CSS INTRODUCTION

1. What is CSS and why use it?

Ans - CSS stands for cascading style sheets. In short, CSS is a design language that makes a website look more appealing than just plain or uninspiring pieces of text. Whereas HTML largely determines textual content, CSS determines the visual structure, layout, and aesthetics. HTML is a markup language, and CSS is a style sheet language. Think “look and feel” when you think CSS.

There are a number of benefits of CSS, including:

**1) Faster Page Speed**

More code means slower page speed. And CSS enables you to use less code. CSS allows you to use one CSS rule and apply it to all occurrences of a certain tag within an HTML document.

**2) Better User Experience**

CSS not only makes web pages easy on the eye, but it also allows for user-friendly formatting. When buttons and text are in logical places and well organized, user experience improves.

**3) Quicker Development Time**

With CSS, you can apply specific formatting rules and styles to multiple pages with one string of code. One cascading style sheet can be replicated across several website pages. If, for instance, you have product pages that should all have the same formatting, look, and feel, writing CSS rules for one page will suffice for all pages of that same type.

**4) Easy Formatting Changes**

If you need to change the format of a specific set of pages, it’s easy to do so with CSS. There’s no need to fix every individual page. Just edit the corresponding CSS stylesheet and you’ll see changes applied to all the pages that are using that style sheet.

1. What are the different ways of adding CSS to your HTML file?

Ans- CSS can be added to HTML documents in 3 ways:

* **Inline** - by using the style attribute inside HTML elements
* **Internal** - by using a <style> element in the <head> section
* **External** - by using a <link> element to link to an external CSS file

The most common way to add CSS is to keep the styles in external CSS files.

**Inline CSS**

Inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the style attribute of an HTML element.

The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

Example

<h1 style="color:blue;">A Blue Heading</h1>  
  
<p style="color:red;">A red paragraph.</p>

**Internal CSS**

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the <head> section of an HTML page, within a <style> element.

The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the <p> elements to red. In addition, the page will be displayed with a "powderblue" background color:

Example

<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {background-color: powderblue;}  
h1   {color: blue;}  
p    {color: red;}  
</style>  
</head>  
<body>  
  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
  
</body>  
</html>

**External CSS**

An external style sheet is used to define the style for many HTML pages.

To use an external style sheet, add a link to it in the <head> section of each HTML page:

Example

<!DOCTYPE html>  
<html>  
<head>  
  <link rel="stylesheet" href="styles.css">  
</head>  
<body>  
  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
  
</body>  
</html>

The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

Here is what the "styles.css" file looks like:

"styles.css":

body {  
  background-color: powderblue;  
}  
h1 {  
  color: blue;  
}  
p {  
  color: red;  
}

1. What do you mean by Specificity in CSS?

Ans - If there are two or more CSS rules that point to the same element, the selector with the highest specificity value will "win", and its style declaration will be applied to that HTML element.

Think of specificity as a score/rank that determines which style declaration is ultimately applied to an element.

Look at the following examples:

**Example 1**

In this example, we have used the "p" element as selector, and specified a red color for this element. The text will be red:

<html>  
<head>  
  <style>  
    p {color: red;}  
  </style>  
</head>  
<body>  
  
<p>Hello World!</p>  
  
</body>  
</html>

**Example 2**

In this example, we have added a class selector (named "test"), and specified a green color for this class. The text will now be green (even though we have specified a red color for the element selector "p"). This is because the class selector is given higher priority:

<html>  
<head>  
  <style>  
    .test {color: green;}  
    p {color: red;}  
  </style>  
</head>  
<body>  
  
<p class="test">Hello World!</p>  
  
</body>  
</html>

**Example 3**

In this example, we have added the id selector (named "demo"). The text will now be blue, because the id selector is given higher priority:

<html>  
<head>  
  <style>  
    #demo {color: blue;}  
    .test {color: green;}  
    p {color: red;}  
  </style>  
</head>  
<body>  
  
<p id="demo" class="test">Hello World!</p>  
  
</body>  
</html>

**Example 4**

In this example, we have added an inline style for the "p" element. The text will now be pink, because the inline style is given the highest priority:

<html>  
<head>  
  <style>  
    #demo {color: blue;}  
    .test {color: green;}  
    p {color: red;}  
  </style>  
</head>  
<body>  
  
<p id="demo" class="test" style="color: pink;">Hello World!</p>  
  
</body>  
</html>

**Specificity Hierarchy**

Every CSS selector has its place in the specificity hierarchy.

There are four categories that define the specificity level of a selector:

1. **Inline styles** - Example: <h1 style="color: pink;">
2. **IDs** - Example: #navbar
3. **Classes, pseudo-classes, attribute selectors** - Example: .test, :hover, [href]
4. **Elements and pseudo-elements** - Example: h1, ::before