

Cascading Style Sheets (CSS)

PART - 1

1

RMUA Rathnayake

University of Colombo School of Computing



Cascading Style Sheet (CSS)

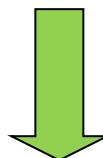
- Recommended by W3C
- The Characteristics of CSS
 - The layout of the page can be flexibly
 - It can specify the font name and size precisely
 - It can specify margin and indent
 - It can specify the position of the text and image
 - The page and web layout can be managed collectively
 - The changes can be done easily
- To validate your style sheet
 - <http://jigsaw.w3.org/css-validator/validator-uri.html>

Implementation of CSS

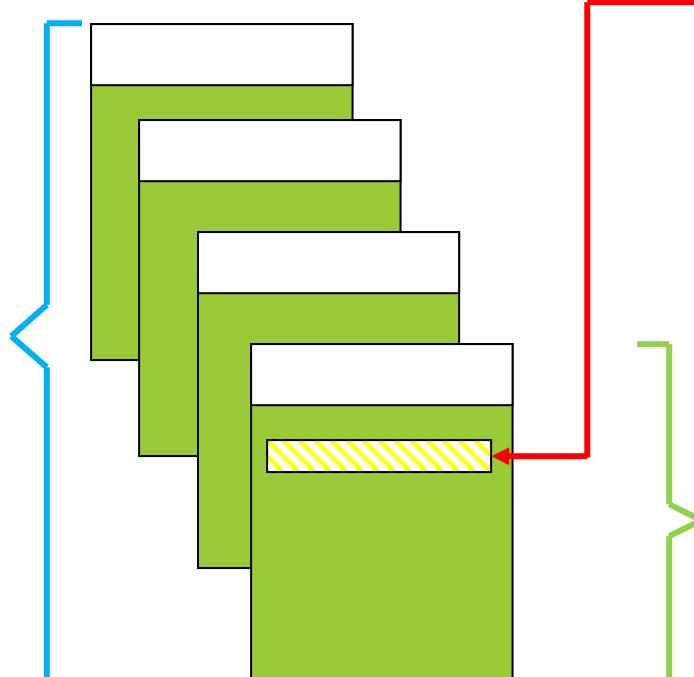
- Methods of Implementation
- Following are the 4 methods of implementing the css.
 - Inline Style sheet (Specify style directly by using the style attributes)
 - Embedded style sheet (Define style in advance to STYLE element, then apply)
 - Linking style sheet (By using LINK elements link the external file where style has been defined)
 - Import style sheet (By using STYLE element, specify the external file (define style) to be imported)

Implementation of CSS

When you want to specify the common style on all the pages



Define and apply style in external file **[Linking/Import]**

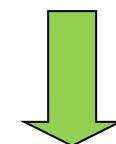


When you want to specify the common style only on this page



Define and apply style with **[Embedded style sheet]**

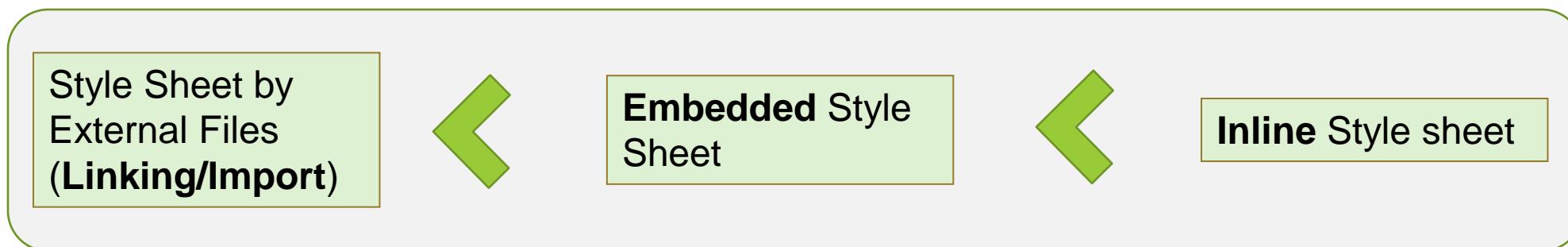
When you want to specify style only at this position



Specify style directly with **[Inline style sheet]**

Priority Level

- When multiple styles are specified in the document, the following priority order shall be applied.

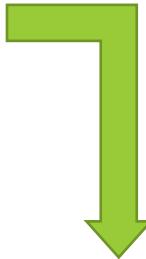


- Define the general style using external css files
- Define the style of a whole page with embedded css
- Define individual style using inline css

Inline Style Sheet

- Specify style directly using “style” attribute.

```
<BODY>  
<Tag style="property:value"> - </Tag>  
</BODY>
```



```
<BODY>  
<H1 style = "color: red" > Red heading 1 </H1>  
<P style = "color: blue; FONT-size:20px" > Blue Paragraph</P>  
</BODY>
```

Separator

- Use with each element in the BODY.
- Uses style attribute to specify the required styles
- Multiple styles can be defined, separated with semi-colon.

Embedded Style Sheet

- Define the style within the HEAD, then apply the style in the BODY.

```
<HEAD>  
  <STYLE TYPE="TEXT/CSS">
```

`selector { property : value }`

```
  </STYLE>
```

```
<HEAD>
```

Selector Tie up the HTML element and style defined by definition part

Property Specify the property toward the specified element in selector

Value Specify the applied value to the style

Embedded Style Example

```
<HEAD>

    <STYLE TYPE="TEXT/CSS">

        H1{ color : red; font-size : 20px }

    </STYLE>

</HEAD>

<BODY>

    <H1>Heading</H1> ← Applied

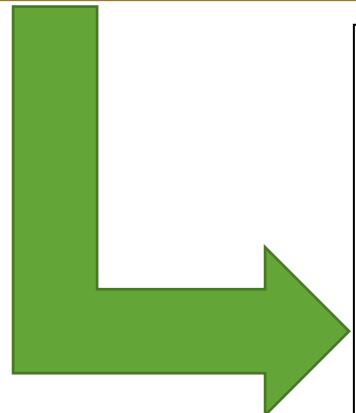
</BODY>
```

The diagram illustrates the relationship between CSS definitions and their application. A green curly brace on the right side of the slide groups the entire `<STYLE>` block under the label "Definition". A blue horizontal arrow originates from the word "Applied" and points back to the opening tag of the `<H1>` element in the `<BODY>` section.

External CSS

- Link the external file that defines the styles inside the HEAD section.
- File extension of the external styles file should be .CSS

```
<HEAD>  
    <LINK rel =“stylesheet” type =“text/css” href =“url”>  
</HEAD>
```



```
<HEAD>  
    <TITLE>title</TITLE>  
    <LINK REL=“stylesheet” TYPE=“text/css” HREF=“style.css”>  
</HEAD>  
<BODY>  
    <H1>Heading</H1>  
</BODY>
```

Linking Style Sheet ...

- REL attribute specifies the relationship with the file linked.
- TYPE attribute specifies the MIME type of style file.
- HREF attribute specifies the style file location and name.
- Both absolute and path can be used to specify the path
- Define only the [rules] in style file

Import Style Sheet

- Import the external file

```
<HEAD>  
    <STYLE TYPE="TEXT/CSS">  
        @import url (file name or URL);  
    </STYLE>  
</HEAD>
```

CSS Selectors

- There are different selectors for specifying css depending on the requirement.
 - **Element Selector**
 - Always specify common style toward the element
 - **Class Selector**
 - Create and define optional name to the specify style, then apply it.
 - **ID Selector**
 - Create and define the optional name towards the specify style, and apply it at one place in a document
 - **Group Selector**
 - Apply the common style to multiple elements
 - **Context Selector**
 - Apply style only specified part where multiple elements are all specified.

1. Element Selector

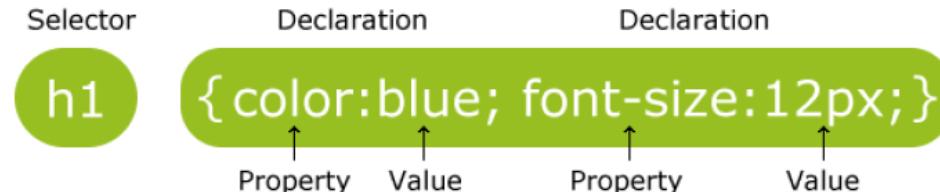
- The **common style** can be applied to an **element** at all time
- In Selector, **specify the element name** to apply the style

```
<HEAD>
  <STYLE TYPE="text/css">
    H1{color:red}
    H2{color:blue} ←
  </STYLE>
</HEAD>
<BODY>
  <H1>Heading 1</H1>
  <H2>Item 1</H2>

  <H1>Heading 2</H1>
  <H2>Item 2</H2>
</BODY>
```

```
<HEAD>
  <STYLE TYPE="text/css">
    Element {Property:Value}
  </STYLE>
</HEAD>
```

Definition / declaration



Applied

2. Class Selector

- In TYPE attribute, specify the MIME type of the style definition part.
- **Class name starts with a period (.)**
- In **Selector**, specify the name created for the define style
- When applying the CLASS, remove the period (.)

```
<HEAD>
  <STYLE TYPE="text/css">
    .red{color:#FF0000}
    .blue{color:#0000FF}
  </STYLE>
</HEAD>
<BODY>
  <H1 CLASS="red">Heading 1</H1>
  <H2 CLASS="blue">Item 1</H2>

  <H1>Heading 2</H1>
  <P CLASS="red"> Paragraph </P>
</BODY>
```

```
<HEAD>
  <STYLE TYPE="text/css">
    .class name {Property:Value}
  </STYLE>
</HEAD>
```

Define

Applied

Do not Apply
Applied

3. ID Selector

- Define and name a specific style, and apply it by specifying the name.
- However it cannot be called twice but some browsers can use it like the class selector for more than twice (using function)

```
<HEAD>
  <STYLE TYPE="text/css">
    #id123{color : red} ← Define
  </STYLE>
</HEAD>
<BODY>
  <H1 ID="id123">Heading 1</H1>
  <P>Content 1</P>
</BODY>
```

When ID is specified, the "#" will be removed

[Applied]

```
<HEAD>
  <STYLE TYPE="text/css">
    #ID {Property:Value}
  </STYLE>
</HEAD>
```

4. Grouped Selector

- By grouping multiple elements, **separated with comma (,),** common style can be applied.

```
<HEAD>
  <STYLE TYPE="text/css">
    Element 1, Element 2,... {Property:Value}
  </STYLE>
</HEAD>
```

```
<HEAD>
  <STYLE TYPE="text/css">
    H1,H2,H3 {font:24px; color:blue} ← Define
  </STYLE>
</HEAD>
<BODY>
  <H1>Heading 1</H1>
  <H2>Heading 1</H2>
  <H3>Heading 1</H3>
</BODY>
```

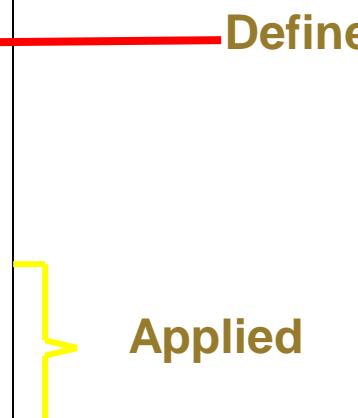
The diagram illustrates the application of a grouped selector. A red arrow labeled "Define" points from the grouped selector H1,H2,H3 in the CSS block of the HEAD section to its corresponding definition in the CSS syntax. A green curly brace labeled "Applied" groups the three H1, H2, and H3 elements in the BODY section, indicating that the shared styles defined in the HEAD are being applied to them.

5. Context Selector

- A specific style can be applied only when multiple elements are specified simultaneously.
- In selector, list the **elements separated with blank**

```
<HEAD>
  <STYLE TYPE="text/css">
    H1 I {color: red}
  </STYLE>
</HEAD>
<BODY>
  <H1>Sri Lanka
    <I> Colombo</I>
  </H1>
</BODY>
```

```
<HEAD>
  <STYLE TYPE="text/css">
    Element1 Element2 .... {Property:Value}
  </STYLE>
</HEAD>
```



CSS Combinators

A combinator is something that explains the relationship between the selectors.

A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinator. There are four different combinators in CSS3:

1. descendant selector (space)
2. child selector (>)
3. adjacent sibling selector (+)
4. general sibling selector (~)

Descendant Selector (Space)

The descendant selector matches **all elements** that are **descendants of a specified element**.

The following example selects all **<p>** elements inside **<div>** elements:

Example

```
div p {  
    background-color: yellow;  
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
div p {
    background-color: yellow;
}
</style>
</head>
<body>

<div>
<p>Paragraph 1 in the div.</p>
<p>Paragraph 2 in the div.</p>
<span><p>Paragraph 3 in the div.</p></span>
</div>

<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>

</body>
</html>
```

Descendant Selector

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

<p> elements inside <div> elements

Child Selector (>)

The child selector **selects all elements that are the immediate children of a specified element.**

The following example selects all `<p>` elements that are immediate children of a `<div>` element:

Example

```
div > p {  
    background-color: yellow;  
}
```

Child Selector

```
<!DOCTYPE html>
<html>
<head>
    <style>
        div > p {
            background-color: yellow;
        }
    </style>
</head>
<body>

    <div>
        <p>Paragraph 1 in the div.</p>
        <p>Paragraph 2 in the div.</p>
        <span><p>Paragraph 3 in the div.</p></span>
        <!-- not Child but Descendant -->
    </div>

    <p>Paragraph 4. Not in a div.</p>
    <p>Paragraph 5. Not in a div.</p>

</body>
</html>
```

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

<p> elements that are immediate
children of a <div> element:

Adjacent Sibling Selector (+)

The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element.

Sibling elements must have the same parent element, and "adjacent" means "immediately following".

The following example selects all `<p>` elements that are placed immediately after `<div>` elements:

Example

```
div + p {  
    background-color: yellow;  
}
```

Adjacent Sibling Selector

```
<!DOCTYPE html>
<html>
<head>
  <style>
    div + p {
      background-color: yellow;
    }
  </style>
</head>
<body>

  <div>
    <p>Paragraph 1 in the div.</p>
    <p>Paragraph 2 in the div.</p>
  </div>

  <p>Paragraph 3. Not in a div.</p>
  <p>Paragraph 4. Not in a div.</p>

</body>
</html>
```

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. Not in a div.

Paragraph 4. Not in a div.

all `<p>` elements that are placed
immediately after `<div>` elements

General Sibling Selector (~)

The general sibling selector selects **all elements that are siblings** of a specified element.

The following example selects all `<p>` elements that are siblings of `<div>` elements:

Example

```
div ~ p {  
    background-color: yellow;  
}
```

General Sibling Selector

```
<!DOCTYPE html>
<html>
<head>
<style>
div ~ p {
    background-color: yellow;
}
</style>
</head>
<body>
    <p>Paragraph 1.</p>
    <div>
        <code>Some code.</code>
        <p>Paragraph 2.</p>
    </div>
    <p>Paragraph 3.</p>
    <code>Some code.</code>
    <p>Paragraph 4.</p>
</body>
</html>
```

Paragraph 1.

Some code.

Paragraph 2.

Paragraph 3.

Some code.

Paragraph 4.