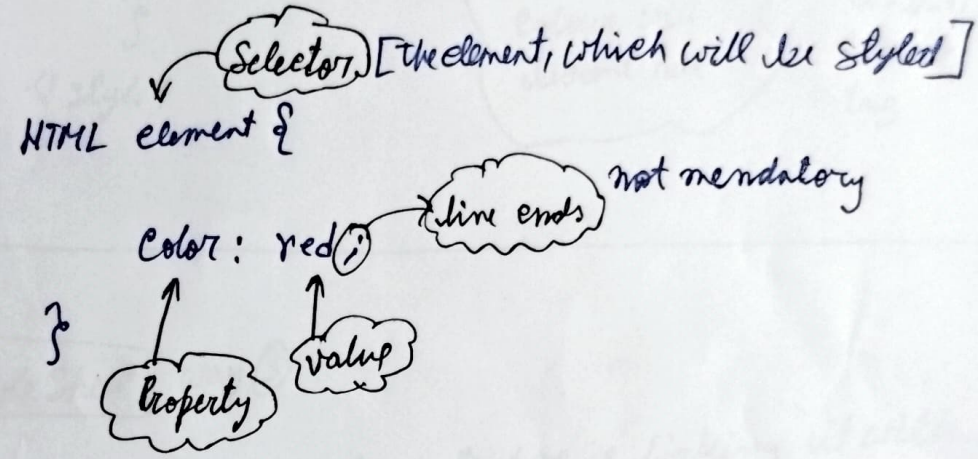
CSS → Cascading Style Sheets.

→ is used to describe the style of a document.

→ (properties) ex: text, color, bg color, button . . . .

Basic Format of CSS



How to include Styles?

Way ① **Inline Styles:** writing style directly on each element.

Ex: `<h1 style="color: red">Apna Col</h1>`

for text color

`<button style = "color : Blue ; background-color : orange ">help</button>`

for background color

<Style> tag way ② Internal Style

- Style is added using <style> element in same doc.

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

↩ By this all, h1 headings text colour will become red

will be written inside the <head> tag

External Style Sheet way ③

- Writing CSS in a separate doc & linking it with html file. (abcd.css)

Linking HTML with CSS File

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

↑ or full path of a diff folder

Color Property

- Used to set the color of foreground (Text)

- color: purple;

color: #ffffff; → color for white (other way)

color: rgb(214, 1222, 127); (another way)

⋮

Background color Property

- Used to set the color of background.

- background-color: purple;

" - " : #ffffff;

Color Systems ⇒ for defining colours in CSS.

Way ① Named Colors ⇒ eg: color: black;

Way ② RGB Colors ⇒ eg: color: rgb(0,0,0); [Black]

Scale ⇒ 0 to 255 ← Imp RGB Notes

- Red + Green → Yellow
- Blue + Green → like sky
- Blue + Red → Violet
- Red + Green + Blue → White

Way ③ **Hexadecimal / Hex Color**

- letters : 0, 1, 2 --- 9, A, B, C, D, E, F

eg: color : #FFFFFF ; [White]
R G B

- RGB ⇒ 0 - 255
- Hexcode ⇒ 00 - ff
- Short form ⇒ 00ff00 → 0f0

Imp **Text Properties**

- text-align
- font-weight
- text-decoration
- line-height
- letter-spacing
- font-size

---

① text-Align Properly

according to parent element not to doc or screen size

text-align : left/start , right/end , center , justify

for same horizontal alignment

② font-Weight

font-weight : normal (400) , bold (900) , 100 , bolder , lighter

Value ⇒ 100 - 900

③ text-decoration - sets the appearance of decorative lines on text like underline, overline, line-through

abc abc ~~abc~~

others

text-decoration : red underline ;
wavy "
dotted "
double "
dashed "
none ;

To remove underline →

④ line-height - controls the height of the line of text.

Values ⇒ normal , 2.5 , 50% , 100px

line 1 → Normal
line 2

line 1 ↕ → 2x of normal
line 2

⑤ letter-spacing - controls the horizontal spacing behavior b/w text chars.

Values ⇒ normal , 0px

abc a↔b↔c

Font-size units in CSS

Normal text → font size = 16px

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

1 inch = 96px

px is not suitable for responsive webs. We use relative unit there

font-family property - Specifies a prioritized list of one or more font family names.

- used for defining font type/style.

```
font-family : arial ;
font-family : avant garde, didot, sans-serif;
```

Backup family of fonts

* use cssfontstack.com for font styles

Up/lower case tranform - **text-transform**: uppercase; lowercase;

Day(2)

Selectors in CSS - used to select an individual or a group of element to add style on.

---

① **Element Selector** Already learnt

eg:
```
h1 {
  color: red;
}
```

We can select multiple selector by seperating them by comma

---

② **Universal Selector** - to select everything in a document.

eg:
```
* {
  Property: value;
}
```

---

③ **ID selector** - selects an element based on the value of the elements id attribute.

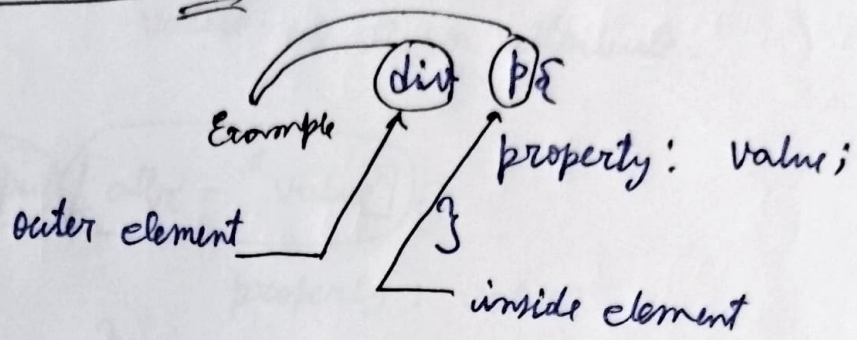
eg:
```
<h1 id = "myID">
#myID {
  property: value;
}
```

---

④ **Class Selector** - Selects an element based on their class attribute.

eg:
```
<h1 class = "a">
<button class = "a">
<p class = "a">
.a {
}
----
```

⑤ **Descendant Selector** Eg: Select all paragraphs inside divs.



---

* **Adjuslent Sibling Combinator** → used to combine multiple selectors

Eg: heading 3 that comes immediately after any paragraph

```
p + h3 {
    property : value;
}
```

---

* **Child Combinator**

Eg: Selects all buttons which are direct children of spans.

```
span > button {
    ----
}
```

Descendent → can select element that are multiple level down.

Child → only select the next descendent.

⑥ **Attribute Selector** - selects elements based on the presence or value of given attribute.

Eg: `input[attr = "value"] {`
`property : value;`
`}`

Pseudo Class - A keyword added to a selector that specifies a special state of the selected element(s).

Selector : hover {} (when hovering)
Selector : active {} (when clicked)
Selector : checked {} (used in radio/checkbox when selected)
Selector : nth-of-type (x) {} → x / xn

Pseudo Element - A keyword added to a selector that let you style a specific part of the selected element(s)

Select :: first-letter {}
Select :: first-line {}
Select :: selection {} (what will show after selection)
(before copy state) ↗

CSS: Cascading Style Sheets

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.

Ex

```
h2 {
    color: yellow;
}
.h2 {
    color: red;        ← final color will be red.
}
```

Selector Specificity

⇒ Specificity is an algo that calculates the weight that is applied to a given CSS declaration.

inline style	id	class, attribute & pseudo-class	element & pseudo element

∴ id > class > element

Inline styles have more specific than id.

PQ `<h1 id="greet" class="myClass">Hello !</h1>`

```
h1 {
  color: pink;
}
#greet {
  color: yellow;
}
.myClass {
  color: brown;
}
```

What will be color of h1 =>

→ element

→ id

→ class

∵ id > class > element

So, final color will be yellow.

PQ

```
<div>
  <p class="para">Paragraph</p>
</div>
```

```
div p {
  color: black;
}
.para {
  color: blue;
}
p {
  color: green;
}
```

→ I/0 C/0 E/1+1

→ I/0 C/1 E/0

→ I/0 C/0 E/1

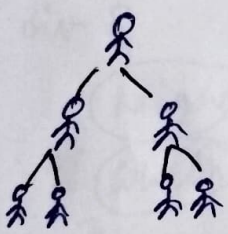
final color = blue

! important - to show the most specific thing in document.

```
H2 {
    color: blue !important;
}
```

most priority even from inline style

Inheritance in CSS



ex:

 ← if defined blue
↓
<p>
↓

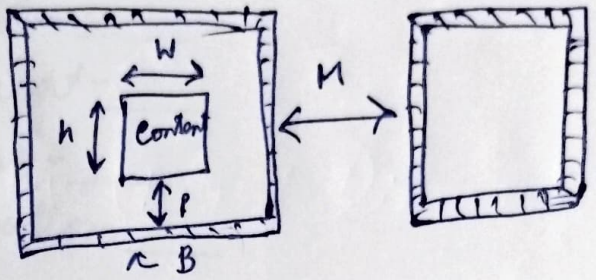
↓

} all will inherit blue color

some properties do not inherit like that. Ex: width, height, border, input, button....

Box Model in CSS

- Height
- Width
- Border
- Padding
- Margin



HTML element stores in a Box's form

① Height
② Width

- By default it sets the content area height of element.

Ex:
```
div {
    height: 100px;
    width: 200px;
}
```

Any empty element without any content takes no height & width.

height & width changes contents hei & wid not the div's

③ Border

- Border is set for outer lining area (opp of height)
- border-width *
- border-style * (dotted, solid, double lined ....
- border-color *

• Border Shorthand — border: width style color

Eg:
```
div {
    border: 2px solid blue;
}
```

Border Sides - to control individual side of box.

⇒ border-left-____
⇒ border-right-____
⇒ border-top-____
⇒ border-bottom-____

Border Radius - used to round the corners of an element's outer border edge

Eg: border-radius : 15px / 50%;

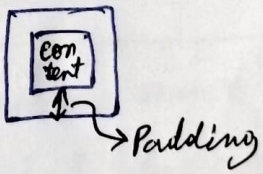
note: border-top-right-radius : 15px;
for individual corner

---

⑤ **Padding**

- padding - left / right / top / bottom

#



Padding Shorthand

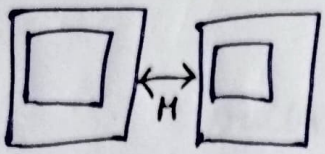
① for all sides
padding : 50px;

② top & bottom | left & right
padding : 1px 2px;

③ top | left & right | bottom
padding : 1px 2px 3px

④ top | right | bottom | left (C.W)
padding : 1px 2px 3px 4px;

⑤ **Margin** property - to set spacing on the outside of border.



margin - left / right / top / bottom

Eg: margin-bottom: 15px;

Same short hand like padding.

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

→ display : inline;
→ display : block;
** → display : inline-block; (due to limit ^ below)

→ display : none; use to hide

Block	Inline
Always start on a new line.	Do not start on a new line.
Take up as much horizontal space as possible (full width of container or browser)	only take as much horizontal space as required by content.
accepts height & width property.	dont accept height & width property.
horizontal & vertical margins both work.	Margin works horizontally not vertically.

Percentage Unit (%) - used to define a size as relative to an elements parent object.

```
<div>
    <div>
    <div>
</div>
```

```
width : 33.33%. // relative to the parent
margin : 50%.   // relative to the parent size
```

---

em - used to define font size of the parent / itself.

```
#outer{
    font-size: 20px;
}
#inner{
    font-size : 2em;
    padding : 1em;
}
```

2em → means 2x of outer div font size (40px)

1em → means 1x compared to font-size, padding will be of ⇒ 40px

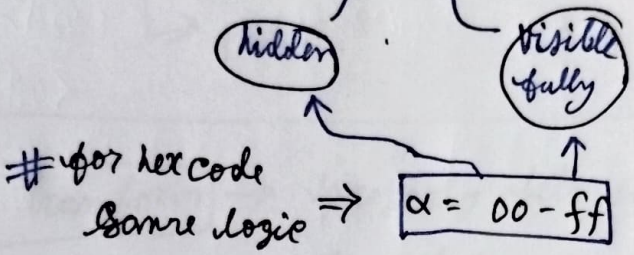
#Drawback: Snowball effect.

---

rem (root em) - font size of the root element.

Alpha Channel (a)

- Sets the opacity for a color.
- ranges from 0 to 1



#for hex code
Same logic ⇒ α = 00 - ff

rgba(255, 255, 25, 0.3)

#9dff00ff

Opacity

- Set the opacity for the element
- ranges from 0 to 1

#opacity : 0.5;

CSS Transition

- Transition enable you to define the transition b/w 2 states of an element.

Eg:
```
div {
    background-color : green;
    transition-duration : 2s;
}
div : hover {
    border-radius : 50%;
}
```

transition - duration
- timing-function
- delay

Values
linear;
ease-in-out;
steps
cubic;

Transition Shorthand

transition : property name | duration | timing-function | delay

Eg: transition : margin-top 2s ease-in-out 0.2s

transition-timing-function:

- values
  - linear
  - ease-in-out
  - steps
  - cubic

CSS Transform → properly lets you rotate, scale, skew or translate an element.

① Eg:

```
#one {
  transform: rotate(45deg)
}
```

for rotate an eelemt



② Eg:

```
#two {
  tranform: scale(1.5)
}
```

itll make the element 1.5 x x area

```
# transform: scale(2, 0.5)   (x : 5)
# transform: scale(X)(0.5)
# transform: scale(Y)(2)
```

③ Eg:

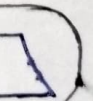
```
#twee {
  transform: translate(100px, 100px)
  tranform: translate(X)(100px)
                      (Y)
}
```

to move element by distance

④ Eg:

```
#four {
  transform: skew(30deg)
}
```

#



for tilt

Box Shadow - it adds shadow effects around an elements frame.

box-shadow: x offset | y offset | blur radius | color;

Eg: box-shadow: 2px 2px 10px green;

Background Image - lets you set an image as a background

background-image: url("loca")

background-size: contain; (repeat to fit in)
cover; (crop to fit in)
auto; (stretch to fit in)

Position

⇒ The position CSS property sets how an element is positioned in a document.

⇒ The top, right, bottom & left properties describe the final location of positioned elements.

1. Static
2. relative
3. absolute
4. fixed

① **Static** position

- here, the top, right, bottom, left & z-index properties have no effect.
- this is **default** value.

Eg: position : static;
top : 100px;] no work
left : 100px;]

② **Relative**

- the offset is relative to itself based the values of top, right, bottom, left.

③ **Absolute**

- the element is removed from the normal doc flow, & no space is created for the element in the page layout.
- It is positioned relative to its closest positioned ancestor, if any otherwise, it is placed relative to the initial containing block.

④ **fixed**

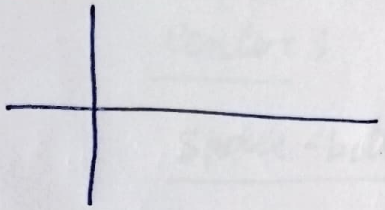
- The element is removed from the normal document flow, and no space is created for the element in the page layout.
- It is positioned relative to the initial containing block established by the view point.

flexbox - flexible box layout.

- It is a one-dimensional layout method for arranging items in row or columns.
- it helps to build responsive website.

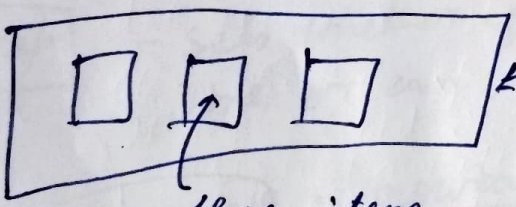
---

flex Model



#dir^n = row

General Example



---

eg:

```
div {
    display : flex;
    flex-direction : column;
                     row;
}
```

→ # it helps to contain elements in that layout without over flowing.

* here, container = div

---

① **flexbox direction**

→ it sets how flex items are placed in the flex container, along which axis & direction.

flexbox-direction : row ; (main axis, left to right)
row-reverse ; (right to left)
column ; (top to bottom)
column-reverse ; (bottom to top)

② **Justify Content** - Tells how the browser distributes space between & around content items along the main-axis.

Justify-content : flex-start;
flex-end;
center;
→ # defines where the content will set.

space-between; - # for space in b/w to cover full area towards main axis

space-around;

space-evenly;

---

③ **flex-wrap** - sets whether flex items are forced onto one line or can wrap onto multiple lines.

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

so it just moves to next line if content not fits in a line with defined area.

it wraps, it reverses the direction of cross axis

---

④ **Align Items** - it distributes our items along cross axis.

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

for align texts for each items in same height

Align Content - it sets the distribution of spaces between & around content items along a flexbox's cross-axis.

align-content: flex-start / end / center;
space-between / -around / -evenly
baseline;

Align Self - it align an item along the cross axis.

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

Note

Align-items → align whole content along ⊗ axis
align-self → align individual element " "
in priority → align-self > align-items

Flex Sizing

① **flex-basis** - it sets the initial main size of a flex item.

flex-basis : 100px;

② **flex-grow** - it specifies how much of the flex container's remaining space should be assigned to the flex item's main size.

flex-grow : 1 (default) ← (Takes available spaces x times)
2 (times)
3 "
: "

max-width/height : 400px
min-width/height : 200px

we can also set available spaces max/min value by these.

③ **flex-shrink** - it sets the flex shrink factor of a flex item.

flex-shrink : 1 (default), 2, 3 . . . .

Flex Shorthands

① flex-grow/ flex-shrink/ flex basis

flex : 2 2 100px;

② flex-grow / flex-basis

flex : 2 100px;

③ flex-grow (if unitless)

flex : 2

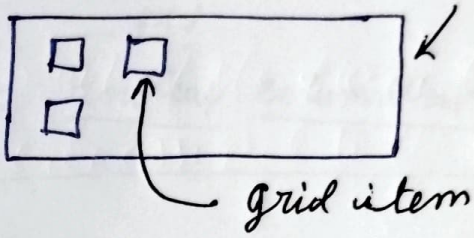
④ flex-basis (with unit)

flex : 100px;

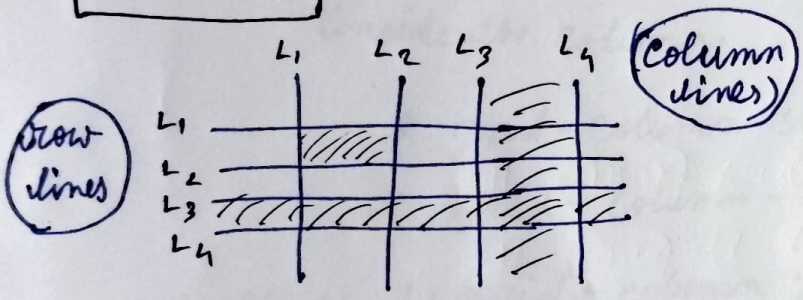
CSS Grid

- Setting a container's display to grid will make all children grid items.

```
container {
    display: grid;
}
```



Grid Model



Grid lines
Grid cells
Grid tracks
distance b/w 2 grid lines

Grid Template

- used to define the lines & track sizing.

* grid-template-rows : ___ ___ ___ ...
* grid-template-colums: ___ ___ ___ ...
* grid-template-r/c : repeat(count, 1fr)

1fr 1fr .... count times

1fr = 1 fraction of 100% width/ available space.

Grid Gaps

- used to define the gaps between the lines.

* row-gap : ___ px
* column-gap : ___ px;
* Shorthand: grid-gap : rowGap columnGap; / ___ px;

Grid Columns

- Used to define an items starting & ending position inside the column.

* grid-column-start : line-number ;
* grid-column-end : line-number ;
* Shorthand: grid-column : start-col / end-col;
  start-col / span number;

Grid Rows

- Same as columns.

Grid Properties

start, end, center

justify-items :
justify-self : > Horizontal

align-items :
align-self : > vertical

place-items :
place-self :

CSS Animations

- It is used to animate CSS elements

@ keyframes Name {
    from { ;}
    to { ;}
}

Properties

① * animation-name :
② * animation-timing-function :
③ * animation-duration :
④ * animation-delay :
⑤ * animation-iteration-count :
⑥ * animation-direction :

(for how many times animation display)

defined inside the container/selector

Values
normal ⇒ starting → end
reverse ⇒ end → starting
alternate ⇒ starting ↔ end
alternate-reverse ⇒ end ↔ starting

Properties Shorthand : animation : 1 2 3 4 5 6

0% in Animation

@ keyframe name {
    0% { ;}
    50% { ;}
    100% { ;}

Media Queries

it helps to create a responsive website.

media features - width (or viewport) (example)

```
@media (width: 400px) {
    div {
        background-color: red;
    }
}
```

(min-width:) {}
(max-width:) {}
@media (min-width:) and (max-width:) {}

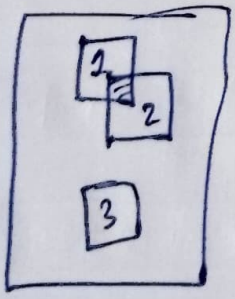
Orientation Media

@media (orientation: portrait / landscape)

---

z-index

- it decide the stack level of elements
- overlaping elements with a larger z index cover those with a smaller one.



* z-index: auto (0) [default]
* z-index: 1 / 2 / .... +ve no [upper]
* z-index: -1 / -2 / ... -ve no [lower]

---

z-index doesn't work for default / static position element