

CSS Introduction

What is CSS?

- * CSS stands for Cascading Style Sheets
- * CSS describes how HTML elements are to be displayed on screen.
- * It can control layout of multiple pages all at ones.
- * External style sheets are stored in CSS files.

Why use CSS?

It is used to define styles of your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Syntax

A CSS rule consists of a selector and a declaration block.

```
h1 { color : blue ; }
```

selector property value

Selectors in CSS

A CSS selector selects the HTML element(s) you want to style.

1- Simple Selector

1. Element Selector

2. Class Selector

3. ID Selector

2. Pseudo-class Selector

3. Multiple Selector

Element Selector

CSS element selector selects HTML elements based on the element name.

Ex - `p {`

`text-align: center;`

`color: red;`

`}`

Class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

Ex .center {

color: red;

}

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!
- ⇒ To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Ex #para1 {

color: red;

}

Pseudo-classes Selector

A pseudo-class is a keyword added to a selector that specifies a special state of the selected element(s).

For example :

- * Style an element when a user hovers the cursor over it.

- * Style visited and unvisited links differently.

Syntax

Selector: pseudo-class {

property: value;

}

Multiple Selector / Grouping Selector

A grouping selector selects all the HTML elements with the same style definitions.

⇒ It will be better to group the selectors, to minimize the code.

⇒ To group selectors, separate each selector with a (,) comma.

Ex `h1, h2, p {`
 `color: red;`
 `}`

Exploration Time (HW)

* Universal Selector

* Nested Selector

* Attribute Selector

Universal Selector

The universal selector (*) selects all HTML elements on the page.

Ex `* {`
 `color: red;`
 `}`

Nested Selector

Just like in HTML where you can have elements nested inside other elements, the same can be done in CSS.

Syntax

```
class1-sele class2-sele id-sele {  
    property: value;  
}
```

Ex table tr th {
 background-color: red;
}

Attribute Selector

The [attribute] selector is used to select elements with a specified attribute.

Ex a[target] {
 background-color: yellow;
}

How to add styling in HTML?

* Inline CSS

* Internal CSS

* External CSS

Inline CSS

⇒ To style an HTML element, you can add the style attribute directly to the opening tag.

⇒ To use inline styles, add the style attribute to the relevant ~~element~~ element.

⇒ Inline styles should be avoided at all costs because it makes it impossible to alter styles from an external stylesheet.

Ex `<p style="color:red;"> I am Rishabh. </p>`

Internal CSS / Style Tag

An internal style sheet may be used if one single HTML page has a unique style.

HTML allows us to write CSS code inside the `<style>` element, inside the head section.

Ex `<head>`

`<style>`

`h1 {`

`color:red;`

`}`

</style>
</head>

External CSS

⇒ When the HTML & CSS code are in separate files, they must be linked.

⇒ You can use the <link> element to link HTML and CSS files together. The <link> element must be placed within the head of the HTML file.

Ex <link rel="stylesheet" href="style.css">

Specificity

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

Specificity Hierarchy

Every CSS selector has its place in the specificity hierarchy.

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

- * Inline styles
- * IDs
- * Classes, pseudo-classes, attribute selectors
- * Elements and pseudo-elements

!important Rule

⇒ The **!important** rule in CSS is used to add more importance to a property/value than normal.

⇒ If you use the **!important** rule, it will override all previous styling ~~tag~~ rules.

⇒ It is good to know about the **!important** rule. However, do not use it unless you absolutely have to.

Ex **p {**
 color: red !important;

}