

## - cascading Style Sheets

- It describes how HTML elements are to be displayed on screen, paper / other media.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.

## → Why use CSS?

- It is used to define styles for web pages, including the design, layout & variations in display for different devices & screen size.

## → Applications :-

### 1) CSS Save time :-

You can write CSS once & then reuse same sheet in multiple HTML pages.

You can define style for each HTML ele. and apply it to as many web pages as you want.

### 2) page load faster :-

Do not need to write HTML tag attrib every time.

- Just write one CSS rule of a tag & apply it to all the occurrences of that tag.
- So less code means faster download times.

### 3) Easy maintenance :-

To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

### 4) Multiple Device Compatibility:-

Style sheets allow content to be optimized for more than one type of device.

By using the same HTML doc, different version of a website can be presented for comps, mobile, printing.

## \* Syntax :-

- A CSS rule set contains a selector & a declaration block

Selector { Name1: Value1; Name2: Value2; }

where,

Declaration block

### ① Selector:

Indicates the HTML element you want to style. Any tag.

## ② Declaration Block:-

- It can contain 1/more declarations separated by a semicolon.
- Each declaration contains a property name & value, separated by a colon.

Ex:- `h1 { color: yellow; font-size: 11px; }`

## ③ Property:-

It is a type of attributes of HTML element. It could be color, border etc.

## ④ Value:-

Values are assign to CSS properties.

Selector { prop1: val1; Prop2: val2; ...; }



## CSS Selector:-

CSS selectors are used to select the content you want to style.

CSS selector select HTML elements according to its id, class, type, attributes etc.

## → Different types of selectors

- Element

- Id

- class

- Universal

- Group

## 1) CSS Element Selector

The element selector selects the HTML element by name.

```
<head>
<style>
p {
    text-align: center;
    color: blue;
}
```

## 2) CSS Id Selector:-

The id selector selects the id attribute of an HTML elements to select a specific element.

- Id is always unique within the page so it is chosen to select a single, unique elem

- It is written with hash character (#), followed by the id of the elem.

```
#Para1 {
    text-align: center;
    color: blue;
}
```

```
<body>
    <p id="Para1">....</p>
</body>
```

## 8) CSS class selector :-

- It selects with a specific class attribute.
- It is used with a period character.
- class name should not be started with a no.

```
.center {  
    text-align: center;  
    color: blue;  
}
```

```
<body>  
    <h1 class = "center">...</h1>  
</body>
```

## 4) CSS class selector for specific element.

If you want to specify that only one specific HTML ele. should be affected then use the ele. name with class selector.

```
p.center {  
    text-align: ce.  
    color: blue;  
}
```

```
<body>  
    <h1 class = "center">...</h1>  
    <p class = "center">...</p>
```

## 5) Universal selector:

Universal selector is used as a wildcard character.

- It selects all the elements on the page.

\* {

```
color: green;
font-size: 20px;
```

## 6) Group Selector:

It is used to select all the elements with the same style definitions.

- Grouping selector is used to minimize the code.
- Commas are used to separate each selector in grouping.

h1 {

```
text-align: center;
color: blue;
```

}

h1, h2, p {

```
text-align: center;
color: blue;
```

}

h2 {

```
text-align: center;
color: blue;
```

}

p {

```
text-align: center;
color: blue;
```

}

## \* How to add css :-

- CSS is added to HTML pages to format the document according to information in the style sheet.
- There are three ways to insert CSS in HTML documents.
  - Inline CSS
  - Internal CSS
  - External CSS.

### 1) Inline CSS:-

It is used to apply CSS on a single line or element.

```
<p style="color:blue">Hello</p>
```

```
<h2 style="color:red; margin-left:40px;">...</h2>
```

### → Disadvantage :

- You can't use quotations within inline CSS. If you use quotations, the browser will interpret this as an end of style value.
- The style can not be reused anywhere else.
- These styles are tough to be edited bcs they are not stored at a single place.

## 2) Internal css:

It is used to apply css on a single document / page.

- It can affect all the elements of the page
- It is written inside the style tag within head section of html.

```
<style>
  p {color: blue}
</style>
```

- It is used to add a unique style for a single document.
- It is defined in <head> section of Html page.

## 3) External css:

It is used to apply css on multiple pages or all pages.

- Write all the css code in a css file.
- Its extension must be .css

```
<link style="stylesheet" type="text/css"
      href='xyz.css'>
```

- The link tag must be used inside the head section.
- It is ideal for change on multiple pages bcs it facilitates to change the look of entire web page/site by changing one file

- external stylesheet may be written in any text editor but must be saved with .css.
- This file should not contain HTML elements.
- Not use space b/w property value & unit.

margin-left: 20px ✓

margin-left: 20 Px ✗

body {

background-color: red;

#### \* CSS Comments:-

- Comments are generally written to explain your code.
- It is very helpful for the users who reads your code so that they can easily understand the code.
- Comments are ignored by browsers.
- comments are single / multiple lines statements & written within it ... /\*

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

PS

color: red;

→ ~~color: red; /\* text color red \*/~~

P {

color: red; /\* text color red \*/

→ Multiline comments are enclosed in /\* \*/.

/\* Hello

This is multiline

comments \*/

P {

color: red; /\* text color red \*/

}

\*

CSS colors: red, green, blue, black, white

colors are specified using predefined  
color names or RGB, HEX, HSL, RGBA, HSLA  
values.

- CSS/HTML support 140 standard color names.

→ Text color:-

<h1 style = "color: red;"> Hello </h1>

→ Border color:-

<h1 style = "border: 2px solid red;"> Hello </h1>

## → CSS color Values:-

`<h1 style="background-color:rgb(255,99,71);>...</h1>`

`<h1 style="background-color:#ff6347;">...</h1>`

`<h1 style="background-color:hsl(9,100%,65%)">...</h1>`

## \* CSS Background :-

- CSS background Property is used to define the background effects on element.
- There are 5 CSS background properties that affects the HTML elements:

→ background-color

→ " " - image

→ " " - repeat

→ " " - attachment

→ " " - position.

### 1) Background - color:-

h1 {

background-color:#b0d4de;}

## 2) Background - image :-

- Used to set an image as a background of an element
- By default the image covers the entire element
- Set image for a page as following:

<Style>

```
body {  
    background-image: url('xyz.gif');  
}
```

</Style>

Note: The background-image property repeats the background image should be chosen according to text color.

- The bad combination of text & background image may be a cause of poor designed and not readable webpage.

## 3) Background - repeat:-

- By default, the background-image property repeats the background image horizontally & vertically.

- The background looks better if the image repeated horizontally only.

body {

background-image: url ("xyz.png");

background-repeat: repeat-x; / repeat-y;

}

#### 4) Background-attachment:

- This property is used to specify if the background image is fixed (scroll with the rest of the page in browser window).

- If you set fixed, the background image then the image will not move during scrolling in browser

background: white url ('bb.gif');

background-repeat: no-repeat;

background-attachment: fixed;

#### 5) Background - position :-

This property is used to define the initial position of the background image.

- By-default, the background image is placed on the top-left of the webpage.

14

- you can set following position.

- 1) center
- 2) top
- 3) bottom

4) left

5) right

exm:

background-position: center;

## \* CSS Borders:-

- used to set borders on an element.
- It is used to specify the style, color & size of the border of an element.
- Border properties are :-
  - \* border-style
  - \* border-color
  - \* border-width
  - \* border-radius

### 1) Border style:

It is used to specify the border type.

- Following values are used :

dotted	ridge
dashed	inset
Solid	offset
double	none
groove	hidden

ex: `<style>`

```
  p{border-style: dotted;}
```

`border-style: dotted;`

## 2) Border width:-

- This property specifies the width of the four borders.
- The width can be set as a specific size (px, pt, cm, em, etc) or.

Pre-defined values :

thin, medium, thick

ex:-

P: two {

border-width : 2px / medium / thick

→ Specific Side width:-

- It has one to four values.

Top, Right, Bottom, Left.

① border-width : 5px, 20px;

↑ ↑

T & B R & L

② border-width : 25px 10px 4px 30px

↑ ↑ ↑ ↑

T R B L

## 3) Border color :-

- Color can be set by

- 1) name : "red"
- 2) HEX : "#ff0000"
- 3) RGB : "rgb(255,0,0)"
- 4) HSL : "hsl(0,100%,50%)"
- 5) transparent

- If border-color is not set, it inherits the color of the element.

ex border-color : red green blue pink;  
 T R B L

## 4) Border sides:

1) 4 values

T R B L

2) 3 values

T RRL B

3) 2 values

T&amp;B R&amp;L

4) 1 value

All Four T&amp;R&amp;B&amp;L

ex

border-style : dotted solid double;

## 5) Border - Shorthand:

border-width

border-style (Required)

border-color

ex:-

border: 5px solid Red;

border-left: 6px solid Red;

## 6) Rounded Borders:

border-radius: 5px;

or

border-bottom-color: red;

border-left-style: dotted;

border-right-width: thin;

initial / thick / length / initial

border-radius

border-radius: 5px; border-radius: 10px;

border-radius: 50%;

border-radius: 50% / 50%;

border-radius: 50% / 50% / 50% / 50%;

border-radius: 50% / 50% / 50% / 50% / 50%;

border-radius: 50% / 50% / 50% / 50% / 50% / 50%;

## \* CSS Margin:-

- Used to define the space around elements
- It is completely transparent & doesn't have any background color.
- It clears an area around the element.

### 1) Margin Properties:

- margin → margin-top
- margin-left → margin-bottom
- margin-right

### 2) Margin Values:

- auto
- length → %
- ↑ → inherit

Px, pt, cm

## \* CSS Padding

It is used to define the space between the element content & the element border.

- Padding is affected by the background color.  
It clears an area around the content.

### 1) Padding properties:-

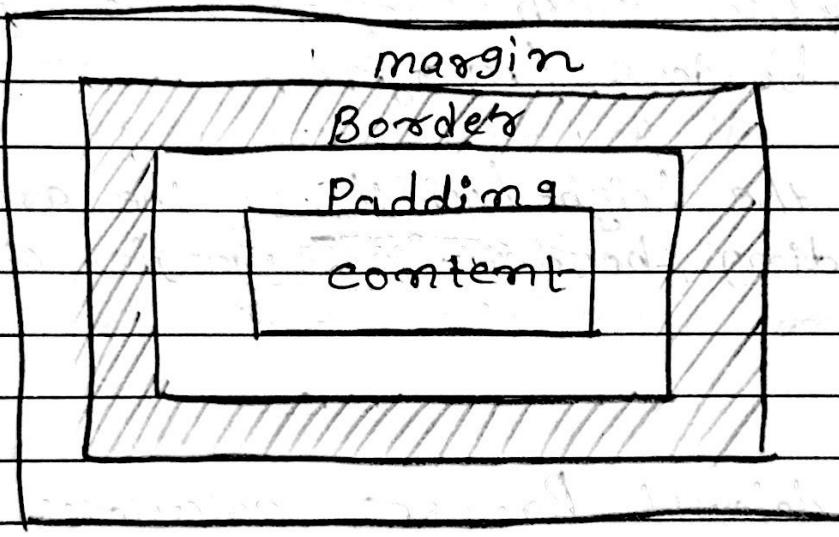
- Padding
- padding-left
- padding-right
- padding-top
- padding-bottom

### 2) Padding Values:

- length: px, pt, cm
- %
- inherit

## \* CSS Box Model:

- All HTML elements can be considered as boxes.
- The CSS box model is essentially a box that wraps around every HTML element.
- It consists of margins, borders, padding and the actual content.



**Content** :- where text and image appear.

**Padding** :- clears an area around the content.  
The padding is transparent.

**Border** :- The border that goes around the padding & content

**Margin** :- clears an area outside the border.  
The margin is transparent.

Ques

```
div {
```

```
    border: 10px solid red;
```

```
    padding: 50px;
```

```
    margin: 20px;
```

```
}
```

### \* CSS Height / width :-

- The height & width properties do not include padding, borders / margins.
- It sets the height / width of the area inside the padding, border and margin of the ele.

#### → Values:

auto: default. Browser calculates H & w.

length: H & w in px, cm.

initial: sets the H/w to its default value.

inherit: H/w inherited from its Parent.

#### → Get max-width:-

- max-width can be specified in length values like px, cm etc. or in (%) of the containing block, or set to none.

- Problem occurs when the browser window is smaller than the width of the ele.
  - The browser then adds a horizontal scrollbar to the page.
  - Using max-width instead, will improve the browser's handling of small windows.
- \* Note :-  
when you set the width and height properties of an element with CSS, just set the width and height of the content area.

- To calculate the full size of an element, must also add padding, borders and margins.

Example :-

```
div {  
    width: 320px;  
    padding: 10px;  
    border: 5px solid gray;  
    margin: 0;  
}
```

$320\text{px (width)} + 20\text{px (left + right padding)} + 10\text{px (left + right border)} + 0\text{px (left + right margin)} = 350\text{ px}$

Total element width = width +  
left padding + right padding +  
left border + right border +  
left margin + right margin.

Total element height = height +  
top padding + bottom padding,  
top border + bottom border +  
top margin + bottom margin.

### \* CSS Text :-

#### 1) Text color:-

color name

HEX value

RGB value.

#### 2) Text Alignment:-

text-align: justify;

Used to set the horizontal alignment of a text.

- The text can be L / R aligned, centered / justified.

- When the text-align property is set to "justify" each line is stretched so that line has equal width and the left & right margins are straight. (like newspaper)

### 3) Text Decoration:-

text-decoration: overline;

- This Property is used to set / remove decorations from text.

- The value text-decoration: none; is used to remove underline from links

### 4) Text Transformation:-

text-transform: uppercase;

- It is used to specify uppercase & lowercase letters in a text.

### 5) Text Indentation:-

Used to specify the indentation of the first line of a text. text-indent: 50px;

### 6) Letter Spacing:-

letter-spacing property is used to specify the space between the characters in a text.

letter-spacing: 3px;

### 7) Line height:-

line-height Property is used to specify the space between lines. line-height: 0.8;

### 8) Text Direction:-

direction and unicode-bidi properties can be used to change the text direction of an ele. direction: rtl;

unicode-bidi: bidi-override;

### 9) Word Spacing:-

word-spacing property is used to specify the space between the words in a text.  
word-spacing: 10px;

### 10) Text shadow:-

Adds shadow to text.

Ex:

text-shadow: 3px 3px red;

↑

↑

↑

horizontal  
Shadow  
Position.

Pos<sup>n</sup>  
of ver.  
Shadow

color of  
shadow.



### CSS Fonts:-

Font properties defines the font family, boldness, size & the style of a text.

#### 1) Font Family:-

- The font-family property should hold several font names as a "fallback" system.
- If the browser does not support the 1<sup>st</sup> font, it tries the next font & so on.
- Start with the font you want & end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

If the name of the font family is more than 1 word, it must be in quotation marks.

ex:- font-family: "Times New Roman", Times, serif;

### e) Font style:-

Mostly used to specify italic text.

#### - Values:-

- ① normal
- ② italic
- ③ oblique

ex:- font-style: normal;

### 3) Font size:-

Font-size value can be an absolute / relative size.

#### ① Absolute size:-

- Sets the text to a specified size.
- Does not allow a user to change the text size in all browsers.
- Absolute size is useful when the physical size of the screen is known.

### ① Relative Size:-

- Sets the size relative to surrounding el.
- Allows a user to change the text size in browsers.
- If do not specify font-size, the default size for normal-text is 16px ( $16\text{px} = 1\text{em}$ )

ex:-

font-size: 100% / 2.5em;

### 4) Font weight:-

font-weight: normal;

### 5) Responsive Font size:-

- The text size can be set with vnu unit
- So text size will follow the / (viewport width) size of browser window

ex:-

font-size: 10vnu;

### 6) Font Variant:-

- font-variant property specifies whether / not a font should be displayed in a small-caps font
- In a small-caps font, all lowercase letters

are converted to uppercase.

- The converted uc letters appears in a smaller font size than the original uc letters in the text.

Font-variant: normal / small-aps;

## \* CSS Links:-

- Four links states are:-

a:link → a normal, unvisited link.

a:visited → a link the user has visited

a:hover → a link when the user mouses over it

a:active → a link the moment it is clicked.

- a:hover must come after a:link and a:visited
- a:active must come after a:hover

## ① styling Links :-

- Links can be styled with any css property colors

font-family  
background

ex:-

```
a {
```

```
    color: hotpink;
```

## 2) Text Decoration:-

Used to remove underlines from links.

```
a:link {
```

```
    text-decoration: none / underline;
```

## 3) Background Color:-

Used to specify a background color for links

```
a:visited {
```

```
    background-color: yellow;
```

## 4) Link Button:-

link of button type.

## \* CSS Lists :-

- CSS list properties allows you to:

- ① Set different list item markers for ordered lists.
- ② Set diff. list item markers for unordered list.
- ③ Set an image as the list item marker.
- ④ Add background color to lists & list items.

### 1) List Item Markers:-

ul {

list-style-type: circle; /square; /upper-alpha;  
lower-alpha;

### 2) Image as the list item marker:

ul {

list-style-image: url('xyz.gif');

### 3) Position list item markers: (bullet point)

Values:-

#### ① outside:

Bullet points will be outside the list item.

The start of a list item will be align vertically  
each line by default.

- Banana
- Apple
- :

## ② Inside :

- Bullet Points will be inside the list item.  
As it is the part of list item, it will be part of the text.

- Banana
- Apple

## ④ Remove Default Settings:

- Used to remove the markers / bullets.  
- List also has default margin and padding.  
- To remove this, add margin: 0 & padding: 0

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
}
```

## ⑤ List - Shorthand Property:

```
list-style: square inside url ("xyz.gif")  
          ↑      ↑      ↑  
        type   position  image
```

- If any one property missing, default value is inserted.

## 6) Lists with colors

- Anything added to the `<ol>/<ul>`, affects the entire list.
- Properties added to the `<li>` will affect the individual list items.

ol {  
background: red;  
}

ol li {  
background: yellow;  
}

X

Information

## \* CSS Tables:

### 1) Table Borders:

table {

border: 1px solid red;

}

### 2) Collapse Table Borders:-

Sets whether the table borders should be collapsed into a single border:

table {

border-collapse: collapse;

border: 1px...

{

### 3) Table width & height:

table {

width: 100%;

height: 50px;

{

### 4) Horizontal Alignment:-

- By-default, the content of <th> elements are center-aligned & <td> are left-align.

th {

text-align: left; }

## 5) Vertical Alignment:

- By default, vert alignment of content in a table is middle (`<th>` & `<td>` both)

`td {`

`vertical-align: bottom;`

## 6) Table Padding:

- To control the space between the border & the content in a table, use the padding property on `<td>` & `<th>` ele.

`th, td {`

`padding: 15px;`

`}`

## 7) Horizontal Dividers:

`th, td {`

`border-bottom: 1px solid #ddd;`

`}`

## 8) Hoverable Table:

Use `:hover` selector on `<tr>` to highlight table rows on mouse over:

tr:hover {

background-color: red;

#### 9) Striped table:-

- For zebra-striped tables, use the  $n^{\text{th}}$ -child() Selector and
- Add a background-color to all even/odd table rows:

tr: nth-child(even) {

background-color: red; }

#### 10) Table color:-

th {

background-color: red; }

#### 11) Responsive Table:-

It will display a horizontal scroll bar if the screen is too small to display full content.

Add a container element (like `<div>`) with `overflow-x: auto` around the `<table>` ele. to make it responsive.

```
<div style="overflow-x:auto;">  
    <table>  
        </table>  
</div>
```

## \* CSS Display:-

The display property is the most important CSS property for controlling layout.

- It specifies if / how an element is displayed.
- Every HTML element has a default display value depending on what type of element it is.
- The default display value for most element is block / inline.

### 1) Display : none

It is used with JS to hide & show elem. without deleting & recreating them.

- The <script> uses display:none; as default.

## a) Override the Default Display Value:-

- changing an inline element to a block element or vice versa, can be useful for making the page look a specific way, and still follow the web Standards.

```
li {  
    display: inline; }  
span {  
    display: block; }
```

## b) Visibility : hidden :-

This property also hides an element.

- However, the element will still take up the same space as before.
- The element will be hidden, but still affect the layout.

```
h1 {  
    visibility: hidden; }
```

```
h1 {  
    visibility: hidden; / display: none; }
```

## \* CSS Position:-

- Specify the type of positioning method used for an element.

→ Values:-

- Static
- Fixed
- Sticky
- relative
- absolute

- Elements are then positioned using the top, bottom, left & right properties.

- These properties will not work unless the **position** property is set first.

- They also work differently depending on the position value.

### ① Static:-

- HTML elements are positioned static by default.

- Static Positioned elements are not affected by the top, bottom, left & right properties.

- An element with position: static; is not positioned in any special way;

- it is always positioned according to the normal flow of the page.

div

{ position: static;

## 2) Relative:-

An element with position: relative; is positioned to its normal position.

- Setting the top, right, bottom & left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.
- Other content will not be adjusted to fit into any gap left by the element.

div {

position: relative;

}

## 3) Fixed:-

An element with position: fixed; is positioned relative to the viewport, which means it always stay in the same place even if the page is scrolled.

- The top, right, bottom & left properties are used to position the element.
- A fixed element does not leave a gap in the page where it would normally have been.

located.

```
div {  
    position: fixed;  
}
```

#### 4) Absolute:

An element with `position: absolute;` is positioned relative to the nearest positioned ancestor.

- If an absolute positioned ele. has no positioned ancestors, it uses the document body, & moves along with page scrolling.

```
div {  
    position: absolute;  
}
```

#### 5) Sticky :-

It is positioned based on the user's scroll position.

- The sticky element toggles between relative & fixed, depending on the scroll position
- It is positioned relative until a given offset position is met in the viewport- then it "sticks" in place.

div {

position: sticky;

}

## 6) Overlapping Element:

- When elements are positioned, they can overlap other elements.
- The z-index property specifies the stack order of an element by defining which one is on top.
- An element can have a +ve / -ve stack order.

img {

z-index: -1; position: absolute;

}

## \* CSS Layout:

- The CSS "float" property is a positioning property.
- It is used to push an element to the left/right; allowing other elements to wrap around it.
- It is generally used with images & layouts.

### → How it works:

- Elements are floated only horizontally. So it is possible only to float elements L/R, not ↑/↓.
- 1) A floated element may be moved as far to the left/Right as possible.
  - It means, a floated element can display at extreme L/R.
- 2) The elements after the floating ele. will place around it.
- 3) The elements before the floating ele. will not be affected.
- 4) If the image floated to the right, the texts flow around it, to the L & if the image floated to the left, the text places around it, to the right.

## → CSS Float Properties:

### ① clear:

Used to avoid elements after the floating elements which flow around it.

Values:- Left, Right, both, none, inherit.

### ② float:

Specifies whether the box should float/not.

Values: left, Right, none, inherit.

## → CSS Float Property Values.

As above, we can use left, right, none.

img {

float: none / right / left;

img {

clear: left;

left