* CSS - cascading style sheet.
  - Used to styling our webpage.
  - Makeup and beautiful our webpage.

* Framework - Bootstrap 5.

* Three way to write our CSS.

* Inline CSS - we can write style attribute

* External CSS - We can write CSS in style tag above head tag.

* External CSS - We can create external file style.css and then import in our file using Link.

* selectors - fetch the html elements for styling and for more work.

* simple selectors. 1) class selector
  2) id selector
  3) name selector or tag selector.

* Combinator selector

1) Desendent selector

select all Html element which are in the specific element.

Example -
```
<style>
    div_items li {
    color: black
    }
</style>
```

Example -
```
<body>
    <div class="div-items">
    <li> HTML5 </li>
    </div>
</body>
```

*2) child selectors.

select all the elements which are children of specified element.

Example:-
```
<style>
    div p { color: red; }
</style>
```

Example:-
```
<div>
    <p> hellow </p>
    <p> HTML5 </p>
</div>
<p> hellow </p>
<p> css </p>
```

--- parent

} --- child.

} --- siblings.

3) General sibling selectors.

select all the elements which are sibling of specific element

Example :-
```
<style>
    div ~ p, h1 {
        color: red
    }
</style>
```

Example :-
```
<div>                          --- parent
    <p> Hello </p>
    <p> world </p>             } --- child
</div>
<p> Hello </p>
<p> world </p>                 } - siblings
<h1> Reactjs </h1>
```

4) Adjecent sibling selectors.

Select element which are direct sibling of specific element.

Example:
```
<style>
    div + p {
        color: red
    }
</style>
```

Example:-
```
<div>                          --- parent
    <p> Hellow </p>            } -- child
    <p> HTML </p>              }
</div>
<p> Hellow </p>                } ← direct sibling (1)
<p> CSS </p>                   } --- siblings.
```

* Universal selector.

It is used to styling all html content
Indicate by (*)

Example:- <style>
```
*{
  margin : 0;
  padding : 0;
  box-sizing : border-box;
 }
```

* pseudo classes.

:link , hover , visited , active, first-child , last-child,

nth-type(2) · nth-type(even)odd) , focus.

* pseudo element

:: before , :: after , :: selection , ::first line, ::last-line

* link - yet to vinvisited
* visited - yet to visited
* hover - when hover
* active - when select or click.

```
p:before {
        content: Hellooo        --- by default property.
        }
```

:: selection --- selection color change.

* Box model in css.

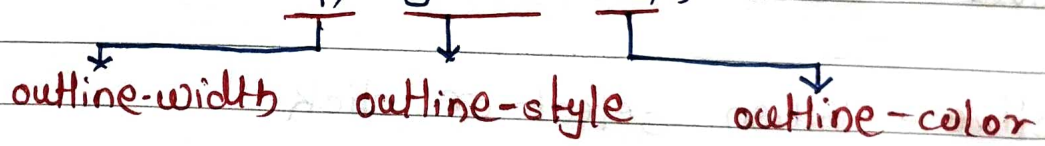
- everything in a html is box
- It can be inline level either block level element
- It contain border, margin, padding.

border-style: solid, dotted, dashed, groove, Inset.

* Example: border: 2px solid black;

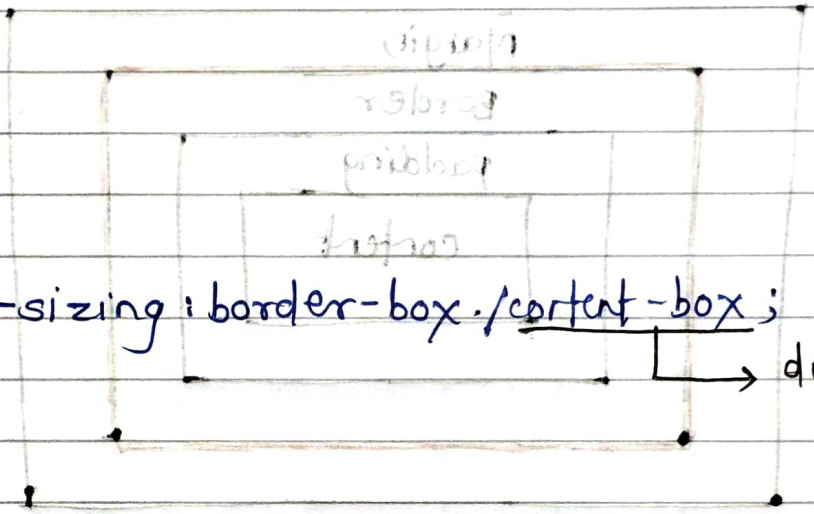
* outline - Apply outside the border

* Example: outline: 2px groove red;



* Box sizing Border Box.

Remove the outside border and adjust into inside box.



Example:

```
* {
  box-sizing : border-box / content-box ;
                                  └→ default
  }.
```

* Border Radius Box.

Helps to give curve shape, rounded shape

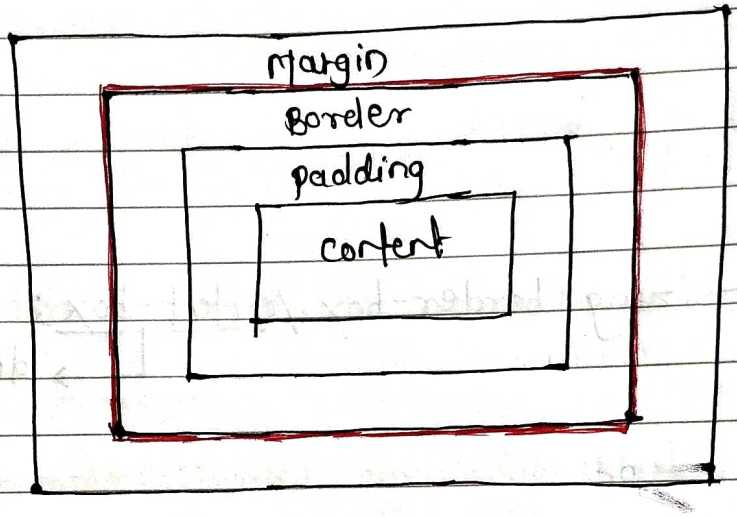
Example:

① border-radius : 50px 20px ;
                  ↓ ↓
                  Top bottom

② border-radius : 10px 20px 40px 60px ;
                  ↓ ↓ ↓ ↓
                  Top Right Bottom left

* Margin and padding.

* Margin : Give spacing outside the border or outside the box.



Example :

Margin : 10px 20 px ;

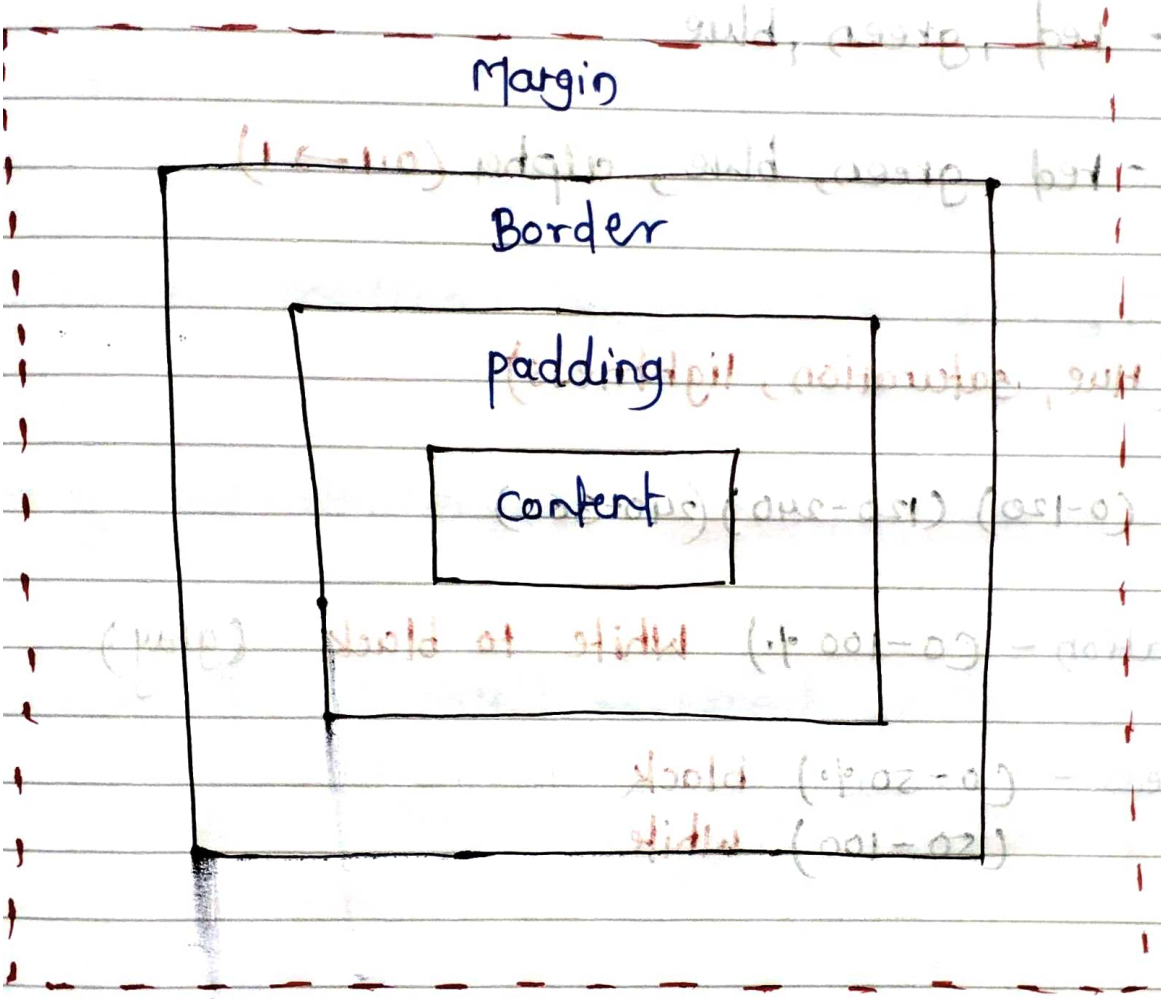
Top, Bottom left, Right

Margin : 10px 20px 30px 40px ;

Top Right Bottom left

padding. Give spacing Inside the border or Inside the box.

Example:-



Example:

```
padding : 10px 20px;
```

Top-Bottom left-Right

```
padding : 10px 20px 30px 40px;
```

Top Right Bottom left

* Color ways.

1) By color name - e.g. 'red'

2) Using hex code - e.g. #FFFF

3) rgb - red, green, blue

4) rgba - red, green, blue, alpha (0.1 → 1)

* Hsl (Hue, saturation, lightness)

  * Hue - (0-120) (120-240) (240-360)

  * saturation - (0-100 %) White to black (gray)

  * Lightness - (0-50 %) black
    (50-100) white

* set background Image

background-image : Url ("path")

background-repeat : no-repeat , repeat , repeat-x , repeat-y.

background-size : (100 %) (100 %)
(x-axis) (y-axis)

background-position : Top left , Top center , Top right
: bottom left , bottom center , bottom right

background-attachement : Fixed , scroll default

* background : url('path') no-repeat center center /cover ;

- url('path') → Url (path of Image)
- no-repeat → repeat property
- center → position of x axis
- center /cover → position of y axis

* Font property

* text-decoration : none
  : underline
  : line-through (same as delete tag)
  : overline (line at top)

* text-decoration-color : (color Name)

* text-decoration-style : dotted.

* text-transform : Uppercase
  : lowercase
  : capitalize

* text-align : center
  : justify (Inline spacing)
  : left, right

* text-align-last :

  Last line of paragraph spacing.

* text-align-indent :

  First Line of paragraph spacing.

* word-spacing : 10px

* line-height : 1.0 → 3.5

Give spacing between two lines.

* Units

* 96px - 1 inch

* em - Give font size relative to the parent element or font size multiple of their actual font size

* rem - Give font size relative to the document root elements or font size according to the root element

* % - font size according to the actual font size

* Height and width.

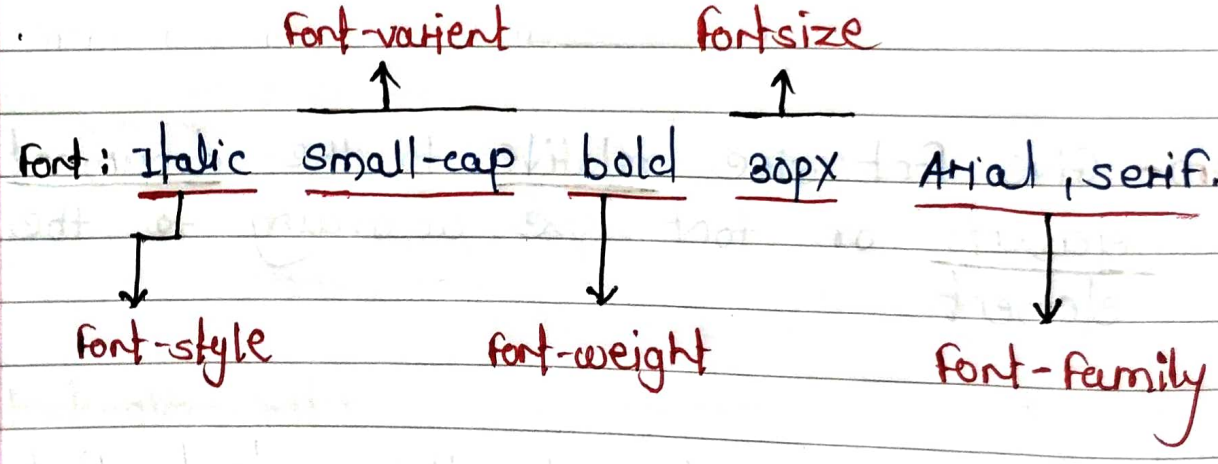
* vh - view port height

* vw - view port width.

* Web Fonts

  * Font-varient : small-caps.

Example:



* shadow effects in css

* text-shadow : x-axis y-axis blur color
: 1px 1px 10px orange

* box-shadow : x-axis y-axis blur spread color
: 10px 20px 10px 10px red.

* Gradient mixture of color.

* linear-Gradient start vertically (from Top side)

Example : background-image : linear-gradient (to left gold)

background-image : linear-gradient (to left red blue pink)

direction, color-stop1, color-stop2, 3

* radial-gradient start from center
(position - default is center)

Example: 1

background-image: radial-gradient (red, green, blue)

color 1 color 2 color 3

Example: 2

background-image: radial-gradient (farthest-corner at 60% 5
blue, yellow, green, black)

* background-image: radial-gradient (shape size at position
* background-image: radial-gradient (circle red green)

* shapes - ellipse (default)
- circle

* size → farthest-corner (default)
→ closest-side
→ closest-corner
→ farthest-side

* Conic-Gradient - spread equally to all sides - by default 0 degree to center.

Example:

background-image: conic-gradient (red, yellow, green)

color 1, color 2, color 3

Example 2:

background-image: conic-gradient (red 45deg, yellow 90deg, green 210deg)

color and degree for each color

* Overflow.

Adjust the extra content which goes outside the box.

overflow : auto , overflow : hidden , overflow : scroll

overflow-x : auto , overflow-x : hidden , overflow-x : sc

overflow-y : auto , overflow-y : hidden , overflow-y : scrol

* Display and visiblity.

Display : none

Removes its orignal space as it is

Visibility : hidden

Remains space as it is either just hidden the box

* Positioning in css.

1) static - default

2) relative - Box move from its orignal space.

3) absolute - move from its parent element either according to its window's size.

4) fixed - fixed to screen according to the window

5) stickey - scroll from something and then fixed according to the window.

* Top, bottom, left, right.

Example:

```
div.relative{
        position : relative, absolute, fixed, sticky
        top: 80px;
        right: 0px;
        width: 200px
        height: 100px

        }
```

* Flexbox (Flexible box)

- Used to give flexible length to the flexible items.
- used in Grid system.
- Helps to align items in a container
- Works two dimensional - either row or columns wise
- Always work in parent box
- Direction - horizontal (x-axis)

* justify-content : center, flex-end, flex-start, space-around space-evenly, space-between.

* flex-grow : To increase specific box size in terms multiple.

* flex-grow : 1 ; --- used in child.

* order : change the order and shifted towards last

* order : 1

* Flex-direction : row, column, row-reverse;

Flex-direction : row.

combination of Flex-direction and Flex-wrap.

* Flex-wrap for responsive layout for mobile size.

Flex-wrap : wrap;

Flex-Flow : row, wrap-reverse, wrap;

* text-align-last : Helps to overlapping other element.

z-index : 1 ---- (0-1, 0-999)

* transition - exicute the css property smoothly over specific duration.

* transition-property : background-color;

* transition-duration : 0.5s;

* transition-delay : 3s;

* transition-timing-function : ease, ease-in, case-in-out, ease-out.

* ease → start slowly fast end slowly.
* ease-in → start slowly.
* ease-out → End slowly.
* ease-in-out → start slowly End slowly.

short Hand property.

* transition : all 8s ease-in
* transition : width 2s ; height 4s;
* ~~transition~~
transition : width 2s , height 2s , transform 2s ;

* Transition transform.

It is used to apply 2d or 8d effects.

zoom in zoom out effect

* >1 (scale to zoom in)
* <1 (scale to zom out)
* 1 (scale is stable)

Example:

transform : scale(0.5) --- zoom out

transform : scale(1.3) --- zoom in

transform : scaleX(1.3) --- zoom in X axis

transform : scaleY(1.3) --- zoom Y axis

* transform - translate Move from its orignal space.

Example :

transform : translate (120px 300px)

transform : translateX(300px)

transform : translateY(300px)

* skew - tilted angle wise (deg)

transform : skew (30 deg) ;

transform : skewX (70 deg) ;

transform : skewY (10 deg) ;

* 2D transform methods

transform : translate (100px)

transform : rotate (90 deg) --- 3d transform

transform : scale (1.1) --- zoom out

transform : scale (0.5) --- zoom in

transform : skew (30 deg)

transform : matrix (scaleX(), skewY(), skewX(), scaleY()
translateX(), translateY())

* Animation Helps to build animated effect.

* animation-name : demo ; --- animation Name

* animation-duration : 5s ; -- time to complete 1 rotation

* animation-delay : 5s ; --- delay to start

* animation-iteration-count : infinite ; -- animation exicuted infinite times.

* short Hand property.

animation-iteration-count

* animation : demo 5s infinite ;

animation-name animation-duration

(0-100% partition)

@keyframes animation-name {

10% {

}

}

* Media Query.

Helps to build responsive UI for various devices or media types.

Example:
Mobile, dekstop, tablet, laptop, larger screen

* Media types.

screen - Only for laptop, mobile, dekstop screen.

All - All media types.

printer - print devices.

Examples:

@media screen And (max-width: 667 px)
work between 0 - 667px.

OR

@media (max-width: 667px)
{
styling for mobile devices.
}

* for tablet screen

```
@media screen And (min-width: 668px) And (max-width: 927px)
{
    tablet (working > 668px)
}
```

* For laptop devices.

```
@media screen And(min-width: 927px)
{
    work - > 927 px
}
```

* for Larger screen

```
@media screen And(min-width : 1200)
{
    work > 927px to 1199px
}
```