

UNIT 2: CSS

CSCSE11 : Bachelor of Science (1st Semester)

By

Shashvat Pratap Singh

Department of Computer Science
Institute of Science
Banaras Hindu University

UNIT II : Cascading Stylesheet

CSS rules can be applied to HTML elements in **three standard ways**.

1. Inline CSS (Using *style* attribute)
2. Internal CSS (Using <style> element inside <head> element)
3. External CSS (Separate CSS stylesheet file)

UNIT II : Cascading Stylesheet

1. Inline (Using *style* attribute)

CSS rules can be placed directly within most HTML tags by setting the core attribute **style** to the rule. For example, to set the color and alignment of an h1 heading, we might use

```
<h1 style="color: red; text-align: center;">Big Red CSS Text!</h1>
```

UNIT II : Cascading Stylesheet

2. Internal CSS (Using `<style>` element inside `<head>`)

Internal CSS is a method of adding styles within the `<style>` tag in the `<head>` section of a single HTML document. It is used when you want to apply styling to only one specific page without affecting others.

it is best suited for small or single-page projects because it cannot be reused across multiple pages and can make the file longer if many styles are added.

UNIT II : Cascading Stylesheet

```
<html lang=eu>
<head>
  <style>
    h2 { color: blue; }
    p { font-size: 16px; }
  </style>
</head>

<body>
  <h2>Hello!</h2>
  <p>This is styled using internal CSS.</p>
</body>
</html>
```

Code Example:
Internal CSS

UNIT II : Cascading Stylesheet

3. External CSS (Separate CSS file)

External CSS is a method of styling where all CSS rules are written in a **separate file** with a **.css** extension. This file is then linked to an HTML document using the **<link>** tag inside the **<head>** section. It is used when you want to apply the **same style across multiple webpages**, making your design consistent and easier to maintain. The main benefits of external CSS are **better organization**, cleaner HTML files, and the ability to reuse one stylesheet for an entire website.

UNIT II : Cascading Stylesheet

webpage.html

```
<html>
<head>
  <link rel="stylesheet" href="style.css">
</head>

<body>
  <h2>Welcome</h2>
  <p>This text is styled using external CSS.</p>
</body>
</html>
```

style.css

```
h2 { color: green; }
p  { font-size: 16px; }
```

UNIT II : Cascading Stylesheet

Common CSS Properties and Values

Text Properties

Property	Common Values
color	red, blue, #333, rgb(0,0,0)
font-size	16px, 1.2rem, 2em
font-family	Arial, "Times New Roman", sans-serif
font-weight	normal, bold, 400, 700
text-align	left, center, right, justify
text-decoration	none, underline, line-through
text-transform	uppercase, lowercase, capitalize

UNIT II : Cascading Stylesheet

Common CSS Properties and Values

Background Properties

Property	Common Values
background-color	red, #f0f0f0, rgb(200,200,200)
background-image	url("image.jpg")
background-size	cover, contain, auto
background-repeat	no-repeat, repeat, repeat-x

UNIT II : Cascading Stylesheet

Common CSS Properties and Values

Box Model Properties

Property	Common Values
width, height	100px, 50%, auto
padding	10px, 1rem
margin	20px, auto, 0
border	1px solid black
border-radius	5px, 50%

CSS SELECTORS

CSS SELECTORS

This is a Container Div

This entire section is inside a `**div**` element.

This is a Main Heading (h1)

This is a simple `**paragraph**` element. It's used for standard text content. It naturally includes some vertical space above and below it.

Click Me! (A Button Example)

This is a simple `**paragraph**` element. It's used for standard text content. It naturally includes some vertical space above and below it.

These all are elements

CSS SELECTORS

1. Element Selectors

The simplest rules can be applied to all occurrences of a particular tag, such as <p>. These selectors are called element selectors and are simply used as follows:

```
element-name { /* properties */ }
```

As an example, to set the line spacing for all paragraphs, use a rule such as the following:

```
p {line-height: 150%;}
```

To set a value for all elements, the **wildcard** selector * (asterisk) can be used. For example, to remove the margins on all elements, use

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

CSS SELECTORS

1. Element Selectors

To set a **value for more than** one but fewer than all elements, we can group elements by separating them with a comma. For example, if you want the tags `<h1>`, `<h2>`, and `<h3>` to have the same basic background and color, you could apply the following rule:

```
h1, h2, h3 {background-color: yellow; color: black;}
```

CSS SELECTORS

2. Id Selectors (#id)

By applying an id rule, a style can be applied to just a single tag. We can then reference it with a CSS selector #id-value. For example,

```
<h1 id="FirstHeading">This is the First Heading!</h1>
```

can be styled with

```
#FirstHeading {background-color: green;}
```

CSS SELECTORS

3. Class Selectors (.class)

Class values don't have to be unique because many elements can be members of the same class. In fact, elements don't even have to be of the same type to be in a common class.

Writing rules for classes is easy: simply specify the class name of your own choosing, such as “nature,” with a period before it as the selector:

```
.nature {color: green;}
```


CSS SELECTORS

3. Class Selectors (.class)

```
<html>
<head>
  <title> CSS Practice </title>
  <style>
    .nature {color: green;}
  </style>
</head>
<body>
  <p class="nature"> This is a paragraph</p>
</body>
</html>
```

OUTPUT:

This is a paragraph

CSS SELECTORS

3. Class Selectors (.class)

NOTE:

- ❑ Multiple elements may belong to same class.
- ❑ An element may belong to several classes.

Default output

This is Heading

This is a paragraph

This is also a paragraph

HTML

```
<h1 class="beta"> This is Heading</h1>
```

```
<p class="beta"> This is a paragraph</p>
```

```
<p class="alfa beta"> This is also a paragraph</p>
```

CSS

```
.beta { color: red; }
```

```
.alfa {text-decoration: underline;}
```

OUTPUT

Any Guess?

CSS SELECTORS

3. Class Selectors (.class)

NOTE:

- ❑ Multiple elements may belong to same class.
- ❑ An element may belong to several classes.

Default output

This is Heading

This is a paragraph

This is also a paragraph

HTML

```
<h1 class="beta"> This is Heading</h1>
```

```
<p class="beta"> This is a paragraph</p>
```

```
<p class="alfa beta"> This is also a paragraph</p>
```

CSS

```
.beta { color: red; }
```

```
.alfa {text-decoration: underline;}
```

OUTPUT

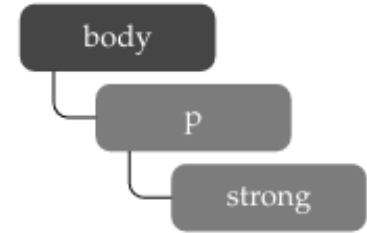
This is Heading

This is a paragraph

This is also a paragraph

CSS SELECTORS

4. Contextual Selection



Contextual selectors are created by showing the order in which the tags must be nested for the rule to be applied. The nesting order is indicated by a space between each selector. For example, given the rule

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

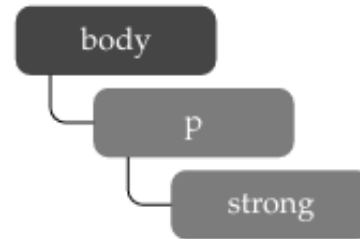
Contextual selection does not require a direct parent-child relationship with elements.

For example,

```
<p> This <span> is not <strong> directly </strong> within </span> the paragraph.</p>
```

The nested tag will still have a yellow background even though it is not directly within the tag. What you are seeing here is that the rule really says that all tags that are “descendents” of a tag are given a yellow background.

CSS SELECTORS



4. Contextual Selection

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Simple HTML Elements</title>
  <style>
    p strong {color:blue}
    h1 strong {color:red}
  </style>
</head>
<body>
  <h1> This is <strong> Heading</strong></h1>
  <p> This is <strong>a strong paragraph</strong></p>
</body>
</html>
```

OUTPUT

This is **Heading**

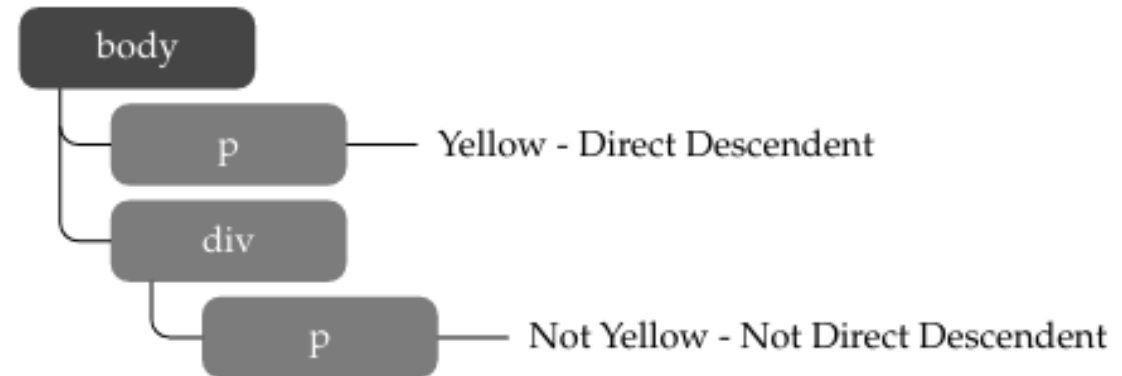
This is **a strong paragraph**

CSS SELECTORS

5. Direct Descendent Selector

CSS2 introduced the child selector specified by the greater than symbol (>) to form a rule to match only elements that are directly enclosed within another element. Consider the following rule:

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



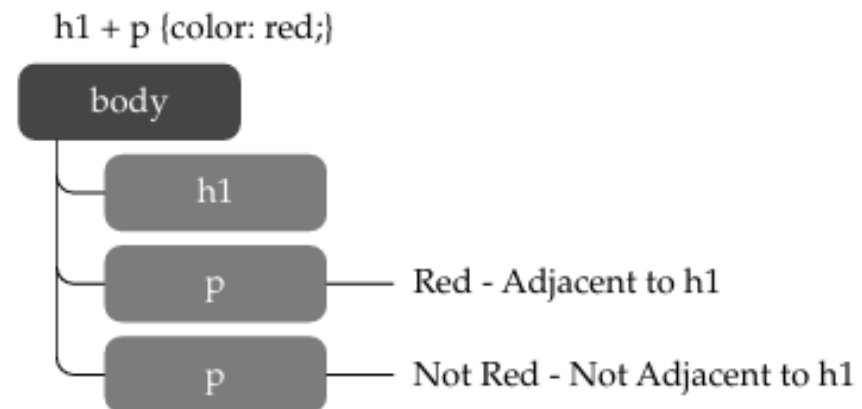
CSS SELECTORS

5. Adjacent Sibling Selector

Adjacent-sibling selector is specified using the plus sign (+) and is used to select elements that would be siblings of each other. For example, consider the following rule: `h1 + p {color: red;}`

This states that all paragraph elements that are directly after an `<h1>` are red, as indicated by this markup:

```
<h1>I am a heading</h1>  
<p>I am an adjacent paragraph so I am red!</p>  
<p>I am not adjacent so I am not red.</p>
```

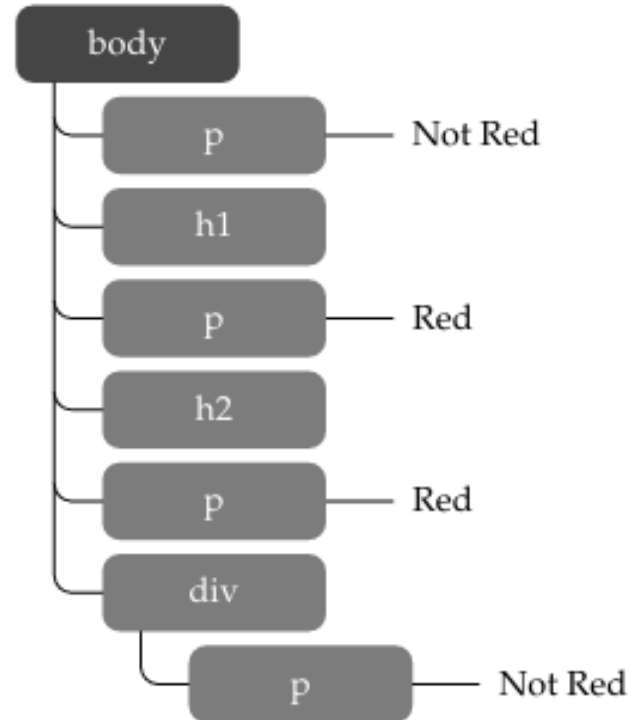


CSS SELECTORS

```
<p>I am not red.</p>  
<h1>Heading 1</h1>  
<p>This is red.</p>  
<h2>Heading 2</h2>  
<p>I am red too.</p>  
<div><p>Not me as I am not a sibling given that I am one level down.</p></div>
```

6. General Sibling Selector

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



The background is a full-page abstract image featuring swirling, ethereal patterns of light blue, teal, and pale orange against a dark, muted green backdrop. The colors blend and flow together in a fluid, organic manner, creating a sense of movement and depth. The overall effect is reminiscent of ink or paint being dropped into water and then gently stirred.

CSS COLORS

CSS COLORS

Format	Syntax	Description
Named colors	red, blue	140 standard color names.
Hex	#RRGGBB	Hexadecimal code (00-FF).
RGB	rgb(r, g, b)	Red, Green, Blue values (0-255).
RGBA	rgba(r, g, b, a)	RGB + Alpha (Opacity 0.0 to 1.0).
HSL	hsl(h, s, l)	Hue (0-360), Saturation (%), Lightness (%).

CSS COLORS

Named color format

Named colors are predefined English keywords (like red, blue, or white) that browsers recognize and render as specific RGB values.

Color Name	Description
black	Absolute black.
white	Absolute white.
red	Pure, bright red.
blue	Pure, bright blue.
green	A standard, medium dark green.
yellow	Bright, primary yellow.
cyan	Bright blue-green (Aqua).
magenta	Bright pink-purple (Fuchsia).

CSS COLORS

hexadecimal color format

CSS's six-digit hexadecimal format specifies a color as `#rrggbb`, where `rr` is the amount of red, `gg` the amount of green, and `bb` the amount of blue, all specified as a hexadecimal value ranging from `00` to `FF`.

Color Name	Hex Equivalent
Black	#000000
Silver	#C0C0C0
Gray	#808080
White	#FFFFFF
Maroon	#800000
Red	#FF0000
Purple	#800080
Fuchsia	#FF00FF
Green	#008000
Lime	#00FF00
Olive	#808000
Yellow	#FFFF00
Navy	#000080
Blue	#0000FF
Teal	#008080
Aqua	#00FFFF

CSS COLORS

rgb color format

CSS colors can also be defined using the keyword `rgb`, followed by three numbers between 0 and 255, contained in parentheses and separated by commas, with no spaces between them. RGB color values can also be defined using percentages. The format is the same, except that the numbers are replaced by percentage values between 0% and 100%.

```
#p1 {color: rgb(204,0,51);}
```

rgb color format

Color Name	RGB Value	Description
black	rgb(0, 0, 0)	No light (Pure Black).
white	rgb(255, 255, 255)	Max light (Pure White).
red	rgb(255, 0, 0)	Pure Red.
lime	rgb(0, 255, 0)	Pure Green (Bright).
blue	rgb(0, 0, 255)	Pure Blue.
yellow	rgb(255, 255, 0)	Red + Green mixed.
cyan	rgb(0, 255, 255)	Green + Blue mixed (Aqua).
magenta	rgb(255, 0, 255)	Red + Blue mixed (Fuchsia).
silver	rgb(192, 192, 192)	Light Gray.
gray	rgb(128, 128, 128)	Medium Gray.

CSS COLORS

rgba color format

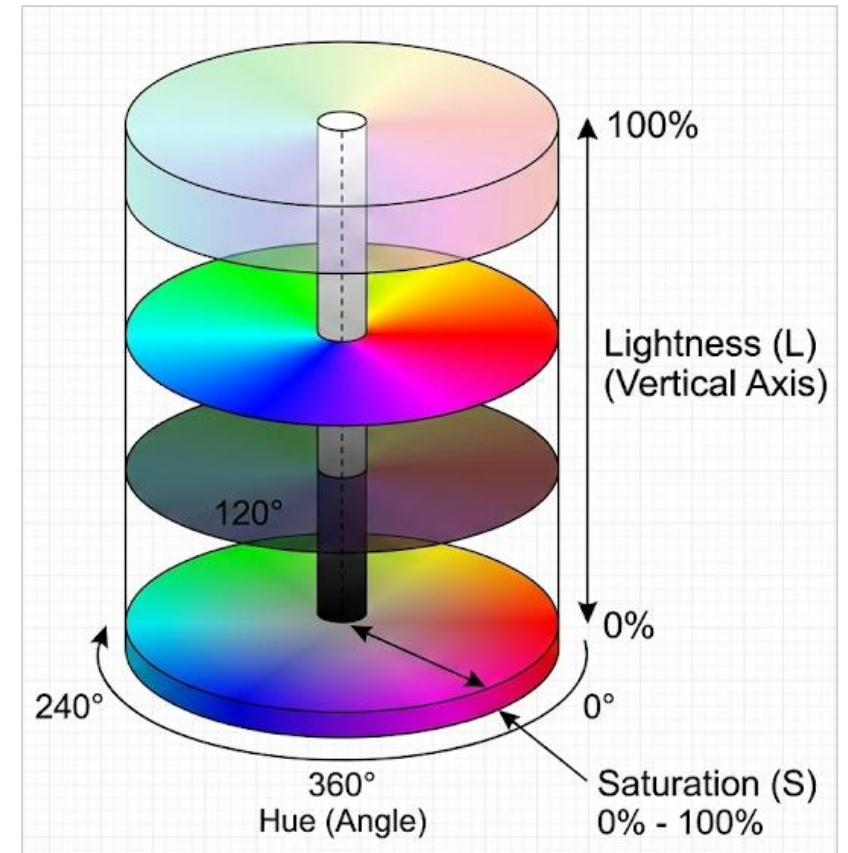
Like RGB color, but this adds an alpha channel value to specify the opacity of the color. An RGBa is specified via a function style `rgba(r,g,b,a)` value, where colors `r`, `g`, and `b` are specified as decimal values from 0 to 255 or a percentage from 0 to 100% and the alpha channel value for defining opacity is a number between 0 (fully transparent) and 1 (fully opaque). Values outside this range will be rounded up or down to fit the closest value.

```
#redtrans {color:rgba (255,0,0,0.4);}
```


CSS COLORS

hsl color format

Color values are specified as **hsl(hue, saturation, lightness)**. Hue is set as the degree on the color wheel, where if you wrap around 0 or 360 is red, 120 is green, and 240 is blue, with the various other colors found between. Saturation is a percentage value, with 100% the fully saturated color. Lightness is a percentage, with 0% being dark and 100% light, with the average 50% being the norm.



CSS BACKGROUND

CSS BACKGROUND

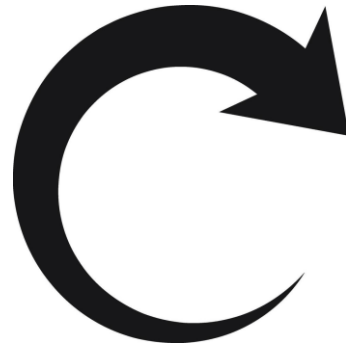
Property	Description	Common Values
background-color	Sets the background color.	Any color value.
background-image	Sets an image as the background.	url('path/to/image.jpg')
background-repeat	Defines if/how the image repeats.	no-repeat, repeat-x, repeat-y
background-size	Specifies the size of the background image.	cover (fills area), contain (shows whole image), px %

```
body {  
  background-image: url('pattern.png');  
  background-repeat: no-repeat;  
  background-size: cover;  
  background-position: center;  
}
```

Margin, Padding & Border

Component	Definition	Properties	Common Values
Padding	Space inside the border, around the content. Clears an area around the content.	padding, padding-top, padding-left...	10px, 2em, 5%
Border	A line that goes around the padding and content.	border-width, border-style, border-color	1px solid black, dotted red
Margin	Space outside the border. Separates the element from neighbors.	margin, margin-top, margin-auto...	20px, auto (centers horizontally)

Margin, Padding Property-value
Rotation effect



CSS LAYOUT

CSS LAYOUT

CSS Layout refers to the techniques used to arrange HTML elements on a page. By default, HTML elements flow in a single column (top to bottom). The display property allows you to change this behavior, letting you create rows, grids, and complex structures.

The **display** property specifies the inner and outer display type of an element. It determines if an element stands alone (like a headline) or sits next to other elements (like a word in a sentence).

```
.selector { display: value; }
```

CSS LAYOUT

Common *Display* Values

Value	Description	Default Elements	Behavior
block	The element starts on a new line and takes up the full width available.	<div>, <p>, <h1>, <header>, <footer>	Stacks vertically (like bricks). Respects width/height.
inline	The element stays on the same line and takes only as much width as necessary.	, <a>, , 	Flows horizontally (like text). Does NOT respect width/height/top-margin.
inline-block	A hybrid. It sits on the same line (like text) but maintains box properties (width/height).	<button>, <input>	Useful for navigation bars or grids of cards.
none	The element is completely removed from the document layout.	None (applied manually)	It acts as if the element does not exist (different from visibility: hidden).

CSS LAYOUT

Modern Layout Values

These values unlock powerful layout systems for complex designs (often called "CSS Modules").

Value	Description	Use Case
flex	Enables Flexbox . Lays out child elements in a single row or column with automatic spacing.	1D layouts (navbars, centering items).
grid	Enables CSS Grid . Lays out child elements in rows AND columns simultaneously.	2D layouts (entire page structures, photo galleries).

CSS Layouts

To be Continued...