

CSS_1

Intro to CSS and selectors

Agenda

- intro to styling and css
- Anatomy of a css rule
- Applying styles
- styling text
- selectors of css

Introduction to CSS

CSS (Cascading Style Sheets): A language used to describe the presentation of a web page . CSS handles the look and feel of the web page, separating content from design.

Importance of CSS:

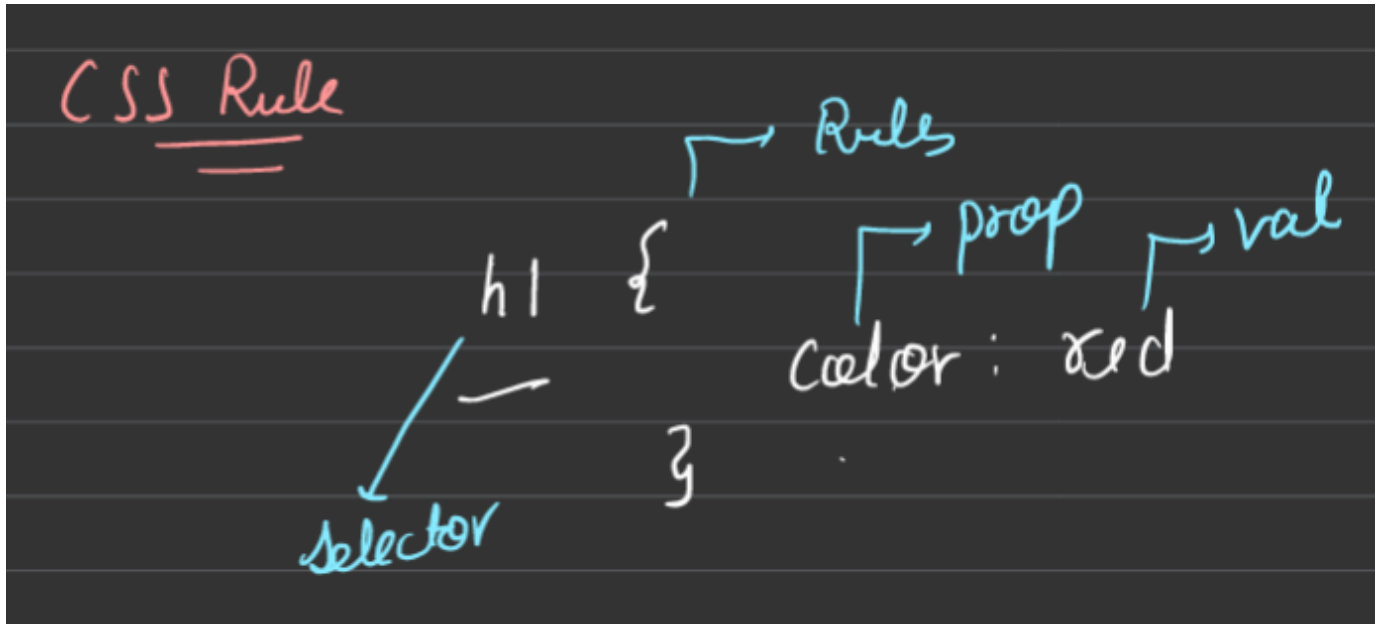
- **Styling Web Pages:** CSS allows you to style HTML elements, such as images, texts, links, and more.
- **Visual Styling:** Enhances the appearance of web pages, making them visually appealing and user-friendly.
- **Positioning:** CSS allows precise control over the placement of elements on the web page.
- **Layout Control:** CSS helps in arranging elements on the page.
- **Responsive Design:** CSS enables web pages to adapt to different screen sizes and devices.

Basic CSS Syntax

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.

- **Selector**: Specifies the HTML element to be styled.
- **Property**: Specifies the style attribute you want to change.
- **Value**: Specifies the value of the property.

For example,



Note : you can have multiple key : value in a given rule

Applying CSS

1. Inline CSS:

- Directly within an HTML element using the `style` attribute.

```
<p style="color: blue; font-size: 16px;">Hello World!</p>
```

2. Internal CSS:

- Within a `<style>` tag inside the `<head>` section of the HTML document.

```
<style>
  p {
    color: blue;
    font-size: 16px;
  }
</style>
```

```
}  
</style>
```

3. External CSS:

- In a separate `.css` file linked to the HTML document.

```
<!-- HTML -->  
<link rel="stylesheet" href="styles.css">
```

```
/* styles.css */  
p {  
    color: blue;  
    font-size: 16px;  
}
```

Note : You will be using external css usually because it makes css logic independent of html and is highly reusable

Styling Text

Properties for Styling Text

1. **Color**: Sets the color of the text.

```
color: blue;
```

2. **Background Color**: Sets the background color of an element.

```
background-color: lightblue;
```

3. **Font Family**: Specifies the font for the text.

```
font-family: Arial, sans-serif;
```

4. **Font Size**: Specifies the size of the font.

```
font-size: 16px;
```

5. **Font Style:** Specifies the style of the font (normal, italic, oblique).

```
font-style: italic;
```

6. **Font Weight:** Specifies the thickness of the font (normal, bold, bolder, lighter, or numeric values).

```
font-weight: bold;
```

7. **Text Decoration:** Adds decoration to the text (underline, overline, line-through, none).

```
text-decoration: underline;
```

8. **Line Height:** Sets the space between lines of text.

```
line-height: 1.5;
```

9. **Text Transform:** Controls the capitalization of text (uppercase, lowercase, capitalize).

```
text-transform: uppercase;
```

Introduction to CSS Selectors

CSS Selectors: Selectors are used to target HTML elements on which to apply styles. They are a fundamental part of CSS and determine which elements the style rules will affect.

Selectors

Basic Selectors

1. Universal Selector:

- Selects all elements.

```
* {  
  margin: 0;  
  padding: 0;  
}
```

2. Type Selector:

- Selects elements by their tag name.

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

3. Class Selector:

- Selects elements with a specific class attribute. Prefixed with a dot (**.**).

```
.container {  
  padding: 20px;  
}
```

4. ID Selector:

- Selects an element with a specific id attribute. Prefixed with a hash (**#**).

```
#header {  
  background-color: lightgray;  
}
```

5. Attribute Selector:

- Selects elements with a specific attribute.

```
[type="text"] {  
  border: 1px solid black;  
}
```

Grouped Code Example for Selectors

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    /* Universal Selector */
    * {
      margin: 0;
      padding: 0;
    }

    /* Type Selector */
    p {
      color: blue;
    }

    /* Class Selector */
    .container {
      padding: 20px;
    }

    /* ID Selector */
    #header {
      background-color: lightgray;
    }

    /* Attribute Selector */
    [type="text"] {
      border: 1px solid black;
    }
  </style>
  <title>Selectors Example</title>
</head>
<body>
  <div id="header">
    <h1>Header</h1>
  </div>
  <div class="container">
```

```
<p>Paragraph inside a container.</p>
<input type="text" placeholder="Type here">
</div>
<p>Another paragraph.</p>
</body>
</html>
```

Combinators

Types of Combinators

1. Descendant Combinator:

- Selects all elements that are descendants of a specified element.

```
div p {
  color: green;
}
```

2. Child Combinator:

- Selects all elements that are direct children of a specified element.

```
div > p {
  color: orange;
}
```

3. Adjacent Sibling Combinator:

- Selects the element that is directly after a specified element.

```
h1 + p {
  color: red;
}
```

4. General Sibling Combinator:

- Selects all elements that are siblings of a specified element.

```
h1 ~ p {
  color: purple;
}
```

```
}
```

Grouped Code Example for Combinators

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    /* Descendant Combinator */
    div p {
      color: green;
    }

    /* Child Combinator */
    div > p {
      color: orange;
    }

    /* Adjacent Sibling Combinator */
    h1 + p {
      color: red;
    }

    /* General Sibling Combinator */
    h1 ~ p {
      color: purple;
    }
  </style>
  <title>Combinators Example</title>
</head>
<body>
  <div>
    <p>Paragraph inside a div (descendant).</p>
    <p>Another paragraph inside the same div (child).</p>
  </div>
  <h1>Heading 1</h1>
  <p>Paragraph next to h1 (adjacent sibling).</p>
</body>
</html>
```



```
<p>Another paragraph next to h1 (general sibling).</p>  
</body>  
</html>
```