Introduction of CSS

CSS, or Cascading Style Sheets, is a style sheet language used for describing the presentation of a document written in a markup language. It enables the separation of presentation and content, so that the same HTML document can be presented in different ways depending on the browser or device used to view it.

CSS is used to control the appearance of web pages, including the following:

- Fonts: CSS can be used to specify the font family, size, and color of text.
- Colors: CSS can be used to specify the background color, text color, and colors of other elements on a web page.
- Layout: CSS can be used to control the layout of a web page, such as the size and position of elements.
- Decorations: CSS can be used to add decorations to web pages, such as borders, shadows, and gradients.

CSS can be used to style web pages in a variety of ways, including:

- Internal style sheets: Internal style sheets are embedded in the HTML document itself.
- External style sheets: External style sheets are stored in separate files and linked to the HTML document.
- Inline style sheets: Inline style sheets are applied to individual HTML elements.

CSS is a powerful tool that can be used to create visually appealing and functional web pages. It is a relatively easy language to learn, and there are many resources available to help you get started.

Here are some of the benefits of using CSS:

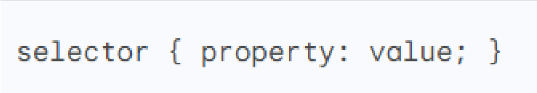
- Increased flexibility: CSS allows you to change the appearance of your web pages without having to change the underlying HTML code. This makes it easy to create different versions of your pages for different devices or screen sizes.
- Improved readability: CSS can be used to make your web pages more readable by using different fonts, colors, and layouts.

- Increased accessibility: CSS can be used to make your web pages more accessible to people with disabilities by using clear and concise text and providing alternative text for images.
- Reduced file size: CSS can be used to reduce the file size of your web pages by storing style information in separate files. This can improve the loading speed of your pages.

CSS Syntax

The basic syntax of CSS is as follows:

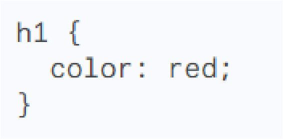
Code snippet



```
selector { property: value; }
```

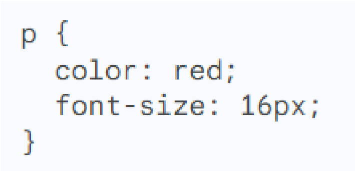
- `selector` is the HTML element that you want to style. It can be a tag name, a class name, or an ID.
- `property` is the name of the CSS property that you want to set. There are many different CSS properties, such as `color,` `font-size`, and `background-color.`
- `value` is the value of the CSS property. The value can be a number, a string, or a keyword.

For example, the following CSS code will set the color of all `h1` elements to red:



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

You can also style multiple properties in a single declaration. For example, the following CSS code will set the color and font size of all `p` elements:



```
p {
  color: red;
  font-size: 16px;
}
```

CSS declarations can be placed in a variety of locations, including:

- Internal style sheets: Internal style sheets are embedded in the HTML document itself. They are typically placed in the `<head>` section of the document.

- External style sheets: External style sheets are stored in separate files and linked to the HTML document. They are typically named with the `.css` file extension.
- Inline style sheets: Inline style sheets are applied to individual HTML elements. They are typically placed in the `style` attribute of the HTML element.

CSS Selector

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

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

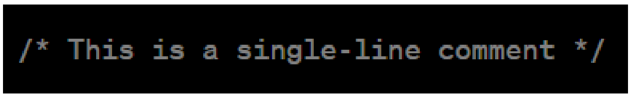
- Element Selector: Targets elements based on their tag name. For example, `p` selects all `<p>` elements.
- Class Selector: Targets elements based on their class attribute. It is denoted by a dot followed by the class name. For example, `.my-class` selects all elements with `class="my-class"`.
- ID Selector: Targets elements based on their ID attribute. It is denoted by a hash (#) followed by the ID name. For example, `#my-id` selects the element with `id="my-id"`.
- Universal Selector: Targets all elements on the page. It is denoted by an asterisk (*). For example, `*` selects all elements.
- Attribute Selector: Targets elements based on their attribute values. For example, `[type="text"]` selects all elements with `type="text"`.
- Descendant Selector: Targets elements that are descendants of another element. It is denoted by a space between two selectors. For example, `div p` selects all `<p>` elements that are descendants of `<div>` elements.
- Child Selector: Targets elements that are direct children of another element. It is denoted by a greater than sign (>). For example, `ul > li` selects all `<li>` elements that are direct children of `<ul>` elements.
- Adjacent Sibling Selector: Targets elements that are immediately preceded by another element. It is denoted by a plus sign (+). For example, `h2 + p` selects the `<p>` element that directly follows an `<h2>` element.
- Pseudo-class Selector: Targets elements based on a certain state or condition. For example, `a:hover` selects links when they are being hovered over by the mouse.
- Pseudo-element Selector: Targets a specific part of an element. For example, `p::first-line` selects the first line of a paragraph.

These are just a few examples of CSS selectors. CSS selectors provide a powerful way to select and style HTML elements, allowing you to create visually appealing and well-structured web pages.

CSS Comments

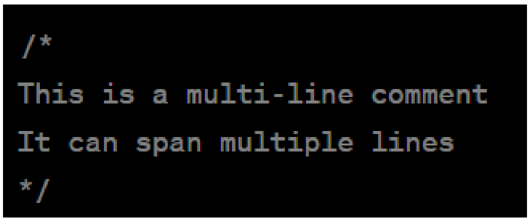
CSS comments are used to add explanatory notes or to temporarily disable portions of CSS code without affecting the rendering of the webpage. CSS comments are not displayed in the browser, as they are meant for developers and designers to provide information or annotations within the CSS code. There are two types of CSS comments:

1. Single-line comments: These comments start with `//` and continue until the end of the line. For example:



```
/* This is a single-line comment */
```

2. Multi-line comments: These comments start with `/*` and end with `*/`. They can span multiple lines. For example



```
/*
This is a multi-line comment
It can span multiple lines
*/
```

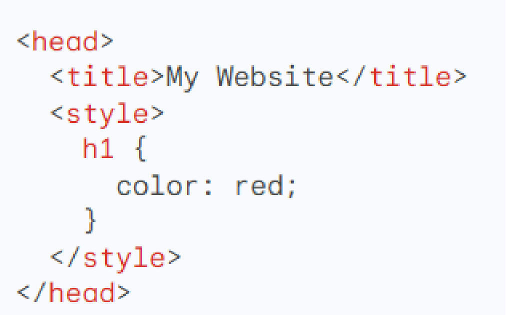
CSS comments are useful for documenting your code, providing explanations, or temporarily disabling parts of your CSS during development or debugging. They are ignored by the browser when rendering the webpage

CSS Internal Style Sheet

An internal style sheet is a style sheet that is embedded in the HTML document itself. It is typically placed in the `<head>` section of the document.

To create an internal style sheet, you need to add a `<style>` element to the `<head>` section of your HTML document. Inside the `<style>` element, you can add CSS declarations.

For example, the following code shows an internal style sheet that sets the color of all `h1` elements to red:



```
<head>
  <title>My Website</title>
  <style>
    h1 {
      color: red;
    }
  </style>
</head>
```

Once you have created an internal style sheet, it will be applied to all of the HTML elements in your document.

Here are some of the benefits of using internal style sheets:

- Easy to maintain: Internal style sheets are easy to maintain because they are stored in the same document as your HTML code. This makes it easy to find and edit your CSS declarations.
- Faster loading: Internal style sheets are loaded faster than external style sheets because they are stored in the same document as your HTML code. This can improve the performance of your website.
- Simplified structure: Internal style sheets can simplify the structure of your HTML document by removing the need to use the `style` attribute on individual HTML elements.

If you are working on a small website, internal style sheets are a good option. However, if you are working on a large website or a website that will be used by multiple people, you may want to consider using external style sheets.

Here are some of the benefits of using external style sheets:

- Scalability: External style sheets are scalable because they can be reused on multiple pages or websites. This can save you time and effort in the long run.
- Efficiency: External style sheets can improve the efficiency of your website by reducing the amount of code that needs to be loaded. This can improve the performance of your website, especially for users with slow internet connections.
- Centralized management: External style sheets can be centralized and managed in a single location. This makes it easier to keep track of your CSS declarations and to make changes to them.

If you are working on a large website or a website that will be used by multiple people, external style sheets are a good option.

CSS Inserting way

There are three ways to insert CSS into HTML:

- Internal style sheets: Internal style sheets are embedded in the HTML document itself. They are typically placed in the `<head>` section of the document.
- External style sheets: External style sheets are stored in separate files and linked to the HTML document. They are typically named with the `.css` file extension.
- Inline style sheets: Inline style sheets are applied to individual HTML elements. They are typically placed in the `style` attribute of the HTML element.

Here is an overview of each method:

- Internal style sheets:
  - Pros:
    - Easy to maintain because they are stored in the same document as your HTML code.
    - Fast loading because they are stored in the same document as your HTML code.
    - Simplified structure because it removes the need to use the `style` attribute on individual HTML elements.
  - Cons:
    - Can be difficult to manage if you have a lot of CSS declarations.
    - Not as scalable as external style sheets.
- External style sheets:
  - Pros:
    - Scalable because they can be reused on multiple pages or websites.

        - Efficient because they reduce the amount of code that needs to be loaded.
        - Centralized management because they can be stored in a single location.
    - Cons:
        - More difficult to maintain than internal style sheets because they are stored in separate files.
        - Can be slower to load than internal style sheets because they need to be retrieved from a separate file.
- Inline style sheets:
    - Pros:
        - Easy to apply styles to individual HTML elements.
        - Can be used to override styles from internal or external style sheets.
    - Cons:
        - Can be difficult to maintain if you have a lot of inline styles.
        - Not as scalable as internal or external style sheets.

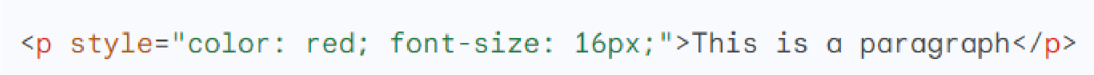
The best way to insert CSS into HTML depends on your specific needs. If you are working on a small website and you don't have a lot of CSS declarations, internal style sheets may be a good option. If you are working on a large website or a website that will be used by multiple people, external style sheets may be a better option. If you need to apply styles to individual HTML elements, inline style sheets may be a good option.

CSS Inline Styling

Inline styling is a way of applying CSS styles to individual HTML elements. It is done by adding a `style` attribute to the HTML element. The `style` attribute contains a list of CSS declarations, separated by semicolons.
For example, the following code shows how to set the color of an `h1` element to red using inline styling:

```
<h1 style="color: red;">This is a heading</h1>
```

You can also use inline styling to set multiple properties for an element. For example, the following code shows how to set the color and font size of a `p` element using inline styling:



```
<p style="color: red; font-size: 16px;">This is a paragraph</p>
```

Inline styling is a quick and easy way to apply styles to individual HTML elements. However, it is not as scalable as internal or external style sheets. If you have a lot of inline styles, it can be difficult to maintain and manage them.

Here are some of the pros and cons of inline styling:

Pros:

- Easy to apply styles to individual HTML elements.
- Can be used to override styles from internal or external style sheets.

Cons:

- Not as scalable as internal or external style sheets.
- Can be difficult to maintain and manage if you have a lot of inline styles.

If you are working on a small website or you only need to apply styles to a few individual HTML elements, inline styling may be a good option. However, if you are working on a large website or you need to apply styles to a lot of HTML elements, you may want to consider using internal or external style sheets.

CSS Color

CSS colors can be specified in a variety of ways, including:

- Named colors: Named colors are predefined colors that have names such as `red`, `blue`, and `green`.
- Hexadecimal colors: Hexadecimal colors are specified using a six-digit hexadecimal number, such as `#ff0000` for red.
- RGB colors: RGB colors are specified using three numbers, each representing the amount of red, green, and blue in the color. The numbers can range from 0 to 255. For example, `rgb(255, 0, 0)` is red, `rgb(0, 255, 0)` is green, and `rgb(0, 0, 255)` is blue.

- HSL colors: HSL colors are specified using three values, each representing the hue, saturation, and lightness of the color. The hue is a degree on the color wheel, ranging from 0 to 360. The saturation is a percentage value that represents the amount of saturation in the color. The lightness is a percentage value that represents the amount of lightness in the color. For example, `hsl(0, 100%, 50%)` is red, `hsl(120, 100%, 50%)` is green, and `hsl(240, 100%, 50%)` is blue.

You can use any of these methods to specify the color of an HTML element in CSS. For example, the following code shows how to set the color of an `h1` element to red using hexadecimal notation:

```
<h1 style="color: #ff0000;">This is a heading</h1>
```

You can also use CSS to specify the opacity of a color. The opacity of a color is represented by a value between 0 and 1, where 0 is completely transparent and 1 is completely opaque. For example, the following code shows how to set the opacity of an `h1` element to 50% using hexadecimal notation:

```
<h1 style="color: #ff0000; opacity: 0.5;">This is a heading</h1>
```

CSS Background

The CSS **`background`** property is used to set the background styles of an element. It allows you to define the background color, image, position, size, and other related properties. Here are some commonly used properties that can be used with the **`background`** property:

1. **`background-color`**: Sets the background color of an element. For example:

```
<h1 style="color: #ff0000;">This is a heading</h1>
```