

• Fire Fox is Best to study CSS

INP Points Related to CSS

- * DOM - Document Object Model. It is constructed as a tree of objects.
- * Id and Class ← Identifiers uniquely represents all nodes of DOM. We cannot have same id for different nodes.

Types of adding CSS

- 1) Inline
- 2) Internal
- 3) External

Inline: we add CSS in HTML using inline CSS.

```
<div style = "background-color: red;"> </div>
```

Internal: we add css in HTML file itself

```
<style>
```

⋮
: Add CSS

```
</style>
```

← This will work
in any part of HTML

External: we can link external CSS file to HTML

```
<link ref="stylesheet" href="location">
```

Selector in CSS

A CSS selector is used to select element from HTML to style

body { }
↑
element Selector

h2 { }
↑
element

#id { }
↑
id selector

.className { }
↑
class selector

$h_1, h_2, h_3 \in \{3\}$

$h_2, \cdot \text{class}, \# \text{id} \in \{3\}$

group
Selector

$P. \text{red} \in \{3\}$



Select element of class

* $\in \{3\}$

↑ universal selector is used to select all elements.

* Inline will override internal and external style.

* We can add multiple class in one element as well.

Colors in CSS

- * We can use color in CSS to color elements in CSS.

Types of color value

- * RGB : rgb (redcolor, bluecolor, greencolor).
- * HEX code : #fffffff
- * HSL : Hue, Saturation, Lightness
eg: hsl (8, 90%, 63%)

Background - Image

{

background-image = url ('index.jpg');

3

There are few properties in CSS to stop repeating of images

background-repeat = x; 2 step x

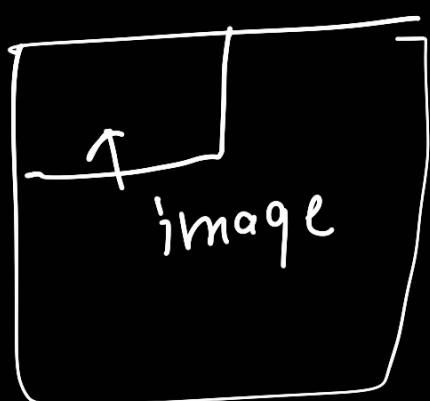
background-repeat = yes } repeat

background-repeat = no; } no repeat

Background Size Property

cover: fits & no empty space
remains

contain: fit & image is fully
visible



auto: image is originally displayed

{ { width } { height } }: sets width
& height

Background - position:

top center

bottom center

Background attachment Property

Background - attachment : fixed

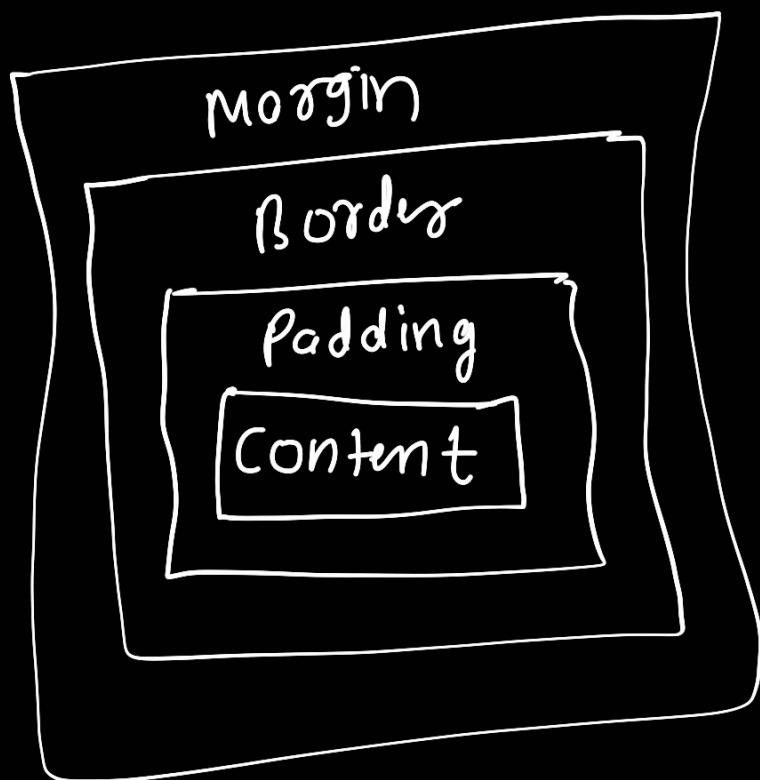
(Image will fixed in background even we will scroll)

there are more property

Background shorthand

Basically write everything in one line.

CSS Box Model



M - Margin
B - Border
P - Padding
C - Content

Total height = height + top/bottom padding
+ top/bottom border
+ top/bottom margin

Margin

.box {

top right bottom left

margin: 2px;

padding: 3px;

}

top
right
left
bottom
right
left
top
bottom

* margin : X X X X Left
top right bottom

* margin : X X
top/bottom left/right

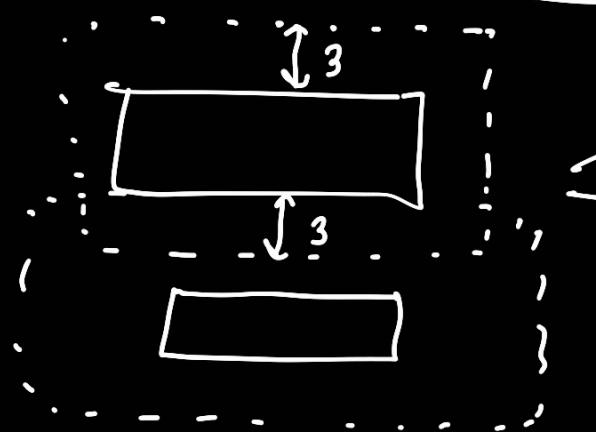
* we can set individually also :
margin-top, margin-bottom .

Border:

Border : Value solid color

Border-radius → Gives rounded corners

Collapse-Margin:



If we mention margin for both div the margin would be made collapse by default .

The majority of both margin is taken and merged with small margin.

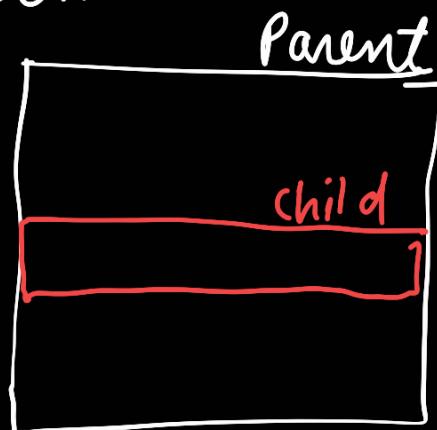
Box Sizing :

Defines the way in which we can include Padding and border with height and width of Box/elements.

There two types of Box-Sizing

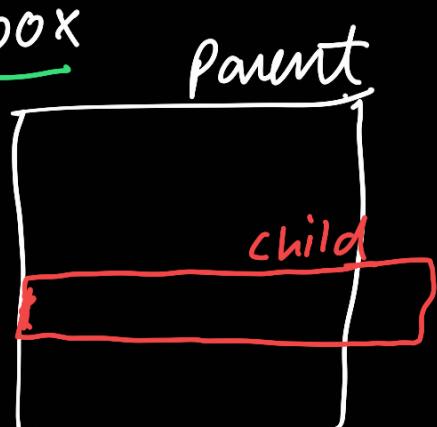
box-sizing: borderbox

In this the child will fit completely in parent adjusting Padding and borders.



box-sizing: contentbox

Parent will let child Padding and border take.



this is default by CSS

Fonts and display

Display Property

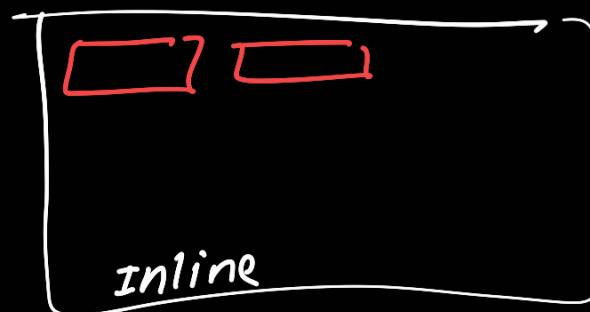
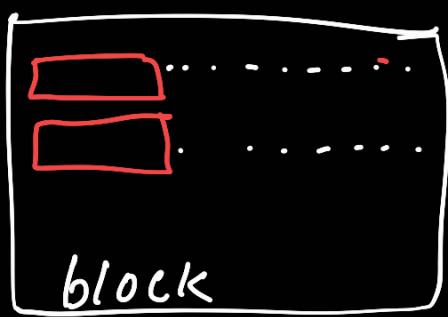
Determine whether element will be inline / block element. It is also used to determine flexbox / grid etc. By default display is block.
margin

`display: inline`

Takes only required space. No line break before and after. Setting width and height is not allowed.

`display: block`

Takes complete block / width of element and new element is added to new line.



`display: inline-block`

some as inline only change is we can adjust height / width as well as padding is allowed.

`display: none`

remove element from dom.

text-align Property

Use to set alignment of text inside container.

text-decoration property

Use to decorate text

can be outline, underline, line-through.
none

text-transformation property

(uppercase / lowercase)

Capitalize: only starting word
uppercase

uppercase : All uppercase

lowercase: All lowercase

line-height Property

Describe height between lines.

Font

Font-family font-style

Font-weight font-variant

Font-size

font-family: 'font1', font2, font3

If font1 is not render then font2
will be render If not font2 then
render font3

Size, Position & List

1. em → unit relative to the parent font size.

2. rem → unit related to the root font size (<html>)
3. vw → unit relative to 1% viewport width
4. vh → unit relative to 1% viewport height
5. % → unit relative to the parent element.

min/max - height/width property

This property will override Normal height and width property in the given range.

Position

static: Properties top/bottom/left/right
Default has no effect. This will follow flow of HTML.

Relative: Properties top/bottom/left/right will work. Otherwise it will work mostly as static.

Absolute: The element is removed from the flow & is relative to its first non-static ancestor- top/bottom etc works.

Fixed: Positioning will happen according to browser window.

sticky: Element will stick to its position and will be visible even when we are scrolling.

Z-index: Works like stack

List-style Property

The list style property is a shorthand for type, position & image.

UL {

list-style-position

list-style: square inside url('x.y.jpg')

↓
List style type

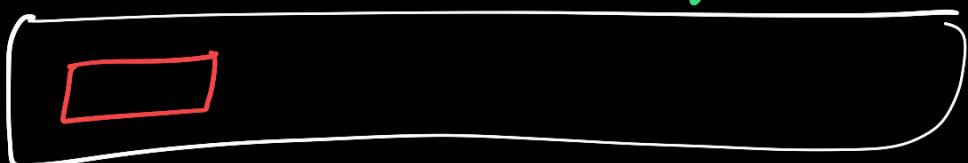
↑
List-style
Image

Flex box

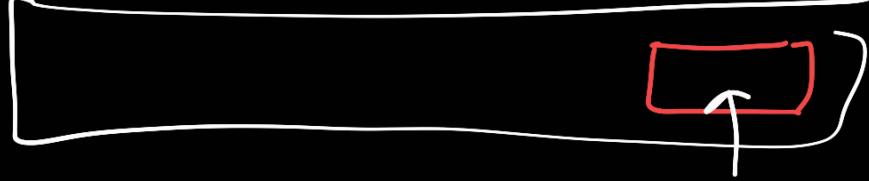
The float property

float: left / right

Before



After



float: right

The clear property

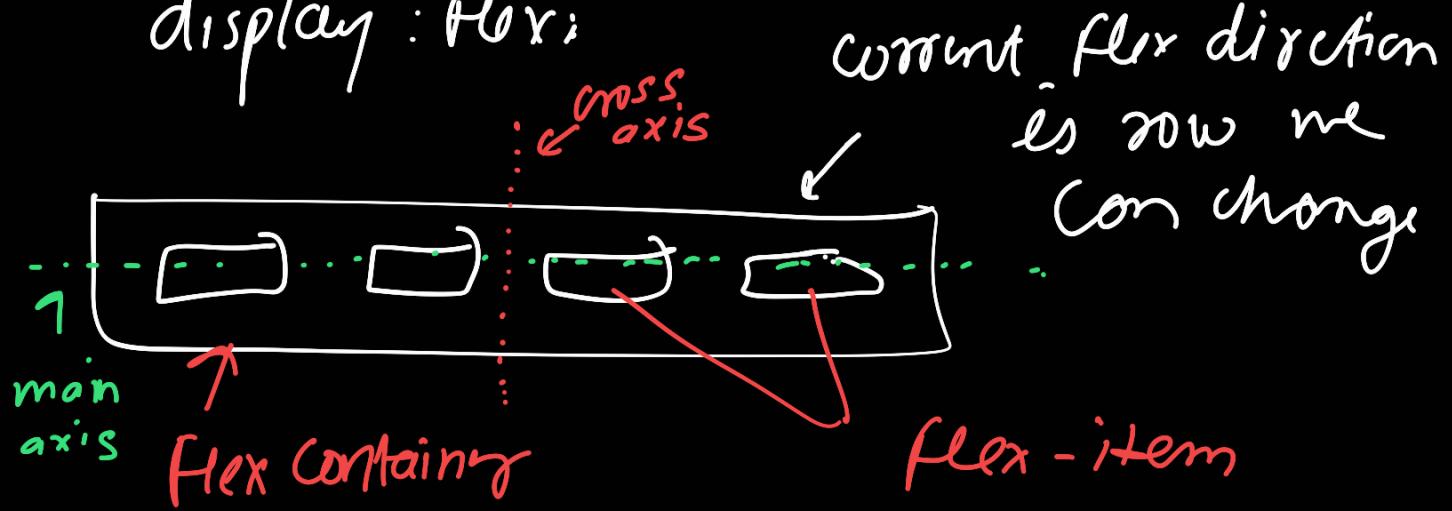
clear: left / right

(Mera left or Right ki Space khalli)

(karo) Order matters karta है।

The flexbox property

display : flex;



Flex-direction: row / column;

(render reverse) row-reverse / column-reverse;

Flex-wrap: wrap

(adjust render)

: justify-content: center; (horizontal center)

: align-items: flex-end;

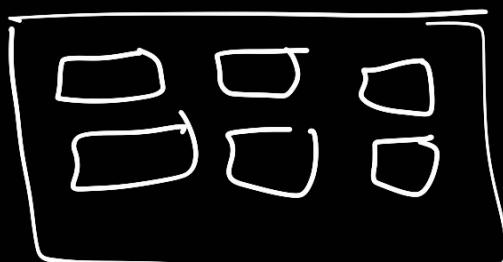
Align-content: center.

Properties for elements in flexbox

- 1> order: Control the order in which the items appear in the flex container.
- 2> align-self: Allow default alignment to be overridden for the individual item.
- 3> flex-grow: Defines the ability to grow flex item.
- 4> flex-shrink: Specifies how much a flex item will shrink relative to the rest of the flex items.

Grid in CSS

display: grid;



Properties

grid-row-gap: 1px; }
grid-column-gap: 1px; } We can
combine both from

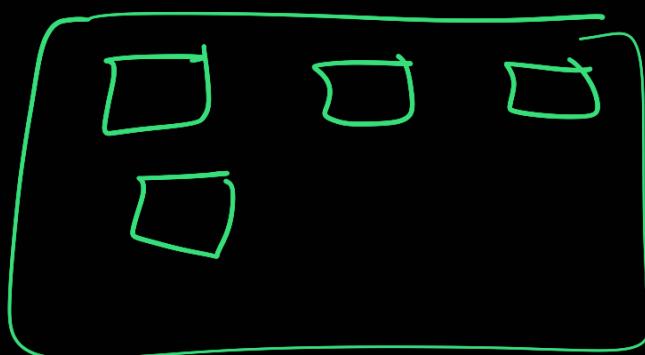
grid-gap: 9px 9px
↑ ↑
row column

Alternate

grid-gap: 9px; ←

for both
row and
column

grid-template-columns: auto auto auto;



This represents
3 columns

justify-content (align whole grid)

align-items (vertically align grid items)

Grid item property

grid-row: 1/3; (start from line
grid-column: 1/3; no line 3)

CSS Media Query

```
@media only screen and (max-width: 800px) {  
    body {  
        background: red;  
    }  
}
```

Transforms, Transition & Animations

(Rotate element, skew element)

