PAGE No.
DATE

CSS → Cascading Style Sheet

Element selector / type selector / tag selector
↳ particular tag ki property change karna

Class selector → (.) → se access
↳ particular class name dena

Id selector → Id is unique
↳ (#) → se access

Pseudo-classes selector → button : active
↳ special state to change particular prop
↳ button : hover

Universal selector → (*) is used to cover entire thing jismai bhi lagaye
↳ (*) → only this means fullpage

* **Inline CSS** → jab tags ke ander hi styles implement karde

* **Internal CSS** → jab <head> tag ke ander <style> declare kiya ho.

External CSS → jab alag se .css file banke <link> tag mai dale.
↳ <link rel="stylesheet" href="style.css">
rel → relationship
href → path of CSS file

PAGE No.
DATE

Specificity → tag selector
→ class selector
→ Id selector
→ Inline CSS

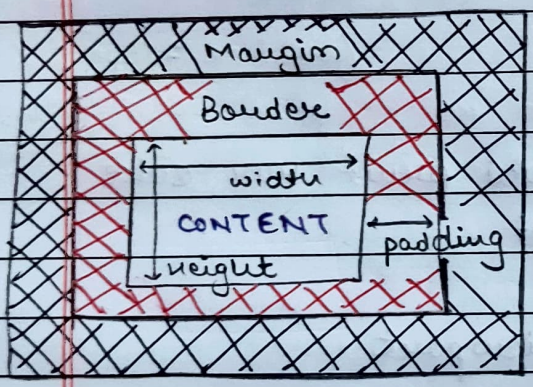
Nice aate hue importance increase

bad practice

** **!important** → jiswai ye lag gaya vho hi sab kuch

BOX-MODEL → koi bhi chiz page mai add karoge vho rectangular form mai ayegi

→ may have → margin, border, height, width, padding



Margin → 2 div ya element ke beech ka gap

Padding → Border or content ke beech ka gap

* By default margin is 8 px.

Colors → Hexadecimal (# - - - - - -) values from 0→f
(red) (green) (blue) (RGB)

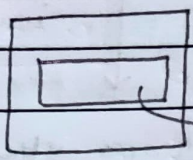
→ RGB (rgb(_, _, _))
(red) (green) (blue)

* 146 predefined colors

padding → add karne se content ki width size same rhta hai box bada ho jata hai

box-sizing : border-box
↳ ye karne se content ka size kam ho jata hai box ka size same rhta hai

Unit → em → relative to parent element font size.



rem
↳ relative to root
↳ html

→ jo value html tag mai hogi voi hoga

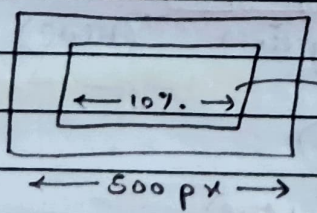
→ 1 rem = 16px → ye value html ki font size = 16px;

* agar parent div nhi hua to body tag ka font size → 16px lega.

vw (viewport)
↳ jitne area mai print kar sake.
↳ 1/100 x width of view port

vh → 1/100 x height of viewport

percentage unit → width: 10%.
↳ parent div ke dimension ka % nikalo voi answer



10% of 500 px = 10/100 x 500 = 50px

*** Q diff b/w 1% and 1vw??
→ 1% is relative to the parent container while 1vw is relative to the viewport

PAGE No.
DATE

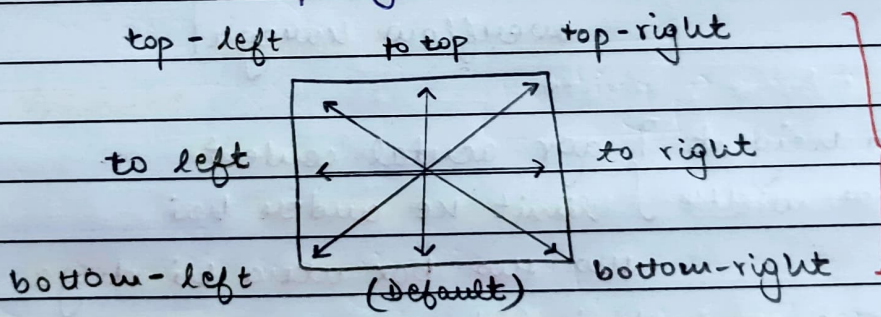
CSS gradient → smooth transition b/w two or more colours

- **Linear** → movement from ↑↓, →←, ↘↖, ↗↙
- **Radial** → movement bhar ki side colour change
- **Conic** → cone ke top view se colour

* **Linear gradient** → background-image : linear gradient
by default top to bottom ← (—, —);

→ background-image : linear gradient (to right, –, –)
→ " " " " (to left, –, –)

Diagonal
→ (to bottom right, –, –) → (to bottom left, –, –)
→ (to top right, –, –) → (to top left, –, –)



Can use multiple colors.

* angle bhi use kar skte hai → (90 deg, –, –)
(210 deg, –, –)

for transparency
→ rgba (–, –, –, —) → (0 to 1)
ye hai for transp

Syntax
→ background-image : linear gradient (rgba(–, –, –, –)
→ background-image : radial gradient (rgba(–, –, –, –)
→ background-image : conic gradient (–, –, –)

Shadows → text shadow

↓

Box-shadow

↓

box-shadow : 10px 10px;

↳ jo text ka colour voli shadow ka colour if no color mentioned.

↳ rest same as text shadow → blur and all

↳ To change position use (-) signs

text-shadow : 3px 3px red;

↳ horizontal → vertical

Multiple shadow ↴

text-shadow : 3px 3px red,
6px 6px blue;

↳ blue

Dimension properties

↳ max height } iss limit ke
↳ max width } baad content overflow karega

Solution

overflow : scroll

* min height } agr acutal content
* min width } limit ke under hai
tbo tbo box utna hi rhega. lekin
agar content zada hua tbo box apne ap inc

overflow property

(default) → overflow : visible (overflow content box ke bar visible hoga)
→ overflow : hidden (overflow content not visible)
→ overflow : scroll (overflow mai scroll add)
→ overflow : auto

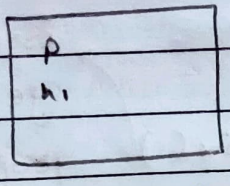
if content more tbo "scroll" ke trha behave karega

if content is less tbo behave like visible.

PAGE No.
DATE

Position Property → Position : _____

* **static** (by default) → jaise chize add karoge vaise ayega.



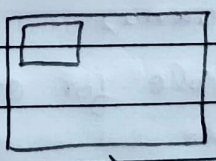

```
<div>
  <p>
  <br>
</div>
```

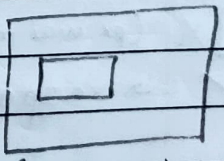
* **relative** (top, bottom, left, right)
  → Eg: position : relative ;
  top : 100px ; right : 70px ;
  left : 50px ; bottom : 60 px ;

* **fixed** → apne position se kilaga nhi
  → scroll karne pe bhi vhi rhega
  → generally use view port (vw)

* **absolute** → ye position hota hai **nearest ancestor** ke according se vha bhi **tab jab** uski position : static na ho.
  → **if position : static** to vo **body** ke according position karta hai

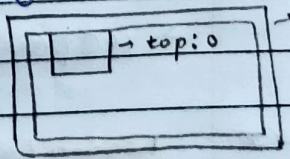
→ Eg :- `<div>` ← ancestor
`<div></div>`
`</div>`

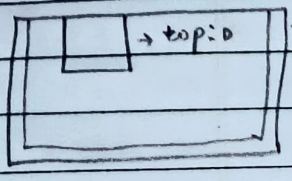

(wrt body)


(wrt ancestor)

* **sticky** → toogle b/w relative and fixed
  ↓
  scroll karne se pehle behave like relative (top, left, right, bottom)
  scroll karte hi fixed ho jata hai

→ top
→ left
→ right
→ bottom





2-D transforms

- translate (for movement)
- rotate()
- scaleX()
- scaleY()
- scale()
- skewX()
- skewY()
- skew()

matrix() → isse ye sari property use ho jayegi

* transform : translate (__, __);
  - X-axis
  - Y-axis

→ responsible for movement.

* transform : rotate (__);
  - angle → (+) clockwise
  - (-) anticlockwise

→ rotate karta hai.

* transform : scale (__, __)
  - horizontal
  - vertical

2px ✗ (px mai nhi likna)
② ✓

→ responsible for stretching

point values bhi use kar sakte hai isko

* transform : skew (__)
  - angle (45deg, 37deg etc)

→ responsible for tilting

** transform : matrix (__, __, __, __, __, __)
- scaleX
- skew(Y, X)
- scaleY
- trans(x, Y)

PAGE No.
DATE

3D Transformation → same as 2D bass
'z' axis add ho gaya

for this we need to turn on perspective

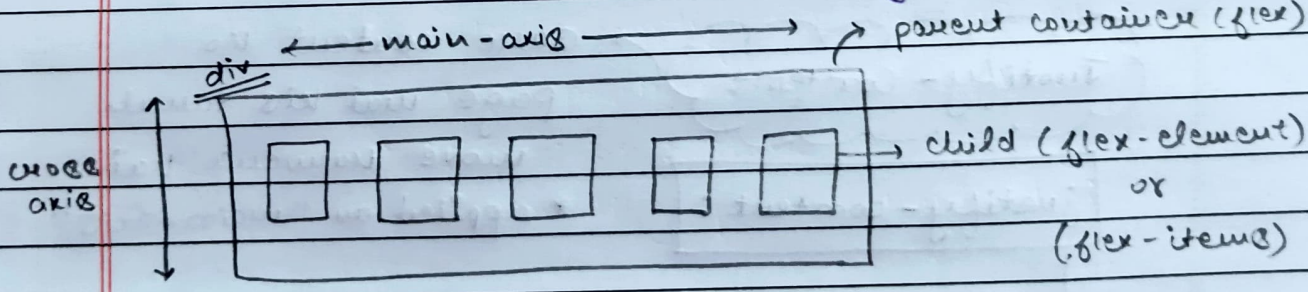
→ transform: perspective() translatez();
→ transform: perspective() rotatez();
→ transform: perspective() scalez();

ismai bina perspec ke bhi chlega.

ismai scaling z mai hui hai so dikhaye nhi dega if thoda rotate karenge (x, Y) mai the we can see the change.

perspective → humari eyes or screen ke beech dekhne wala gap (consider this)

FLEX BOX → Layout model → space distribution
→ alignment capabilities



display: flex → isko karne se main axis ke along align ho jata hai.

flex-direction: column → ek ke upar ek stack.

flex-direction: row → normal main axis ki trah.

flex-direction: column-reverse → bottom m set kar deta hai
flex-direction: row-reverse → right m set

flex-wrap → To fix the width taki stretch karte time utna ki rahe

→ flex-wrap: wrap → jaise width se kam hua, element niche shift

→ flex-wrap: no-wrap → NO CHANGE

→ flex-wrap: wrapreverse → jaise width se kam hua element upper shift

flex-flow → shortcut for flex-direction and flex-wrap dono ek sath use karne ke liye

** flex-flow: row (direction) wrap

Justify-content → sare content ko page mai kis tarah move karwana hai

justify-content:

* applied on "main-axis"

→ flex-start → all elements start mai ayenge
→ flex-end → all elements end mai ayenge
→ space-around → elements ke beech even spaces lekin both corner not equal
→ space-evenly → sb jagha equal space
→ centre → all element centre mai ayenge
→ space-between → elements ke beech mai gap hoga or element jo edge mai hai vho border se touch karenge.

align-item → applied on "cross axis"

* align-item : stretch → by default

baki sab same rehta hai sari property w.r.t cross-margin (vertical)

* align-item : baseline → sabko baseline mai align kar deta hai

** for fully centre → align-item : centre ;
→ justify-content : centre ;

gap → to have gap b/w elements.

→ row gap ⟶ row mai alag se gap.
→ column gap → column mai alag se gap.

align-content → used when row spacing is there.