

CSS:- Cascading Style Sheets

→ Basic of CSS Format:-

Syntax:-

Selector → `h1 {`

Property `color: red;`
 ^ ^
 | |
 Value

→ How do we include styles?

- Inline:

Writing style directly inline on each element.

`<h1> Style = "color: red;" <h1> Hello world!`

Internal CSS/Using <style> tag:

Style is added using the `<style>` element in the same document in `head` tag.

`<style> {`
 `h1 { color: red;`
 `}`
`</style>`

External Style Sheet:- Writing CSS in a separate document & linking it with HTML file.

Linking HTML with CSS file

<head>

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

Color Property:-

used to set the color of foreground

color: blue;

color: #ff0000;

color:rgb(214, 122, 127);

Background Color Property:- used to set the color of background.

background-color: purple;

background-color: #ffffff;

Color Systems

Named Color.

Color = black;

Each browser recognizes around 140 named colors.

Eg:- red, blue, pink, purple, magenta, teal etc.

⇒ RGB
Red Green Blue.

Color: $rgb(214, 122, 127)$;
red green blue
value

red is $rgb(255, 0, 0)$

black is $rgb(0, 0, 0)$

blue is $rgb(0, 0, 255)$

green is $rgb(0, 255, 0)$

⇒ Hexadecimal.

Color = #fffff;

Letters: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

hexacode: - 00-ff

#000000

#ff0000

#00ff00

#0000ff

#fb851c
red green blue
value

Text Properties

text-align: center

font-weight

text-decoration

line-height

letter-spacing

font-size

⇒ text-align:-

text-align: left / start

text-align: right / end

text-align: center

text-align: justify

⇒ font-weight:-

font-weight: normal / 400

font-weight: bold / 700

font-weight: 100

100 - 700

⇒ text-decoration: Sets the appearance of decorative lines on text like underline.

text-decoration: underline; Suraj

text-decoration: overline; Suraj

text-decoration: line-through; Suraj

text-decoration: green underline; Suraj

text-decoration: wavy overline; Suraj

text-decoration: dotted line-through; Suraj

text-decoration-style: solid; (double, dotted, dashed, wavy)

text-decoration: double blue overline;

text-decoration: none;

⇒ line-height: - Controls the height of the line of text

line-height: normal;

line-height: 2.5;

line-height: 32px;

letter-spacing: controls the horizontal spacing behavior between text characters.

letter-spacing: normal;

letter-spacing: 10px;

Ex: I & on Man

Font-size Units in CSS:

Absolute

px

pt

pc

cm

mm

in

Relative

em

rem

ch

vw + many more

Pixels (px): most commonly used absolute unit. 1px = 1 pixel. Not suitable for responsive websites. font-size: 50px;

Font-family:- specifies a prioritized list of one or more font family names.

font-family: arial;

font-family: didot, sans-serif, Times new roman;

Text-transform:-

text-transform: capitalize;

text-transform: uppercase;

text-transform: lowercase;

text-transform: none;

→ Checking Priority in CSS

inline 0 2 1 (2)

ID class element

Specificity same → ^{then} Cascading Property.

मी बाट ज आरेता
के बो ग्वाह उरेता

Selectors :- in CSS

Element Selector h1, h2... h6, p, span, etc.

Selector

h1 {

 color: red;

} ↑

Property

Select all element of the same type

Universal Selector

Universal Selector

*

 Property : Value;

 }

Select everything in a document

h1, h2 {

 color: #b92b2b;

}

ID Selector:-

Selects an element based on the value of the element's id attribute.

```
#myid {
```

```
    Property: value;
```

```
}
```

Class Selector:-

Selects an element based on their class attribute.

```
.myclass {
```

```
    Property: value;
```

```
}
```

Descendant Selector:-

```
div p {
```

```
    Property: value;
```

```
}
```

Selects all paragraphs inside divs

Adjacent-Sibling Combinator

Eg:- Heading 2 that comes immediately after any paragraph
Sibling Combinator

Pt h3 {

Properties: value;

}

Child Combinator:-

Eg:- Selects all buttons which are direct children of Spans

Span>button {

Properties: value;

}

Attribute Selector:- Select elements based on the presence or value of a given attribute

input[attr = "value"] {

type = "text"

Properties: value;

3

Pseudo Classes

A keyword added to a

Selector that specifies a special state of the selected elements.

1. Hover
2. active
3. Checked
4. nth-of-type

Hover:- when moving the mouse over the element then change effect.

button:hover {

background-color: black;

color: white;

}

Active:- when clicking the element then apply changes.

p, button:active {

color: black; background-color:

green;

Checked:-

input[type="radio"]:checked {

color: red;

font-weight: bold;

}

~~the nth-of-type!~~

use for offer again like 1, 2, 12
in counting 1 12

Post: nth-of-type (2)

2 Apply effect
3
4 APPLY effect

by color: orange;

3

pink

purple

yellow

green

blue

red

black

white

grey

light grey

dark grey

light blue

dark blue

light green

dark green

light red

dark red

light orange

dark orange

light pink

dark pink

light purple

dark purple

light yellow

dark yellow

light green

dark green

Pseudo Elements:-

A Keywords added to

a Selector that lets you style a specific part of the selected elements.

:: first-letter

:: first-line

:: Selection. use for drag selection by mouse

h1 :: first-letter {

color: black;

3

CSS:- Cascading StyleSheets.

→ apply when same element.

What is cascade in CSS?

The CSS Cascade algorithm's job is to select CSS declarations in order to determine the correct values for CSS properties.

```
h2 {  
    color: yellow;  
}  
h2 {  
    color: red;    ← final color is blue  
}  
h2 {  
    color: green;  
}
```

Selector Specificity: - [id > class > element]

Specificity:- Specificity is an algorithm that calculates the weight that is applied to a given CSS declaration.

Hundred place .

. Tenth place
class
attribute &
Pseudo-class

One place
element &
Pseudo-element

!important: To show the most specific thing in document.

h2 {

background-color: blue !important;

h3 {

Inheritance in CSS:

<div> ← apply effect
↓
<p> ← also apply color: red;
↓
 ← " " " "

<div>

{ background-color:

<p>

</div>

Some element not inherit when for that:-

input {

background-color: inherit;

button {

Box Model in CSS

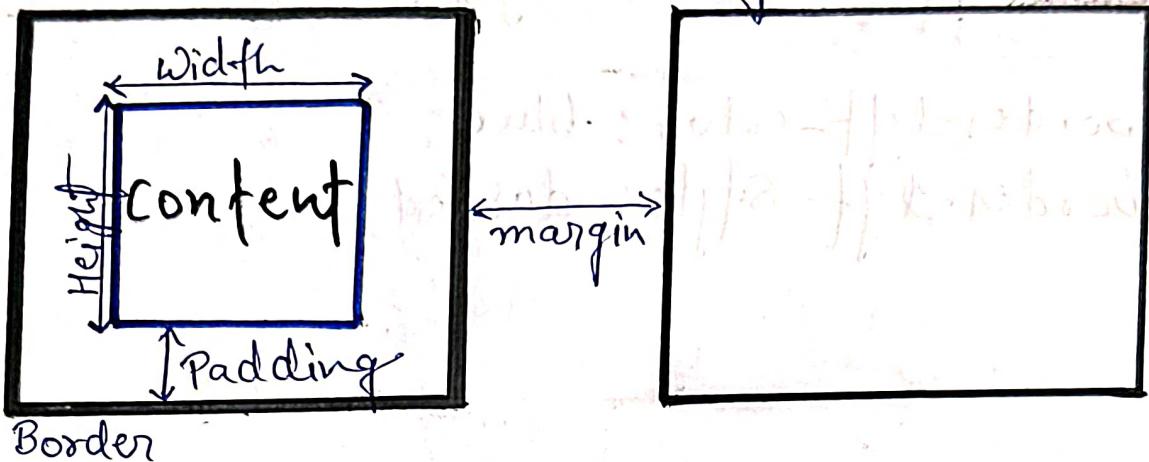
① Height

② Width

③ Border

④ Padding

⑤ Margin



Height:-

By default, it sets the "content" area height of the element.

```
div {  
    height: 100px;  
}
```

Border:-

use to set any element's border.

* border-width

* border-style.

* border-color.

Border shorthand.

[width | style | color]

```
div {  
    border: 2px solid blue.  
}
```

Border Sides To control an individual side of the box:

* border-left, border-right, border-top, border-bottom

```
div { border-left-color: blue;  
      border-left-style: dashed;  
}
```

```
div {  
      border-top: 5px dashed blue;  
}
```

Border Radius:

used for rounding the corners of an element's outer border edge

```
div {  
      border-radius: 15px;  
}
```

```
div {  
      border-top-right-radius: 50px;  
}
```

Padding:-
* padding-left, right, top, bottom

div { padding-left: 50px; }

div { padding: 2px 5px 2px 5px; }

Padding shorthand:

#1. Pad all sides.
padding: 50px.

#2 top & bottom / left & right
padding: 1px 2px;

#3 top | left & right | bottom.
padding: 10px 2px 3px;

#4 top | right | bottom | left.
padding: 1px 2px 3px 4px;

Margin:-

To set spacing on the outside of border.

- * Margin-left
- * Margin-top

- * Margin-right
- * Margin-bottom

div { margin-left: 50px; }

Similar as
Short hand shorthand
padding shorthand

Display:- It sets whether an element is treated as a block or inline element and the layout used for its children.

display: inline

display: block

div {

display: inline-block;

Units in CSS:-

Absolute	Relative
px	%
pt	em
pc	rem
cm	ch
mm	vh
in	vw + many more

Percentages (%) :-

It is often used to define a size as relative to an element's parent object.

e.g:-

width: 33.33%; // relative to the parent

margin-left: 50%; // relative to the parent

Em:-

Relative to font size of the parent, in the case of typographical properties like font-size, and font size of the element itself, in the case of other properties like width.

#outer {

font-size: 20px;

#inner {

font-size - 2em;
padding - 1em; } relative

Drawback of Em:-

Snowball - in nested elements.

Rem (Root Em):-

Relative to font size of the root element.

CSS - NEXT 8 steps

Alpha channel:

Sets the opacity for a color.
Ranges from 0 to 1.

`rgba(255, 255, 255, 0.3)` Hidden visible.

opacity:

Set opacity for the element.
Ranges from 0 to 1.

`opacity: 0.5;`

CSS Transitions:

Transitions enable you to define the transition between two states of an element.

`div {
 transition: 2s; extra.
}`

`transition-timing-function: linear, ease-in, steps(6, end);`

`transition-delay: 1s;`

`transition-duration: 1s;`

Shorthand:

Property name | duration | timing-function | delay

`transition: margin-top 2s ease-in-out 0.2s;`

```
div { margin: 0 auto; height: 10px; transition: margin-top 2s ease-in-out; }  
div:hover { margin-top: 400px; }
```

C.S.S. Transform:-

This property lets you rotate, scale, skew, or transform an element.

① Rotate :-

transform: rotate(45deg);

② Scale:-

transform: Scale (0.5); }
 (0.5, 2),
 Scale x (0.5);
 Scale y (2);

③ Translate!

transform : template (SOPX, SOPX)

(4) Skew

transform: Skew (30 deg); ^{exp!}

transform: rotate(30deg) translateX(50px);

Box Shadow: It adds shadow effects around an element's frame.

box-shadow: $x\text{-axis}$ $2px$ $2px$ $10px$ $Blur$ color; $y\text{-axis}$ green;

Background Image:

lets you set an image as a background.

background-image: url ("img/obj.jpg");

background-size: contain (cover / auto);
means → repeat crop stretch

Creating a Card Hover Effect:

```
<div class="card">
  <img src="" alt="">
  <div class="description">
    <h3> Known person </h3>
    <p> Surf's up & find it! </p>
  </div>
</div>
```

Same as if it's copy-

• card:hover {
 box-shadow: 0 8px 16px black;
}

.card {
 height: 250px;
 width: 200px;
 border: 2px solid black;
 display: inline-block;
 margin: 10px;
 border-radius: 5px;
 box-shadow: 0 4px 8px black;
 position: relative;
}

.description {
 padding: 0 8px 0 8px;
 width: 100%;
 border-radius: 5px 5px 0 0;
}

Position:- The position CSS property sets how an element is positioned in a document. The top, right, bottom, and left properties determine the final location of positioned elements.

- * static
- * relative
- * absolute
- * fixed

Position of static:-

No effect, this is default value.

static of

position: static;

before, after
 no change.

Relative:-

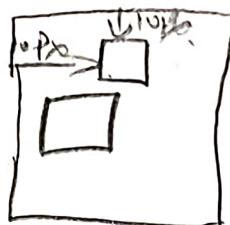
The offset is relative to itself based on the values of top, right, bottom and left.

position: relative;

change according to before place.

Absolute) The element is removed from the normal document flow, and no space is created for the element in the page.

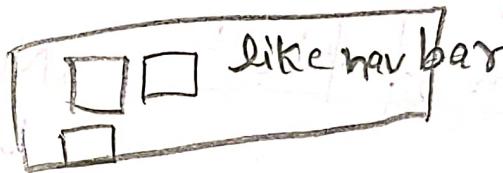
If is placed relative to the initial containing block.



Fixed!

If is positioned relative to the initial containing block established by the viewport.

Position: fixed;



Flexbox: → responsive

→ It is an one-dimensional layout method for arranging items in rows or columns.

```
{ display: flex; }  
flex-direction: column; Row  
row-reverse  
column-reverse
```

Justify Content

```
{ display: flex-direction: Row; }  
justify-content: flex-start;  
flex-end;  
center;
```

2 {

```
justify-content: space-between;  
space-around;  
space-evenly;
```

flex-wrap:-

flex-wrap: nowrap;
wrap;
wrap-reverse;

Align item:-

align-items: center;
flex-end;
flex-start;
baseline

Align content:-

align-content: flex-start / flex-end / center

align-content: space-between / space-around / evenly

align-content: baseline;

Align self:-

Aligns an item along the cross axis

align-self: flex-start;
flex-end;
center;
baseline;

Flex Size:-

⇒ flex-basis: 100px;

⇒ flex-grow: 1;

$\Rightarrow \text{max-width: } 400 \text{ px};$

$\Rightarrow \text{min-width: } 200 \text{ px};$

$\Rightarrow \text{flex-grow: } 3;$

flex shorthand:-

* flex-grow / flex-shrink / flex-basis.

flex: 2 2 100px;

* flex-grow / flex-basis

flex: 2 100px

* flex-grow (uniforms)

flex: 2;

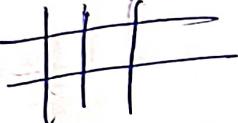
* flex-basis

flex: 100px;

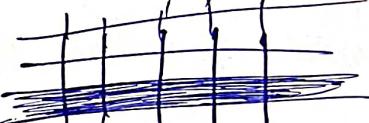
CSS Grid: Setting a container's display to grid will make all children grid items.

Container {

 display: grid;

→ Grid lines 

→ Grid cell 

→ Grid Track 

Grid Template:-

They define the

lines & track sizing.

grid-template-rows: 100px 100px 100px;

grid-template-columns: auto auto auto;

grid-template-rows: repeat(5, 1fr);

grid-template-columns: repeat(3, 1fr);

Grid gaps

They define the gaps b/w the items.

row-gap: 10px;

column-gap: 20px;

grid-gap: rowgap columngap;
10px 20px

Grid columns/rows

One of

grid-column-start: 1;

grid-column-end: 2;

3

grid-column: 1 / 3;

Common properties

- * justify-items: (container) → horizontal
 - * justify-self: (item)
- * align-items: (container) → vertical
 - * align-self: (item)
- * place-items
- * place-self

CSS Animations

elements.

@ Keyframes myName {

from { font-size: 20px; }

to { font-size: 40px; }

}

box {

animation-name: fontAnimation;

animation-duration: 2s;

animation-timing-function: ease-in;

animation-delay: 0s;

animation-iteration-count: 3;

animation-direction: normal;

}

@ Keyframes width animate {

from { width: 10px; }

to { width: 400px; }

}

Animation shorthand :-

animation: myName 2s linear 3s infinite normal;

exp:-

animation: widthAnimate 3s ease-in 0s infinite normal; 1/2

% in Animation

@keyframe myName {

0% { font-size: 20px; }

50% { font-size: 30px; }

100% { font-size: 40px; }

}

Media Queries :- Help create a responsive website

media features - width (of viewport)

@media (width: 400px) {

div {

background-color: red;

// (min-width: 700px)

(max-width: 200px)

@media (min-width: 200px) and (max-width: 300px) {

div {

background-color: red;

Media Queries Orientation Responsive

```
@media (orientation: landscape){
```

```
    div{
```

```
        background-color: red;
```

```
}
```

```
}
```

```
#pets{
```

```
    display: flex;
```

```
}
```

```
@media (max-width: 400px){
```

```
#pets{
```

```
    flex-wrap: wrap; justify-content: space-around;
```

```
}
```

Z-index:-

It decides the stack level of elements.

Overlapping elements with a larger z-index cover those (with a smaller) one.

```
div{ position: relative;
```

```
    z-index: 1; /* default(0)
```

```
}
```

```
    1, 2, 3.
```

```
    auto
```

```
    top-bottom
```

Basic Design Principles.

* For better looking websites.

Color Guidelines:- Visit Canva color wheel site.

Typography:-

Font families.

⇒ Sans-Serif.

⇒ Serif

⇒ monospace

⇒ Cursive

⇒ fantasy

Professional work.

* Times new roman

* Calibri

* Roboto

use max two font in one section like header, ~~section 1~~ section 1

Google fonts:-

<https://fonts.google.com>

Font:-

⇒ google icons free available

⇒ Font Awesome

overflow: hidden;

overflow: scroll;

Bootstrap:-

Framework → toolkit.

- ⇒ Work on flexbox concept.
- ⇒ Pre-built components

- ★ button
- ★ forms
- ★ navbar
- ★ Card.

Definition:-

"Bootstrap is a powerful, feature-packed frontend toolkit. Build anything - from prototype to production - in minutes."

Benefits:-

- ★ Makes development faster & easier.
- ★ Good choice of pre-build components
- ★ Good browser compatibility.

⇒ CDN:- Content delivery Network.

What will we cover?

Components

Pre-built and reusable.

Piece of user interface (UI)

e.g. button, forms.

Layouts

Overall Structure and arrangement of the elements.

e.g. container, grid

Layout:

.container :- Default font size style; layout set Auto

.container-sm

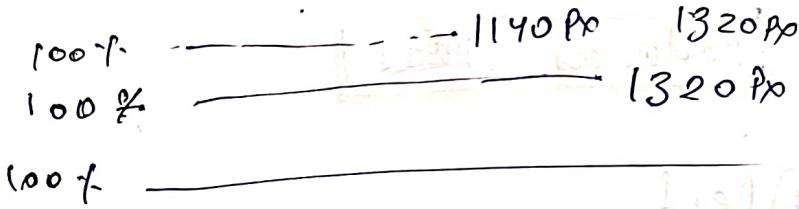
" - md

" - lg

" - xl

" - xxl

" - fdeq



Components

*Buttons :- (class = "btn") - basic button style A.

<div class = "btn btn-primary" > </div >

btn - btn-secondary

warning

success

Danger

info

btn - btn-outline-success

btn - btn-Danger Disabled

* Badges

`<h3> notifications `

`=> <button class="btn btn-primary">`

Followers

` 100+`

` </button> </div>`



* Alerts

`<div class="alert alert-primary" role="alert">`

A simple primary alert - check if out!

`</div>`

* Navbar

* Cards:

* Grid Layout:

Extra small xs 376

Small sm 576

Medium md 768

Large lg 992

Extra large xl 1200

Extra extra large xxl 1400