

CSS Cascading Style Sheet

CLASSMATE

Date _____

Page _____

It is used to style an HTML webpage

→ there are 3 ways to apply style to HTML

i) Inline Styling :-

With the help of style attribute which is placed in the opening tag we can provide CSS.

- It will be applicable to particular element to which we are writing CSS.

Ex:-

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

Attribute → property value

ii) Internal Styling :-

- Here styling will be defined inside the particular HTML page using `<Style>` tag placed inside the head tag.
- This styling will be applied to the particular HTML page where we are writing CSS.

```
<head>
```

```
  <Style>
```

```
    " " " " "
```

```
  <Style>
```

```
  </head>
```

```
  <body>
```

```
  <body>
```

iii) External Styling

- Here styling is written in the separate file having .css extension connected with html
- we can connect one Style Sheet to the multiple html page.

→ Inline Styling

It is having more priority than internal and external.

External and internal priority depends on how we are connecting link tag either below the style tag or above the style tag, when it is at the end having more priority.

* Types of Selectors

a) Simple Selectors

→ Element Selector
Syntax: tagname {

--- Rule ---

y

b) Class Selector

- class name of element
y --- Rule ---

① id Selector

Syntax:

#idname {

--- style ---

}

② attribute Selector

Tags name

• class name

#id name

Syntax:

h1 [id = "...."] {

--- style ---

Priority :-

1) id

2) Attribute

3) class

4) element

③ grouping Selector

h1, h2, p, button {

--- style ---

④ universal Selector

* d

--- style ---

2)

Combinator Selectora) Descendent Selector

→ It is typically represented by a single space character combine 2 selector such that element matched by the second selector if they have an ancestor element matched the first selector.

Ex:-

<body>

<div>

<p> - 4<p>

<button> </button>

</div>

<button></button>

<p> </p>

body div button {

--- style ---

(we can use class or id instead of selector)

b) Child Selector

(immediate)

This will select the elements which are direct children of its parent (>)

Ex:-

body > button {

--- style ---

body > div > button {

--- style ---

c) General Sibling Selector

In this we can select the siblings by using (+) & (~) if all they all are next to each other it will be else to all

ex:-

one

Two

Three

ul > li &

... Sty -

li + li &

- " Sty -

li li &

* Pseudo Selector

i) Pseudo class (:)

It is a key word added to the Selectors that specifies the special state of the selected elements.

ex: i) hover

ii) Active iii) Disabled iv) checked

v) Readonly

vi) placeholder vii) valid

viii) invalid

ix) Required x) focus

xii) nth-child()

xiii) first-child

etc

ii) Pseudo elements ():-)

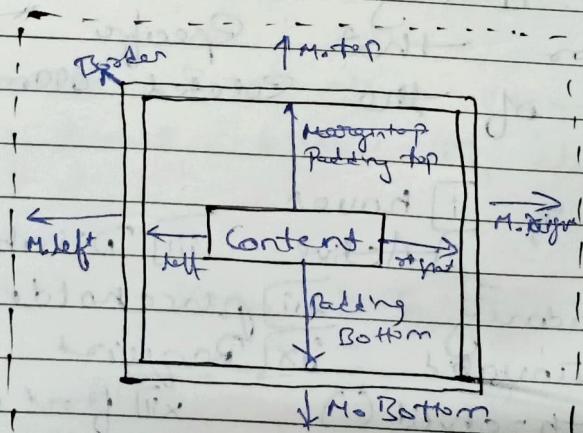
Pseudo elements are used to style specified part of an element.

ex: Before, After, first letter,

* Units

- i) px → fixed size
- ii) % → away relative to parent
- iii) em → (Element) in 16px (relative to parent)
- iv) rem → (Root element) away relative to html
- v) vw → Viewport width along the vertical
- vi) vh → Viewport height refer to screen
- vii) vmin →
- viii) vmax →

* Box model



$$\text{width of Box} = \text{left border width} + \text{padding-left} + \text{content width} + \text{padding-right} + \text{right border width}$$

$$\text{height of Box} = \text{top border width} + \text{padding-top} + \text{content height} + \text{padding-bottom} + \text{bottom border width}$$

→ Box-Sizing :- content-box (it is default)
border-box & it will change it

- every element in HTML when it loads in the browser it will create an box which includes margin, padding, border & content.
- In order to understand the box model property we need to use box-sizing property
- By default Box-Sizing property of all the element is content-box, that means whatever the width and height we specified for the element it will be entirely taken by content.
If we add any padding to this element the box width gets bigger and offset to next element.
In order to prevent this behavior we need to change the box-sizing property of all elements to border-box.
This means whatever the height and width is specified to the element which is having box-sizing as border-box it will be distributed to border, padding and margin as per given by the content.
- Padding ! It will be the space b/w the content and the border that need inside the border.

* Margin : It is the space outside the border. The margin can overlap on margin of next element.

* Padding / Margin

Padding - top :- 25px

- right :- 15px

- bottom :- 10px

- left :- 30px

or

Padding :- 25px 15px 10px 30px
T R B L

or

Padding :- 25px 15px
T & B L & R

or

Padding :- 25px
- All Sides

* Border

border-width :- 1px

border-style :- Solid / dashed

border-color :- Red

or

border :- 1px Solid Red

* Sometime we can use outline property of border, it does not effect Box model.

* Font properties and families

- Font property is a shorthand for setting multiple font-related properties for an element. It allows you to set the font-style, variant, weight, size, line-height and family.
 - font-style :- normal, italic, etc.
 - font-family :- "Times New Roman", serif,

→ Font families

- i] serif
- ii] San-Serif
- iii] cursive
- iv] fantasy
- v] monospace.

* Display

1) Display - Block :-

All the block level element by default having the display property as block, that means this element will occupy the entire width of the screen.

If this element we want come side by side we need to change the display property either to inline or inline-block.

- block
- inline
- inline-block
- flex
- grid

none

classmate
Date _____
Page _____

ii) Inline

Some of the inline element such as ``, `<a>` and `<i>` will be having the `display` property as inline, thus elements will not take any width and height.

If we want to make the element to obey width and height we need to change the `display` property either to `block` or `inline-block`.

iii) inline-block

Some of the inline element such as ``, `<button>`, `<iframe>` having the `display` property of `inline-block`, that mean act as a inline element and obey the width and height.

* Image property

clear
we can use float
that it will allow for
new content

object-fit :- It is a property used to specify how audio and video should be resized to fit in its container.

→ Fit :- Stretches the object to fit the container

→ contain :- Scales the object to fit within the container while maintaining the aspect ratio.

* If we want to make an image background
→ to background

Mix-blend-mode: color-burn

CLASSMATE

Date _____

Page _____

→ cover :

Sometimes it will display letter bonding
(mean left space)

→ cover : Scale the object to cover
the entire container while maintaining
the aspect ratio.

* color property

- i) color :- we can directly give name (Red)
- ii) hexcode :- # ffffff or any # 92a8d1
- iii) Rgb :- Red green blue (201, 25, 76)
- iv) Rgba :- (201, 203, 255, 0.6) totally
- v) linear gradient :-
- vi) radial gradient :-

* Background image in CSS

- The background image property specifies an image to use as the background of an element.
- By default the image is repeated so it covers the entire element.

Properties

- i) background-image : url (" ")
- ii) background-repeat : no-repeat
- iii) background-position : right-top / top-left
- iv) " " - attachment : - scroll / fixed
- v) Background-size : - cover, / contain
- In Single line we can write
background : # 000000 or (" ") no-repeat right top ;
background-filter :-



Flex

i) Display :- flex

ii) flex-direction :- row, column, row-reverse, column-reverse

iii) justify-content :- flex-start, flex-end, center, space-between, space-around, space-evenly

iv) align-items :- start, end, center

main axis →



cross axis

v) flex-wrap :- used to wrap items after certain (wrap)

vi) align-content :- baseline, center, end, flex-start, flex-end.

vii) order :- It is used to change the item from one place to another.

viii) flex-grow :- It will take the extra width of place from the parent container.

ix) flex-shrink :- It will try to shrink the item, when the parent will become smaller.

→ It is a one-dimensional layout model.

→ Distribute space among items.

→ Control their alignment.

* In position relative we can align others
they inherit of writing position relatively to the
parent.

- transform & translate (o)

- filter & invert (o)

- perspective : oam

Date _____

Page _____

→ Position

- the all position property tells how element is positioned in a document.
- top, bottom, left and right property

i) Static :- It is the default value of position for all elements. It is positioned according to the normal flow of the document.

ii) Relative :- It will position the element from its normal static position, it will create a gap for its normal flow.

It will always inside the parent container.

iii) Absolute :- It will gets out from the parent container and it takes position from parent. It can be placed to any corner.

iv) Fixed :- This position removed from normal flow position. Then positioned according to the body. If we scroll down, the item will be at the same position according to the body even.

v) Sticky :- If we make position of an item sticky, it will be acts as relative without losing the top, left, right, bottom. On Scrolling it will be top of browser hits that position then it will stick a sticky closely to top.

Position : sticky, webkit - sticky, o - sticky

* Box-Shadow : horizontal, vertical, blur
* text-shadow:

Date _____
Page _____

* Transition and transform Property

i) Transition :

transition property is used to control the speed to transform from one thing to another thing.

e.g. Box → circle

- transition - property
- transition - delay : 2s
- transition - timing - function
- transition - property delay
- transition

ii) Transform :

transform : scale (1.1), rotate, skew, translate, translateX, translateY, x, y

It is used to rotate, translate, skew, scale the element in 2 dimension.

- translateX () :- only for x-axis
- translateY () :- only for y-axis
- Scale x () :- only for x-axis
- Scale y () :- only for y-axis
- Scale (x, y) :- From center it will increase in x and y axis
- Rotate (25 deg) :- Rotate the element
- Skew (25 deg) :- give slope or bend

* Overflow : The content do not溢出溢出 when content is greater than its fit size.

* Overflow : scroll, hidden, visible, auto.

overflow = x and overflow = y

→ Media query in CSS

Media queries are a key part of responsive web design, as they allow you to create different layouts depending on the size of the screen.

Ex: @media only screen and (max-width: 600px) {

body {

background-color: #fff;

}

→ min-width

→ max-width

* Animation

The animation are written in the cell to animate the things, when we declare one time and use it many times.

@keyframe Name {

0% {

}

25% {

}

50% {

}

100% {

}

}

TO

{

}

After writing this add the animation-name to the elements so it can perform the change.

-Animation-name :-

-Animation-duration:-

-Animation-delay

-Animation-direction : alternate, reverse

-Animation-fx mode :

* Variables

The var() function is used to insert the value of a class variable.

Class variables have access to the DOM, which means that you can create variables with local or global scope.

:root {

--primary-color : Red ;

y

color : var(--primary-color)

* Visibility or opacity

visibility : hidden, visible

opacity : 0 to 1

* child - Selector

: first-child

: last-child

: nth-child

not (:first-child)