighted.</p>
</body>
</html>

```

4. ID Selector

The ID selector targets a specific element with a unique ID attribute.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    #main-heading {
      font-size: 24px;
    }
  </style>
  <title>ID Selector Example</title>
</head>
<body>
  <h1 id="main-heading">This heading has a larger font size.</h1>

```

```
<p>This won't be affected.</p>
</body>
</html>
```

5. Attribute Selector

The attribute selector targets elements based on their attributes.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    input[type="text"] {
      border: 1px solid #ccc;
    }
  </style>
  <title>Attribute Selector Example</title>
</head>
<body>
  <input type="text" placeholder="Type something here">
  <input type="password" placeholder="Password">
</body>
</html>
```

6. Child Selector

The child selector (>) selects direct children of a specified element.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    ul > li {
      list-style-type: square;
    }
  </style>
  <title>Child Selector Example</title>
</head>
<body>
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <ul>
      <li>Sub-item 1</li> <!-- This won't have square bullets -->
      <li>Sub-item 2</li> <!-- This won't have square bullets -->
    </ul>
  </ul>
</body>
</html>
```


7. Descendant Selector

The descendant selector (whitespace) selects all descendants of a specified element, regardless of their depth.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    article p {
      font-style: italic;
    }
  </style>
  <title>Descendant Selector Example</title>
</head>
<body>
  <article>
    <p>This paragraph is italicized.</p>
    <div>
      <p>This paragraph is also italicized.</p>
    </div>
  </article>
</body>
</html>
```

8. Adjacent Sibling Selector

The adjacent sibling selector (`+`) selects an element that is immediately preceded by a specified element.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    h2 + p {
      font-weight: bold;
    }
  </style>
  <title>Adjacent Sibling Selector Example</title>
</head>
<body>
  <h2>Heading 2</h2>
  <p>This paragraph has bold text.</p>
  <p>This paragraph won't be affected.</p>
</body>
</html>
```

9. General Sibling Selector

The general sibling selector (`~`) selects all elements that are siblings of a specified element.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    h2 ~ p {
      color: green;
    }
  </style>
  <title>General Sibling Selector Example</title>
</head>
<body>
  <h2>Heading 2</h2>
  <p>This paragraph has green text.</p>
  <p>This paragraph also has green text.</p>
</body>
</html>
```

Advanced CSS Selectors

1. `:hover` , `:active` , `:focus` Pseudo-classes

Explanation:

- `:hover` : Styles a button when the mouse cursor is over it.
- `:active` : Applies styles to the button when it is being clicked.
- `:focus` : Adds a green border to an input element when it has keyboard focus.

HTML and CSS:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    button:hover {
      background-color: lightblue;
    }

    button:active {
      transform: scale(0.9);
    }

    input:focus {
      border: 2px solid green;
    }
  </style>
```

```
</head>
<body>
  <button>Hover me</button>
  <input type="text" placeholder="Focus me">
</body>
</html>
```

2. `:nth-child()` Pseudo-class

Explanation:

- `:nth-child(odd)` : Styles odd list items with a light gray background.
- `:nth-child(even)` : Styles even list items with a light blue background.

HTML and CSS:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    li:nth-child(odd) {
      background-color: lightgray;
    }

    li:nth-child(even) {
      background-color: lightblue;
    }
  </style>
</head>
<body>
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
  </ul>
</body>
</html>
```

3. `:not()` Pseudo-class

Explanation:

- `:not(.special)` : Styles paragraphs that do not have the class "special" with a dark gray color.

HTML and CSS:

```
<!DOCTYPE html>
<html lang="en">
```

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    p:not(.special) {
      color: darkgray;
    }
  </style>
</head>
<body>
  <p>This is a regular paragraph.</p>
  <p class="special">This is a special paragraph.</p>
</body>
</html>

```

4. :first-child, :last-child, :only-child Pseudo-classes

Explanation:

- `:first-child` : Makes the first list item bold.
- `:last-child` : Changes the color of the last list item to red.
- `:only-child` : Makes the only paragraph inside its container italic.

HTML and CSS:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    li:first-child {
      font-weight: bold;
    }

    li:last-child {
      color: red;
    }

    p:only-child {
      font-style: italic;
    }
  </style>
</head>
<body>
  <ul>
    <li>First</li>
    <li>Second</li>
    <li>Last</li>
  </ul>

  <p>First paragraph</p>
  <p>Second paragraph</p>
  <p>Last paragraph</p>

```

```
</body>
</html>
```

5. `:nth-of-type()` Pseudo-class

Explanation:

- `:nth-of-type(2n)` : Styles every second paragraph with a light gray background.

HTML and CSS:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    p:nth-of-type(2n) {
      background-color: lightgray;
    }
  </style>
</head>
<body>
  <p>First paragraph</p>
  <p>Second paragraph</p>
  <p>Third paragraph</p>
</body>
</html>
```

6. `:first-of-type` , `:last-of-type` , `:only-of-type` Pseudo-classes

Explanation:

- `:first-of-type` : Styles the first heading with a green color.
- `:last-of-type` : Increases the font size of the last paragraph.
- `:only-of-type` : Adds a blue border to the only div containing a paragraph.

HTML and CSS:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    h2:first-of-type {
      color: green;
    }

    p:last-of-type {
```

```

        font-size: 18px;
    }

    div:only-of-type {
        border: 1px solid blue;
    }
</style>
</head>
<body>
    <h2>First Heading</h2>
    <h2>Second Heading</h2>

    <p>This is a paragraph in a div.</p>
    <div>
        <p>Another paragraph</p>
    </div>
</body>
</html>

```

7. `:checked` , `:disabled` Pseudo-classes

Explanation:

- `:checked` : Styles a checked checkbox with a yellow background.
- `:disabled` : Reduces the opacity of a disabled checkbox.

HTML and CSS:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        input:checked {
            background-color: yellow;
        }

        input:disabled {
            opacity: 0.5;
        }
    </style>
</head>
<body>
    <input type="checkbox" checked> Checked
    <input type="checkbox" disabled> Disabled
</body>
</html>

```

8. `:before` , `:after` Pseudo-elements

Explanation:

- `:before` : Inserts ">> " before the content of each paragraph with a blue color.
- `:after` : Inserts " <<" after the content of each paragraph with a red color.

HTML and CSS:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    p:before {
      content: '>> ';
      color: blue;
    }

    p:after {
      content: ' <<';
      color: red;
    }
  </style>
</head>
<body>
  <p>This is a paragraph.</p>
</body>
</html>
```

Feel free to use these examples as standalone HTML files to understand and experiment with each advanced CSS selector.

Further Resources

4. Resources for Further Learning:

- [Mozilla Developer Network \(MDN\)](#): A comprehensive CSS resource.
- [CSS Tricks](#): Articles, tips, and tricks for CSS.

5. Practice Resources:

- [CSS Diner](#): A game for learning CSS selectors interactively.