1. Why are CSS selectors used in HTML?

CSS selectors play a crucial role in web development, allowing developers to *target specific HTML elements* and apply styling rules. we'll explore the five categories of CSS selectors, from simple selectors to pseudo-element selectors, and provide examples

Why Selector in CSS?

1. They allow developers to target specific HTML elements and apply styling rules to them.
2. They provide a way to customize the appearance of a website and make it visually appealing and user-friendly.

2. What are the different types of Selectors in CSS?

Types of Selectors

We can divide CSS selectors into 5 categories:

1. Simple selectors.

2. Combinators.

3. Attribute selectors.

4. Pseudo-class selectors.

5. Pseudo-elements selectors.

NOTE: CSS Selectors cannot be used with inline CSS.

We will be using *internal CSS* to demonstrate CSS Selectors.

In this blog, we are going to discuss Simple selectors. A simple selector is the most basic type of selector in CSS and it selects elements based on their tag name, class, or ID.

There are five types of simple selectors in CSS:

1. Universal Selector

2. Element Selector
3. Class Selector
4. ID Selector
5. Selector list

Universal Selector:

The styles applied to the universal selector will be applied to every element on the page.

Asterisk (*): symbol denotes the universal selector in CSS.

```css
/* Universal selector */

/* Target all elements on the page */

*{

   margin:0;
   padding:0;

}
```

The universal selector, denoted by the asterisk (*), targets all elements on a web page. In this particular example, the code sets the margin and padding of all elements to 0, effectively removing any default spacing or indentation that may be present.

Element Selector:

The element selector targets elements based on their tag name.

```
/* Element Slector */

/* Target all <h1> elements on the page */

/* h1{
    color:#808081;




}
```

An element selector targets all instances of a particular HTML element on a web page. In this example, the code targets all `<h1>` elements and sets their color to `#808081`.

Class Selector:

The class selector targets elements based on given the class value in their class attribute.

To apply a class selector to an element, you need to add a period (.) followed by the class name in your CSS code.

```
/* Class Selector  */
```

```
.para{

color:rgba(255, 38, 0, 0.836);

}
```

```
.hello{

    font-size: 80px;

}
```

The first class selector targets all elements with the class name “*para*” and sets their color to a *shade of red*. To apply this class to an HTML element, you would need to add the class name “para” to the element’s class attribute. For example:

```
<p class="para">Lorem ipsum dolor sit amet consectetur adipisicing
elit.</p>
```

The second class selector targets all elements with the class name "hello" and sets their font size to 80 pixels. To apply this class to an HTML element, you would need to add the class name "hello" to the element's class attribute.For example:

```
<h1 class="hello">Hello World!</h1>
```

Let's check your knowledge, shall we? Can you tell output for paragraph element below:

```
<p class="para hello">Lorem ipsum dolor sit amet.</p>
```

Above paragraph element has the class "para" and "hello", which means it belongs to both the "para" class and the "hello" class.

So the output of this HTML code will be a paragraph element with the following styles applied:

- The text color will be a shade of red, specified by the “color” property in the “.para” class.
- The font size will be 80 pixels, specified by the “font-size” property in the “.hello” class.

These styles are applied to the paragraph element because it has both the “para” and “hello” classes, and the CSS rules defined for those classes will apply to any element that has them.

ID selector:

ID selectors target an element based on its unique ID attribute value, providing **higher specificity** compared to class selectors or other selectors. An element can only have one ID attribute, which must be unique throughout the HTML document.

To apply an ID selector to an element, you need to add a hash (#) followed by the ID name in your CSS code. For example, if you want to target an element with ID=”id1”, you can use the ID selector like this:

```
/* Target the element with ID="id1" */

#id1{

   color: whitesmoke;

   background-color:black;



}
```

Html code

```
<p class="id="id1">Lorem ipsum dolor sit amet.</p>
```

Understand Higher specificity through code:

```html
<p class="para hello" id="id1">Lorem ipsum dolor sit amet.</p>
```

Analyze output of above Html code :

The `id` attribute with the value of "id1" is unique to this element, making the selector targeting this ID even more specific than a class selector targeting a class shared among multiple elements.

Styles applied through the ID selector will take precedence over styles applied through the class selector.

Selector list:

A selector list allows you to apply styles to multiple selectors at once. To create a selector list, you need to separate the selectors with a comma. For example, if you want to apply the same style to all <h1> and <h2> elements on the page, you can use a selector list like this:

```css
/* Target all <h1> and <h2> elements on the page */
```

```
h1, h2 {

 font-weight: bold;

}
```

3. How do you include external fonts and apply them?

Google Font

Google Fonts is a free, open-source library of over 800 fonts designed specifically for the web. It is a popular source for web fonts and can be easily integrated into web projects by including a link in the HTML file. The fonts are customizable and can be previewed and compared using Google Fonts' range of useful tools. Overall, Google Fonts is a convenient and reliable resource for finding and using high-quality fonts on the web.

How to add google font?

To add Google Fonts to your web project, follow these steps:

1. Browse the Google Fonts library to find the font(s) you want to use.
2. Select the fonts and customize the styles, such as the font weight, style, and size.
3. Generate the code for adding the fonts to your project by clicking on the "Embed" button and copying the code provided.
4. Use the fonts in your CSS by adding the code to the head section of your HTML file or by linking to the Google Fonts stylesheet in your CSS file to specify families.

To embed a font, copy the code into the `<head>` of your html

index.html file

```
<head>
```

```
    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <!-- roboto fonts -->

    <link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link
href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,400;1,300&display=swap" rel="stylesheet">

<!-- external style sheet -->

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

    <title>Fonts</title>
    </head>
```

In the case of fonts, the <link> tag in the head section of the HTML file typically links to a stylesheet that contains the `@font-face` rule, which specifies the font files to be used and how they should be rendered on the webpage. By linking to the stylesheet in the head section, the font can be used throughout the HTML document without having to repeat the code for linking to the font files in every part of the document.

In above code ,the @font-face rule is included in the link to the Google Fonts stylesheet, which specifies the code you have provided is a link to include the Roboto font family from Google Fonts in a web project. The code includes three lines:

1. The first line is a preconnect link to the Google Fonts domain, which helps to optimize the loading time of the font file.
2. The second line is a preconnect link to the Google Fonts domain with a “crossorigin” attribute, which allows for secure data transfer between domains.
3. The third line is the *actual link to the Roboto font family* with specified styles, such as italic and weight, and the “display” property set to “swap” to ensure that text appears as quickly as possible on the website.

style.css file

Linking to the Google Fonts stylesheet in your CSS file to specify families.

```css
body{

    font-family: 'Roboto', sans-serif;
}
```

This is a CSS code snippet that sets the font family for the entire body of an HTML document to “Roboto”, a popular sans-serif font.

In this example, the *font family* “Roboto” is specified as the first choice. If Roboto is not available, the browser will use any available *generic* sans-serif font as a fallback.

By following these steps, you can easily add high-quality, customizable fonts to your web project using Google Fonts.