Turwash Notes

Date :- 12th Jan 2023

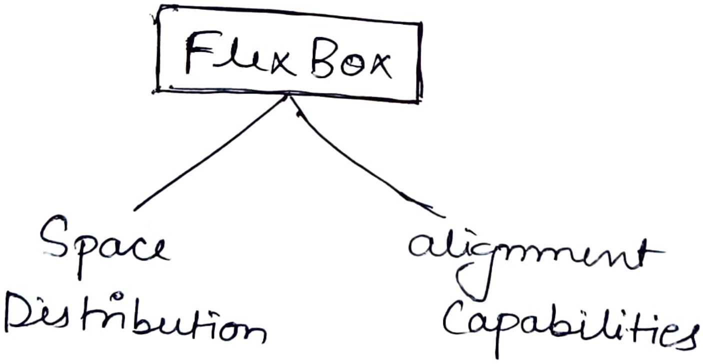
CSS - BASICS - III

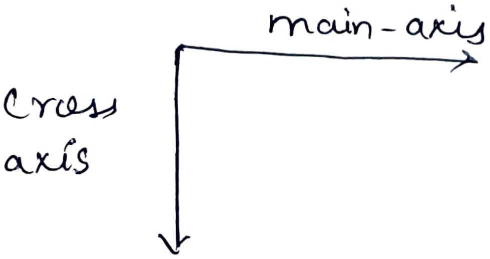
FlexBox :-

way to get more Flexibility in your layouts & to Simplify responsive layout design.

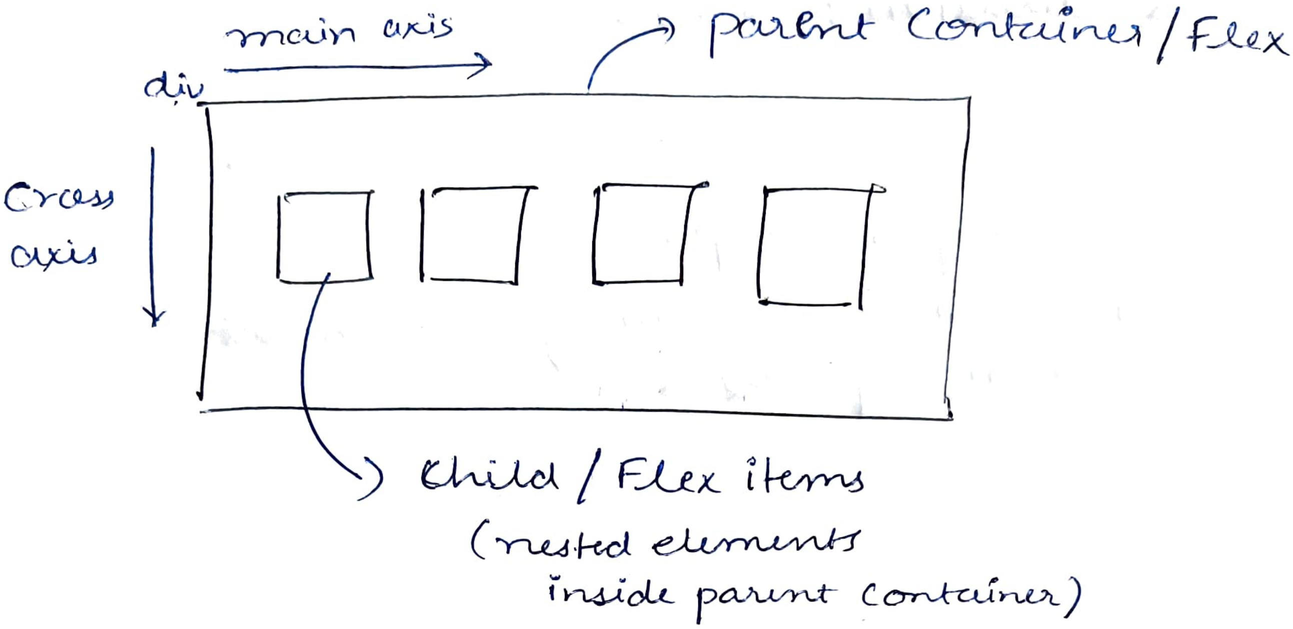
→ It makes it easy to align elements on a 2D plane.

→ It is a 1D layout model where we can do Space distribution b/w elements, with powerful alignment Capabilities.





parent- child relation in Flexbox



positioning tricks

1) Block
2) Inline
3) CSS position property
4) table
5) Flex Box (Imp)

———X———X———

For making Flex.

CSS → display : Flex;

By default it changes horizontal to vertical.

But for manual change use.

Flex-direction : column; (vertical)
Flex - direction : row; (horizontal)

For reverse, we use :-

Flex-direction : Column-reverse;
Flex-direction : row - reverse.

———————— x ———————— x ————————

✓ Flex -wrap : wrap;

this makes the width of div intact. when Squeezed.

another values { default → nowrap ✓
also → wrap-reverse; ✓

Flex-container properties :-

1) Flex-direction
   - row
   - row-reverse
   - column
   - column-reverse
2) Flex - wrap
   - wrap
   - no-wrap
   - wrap-reverse.
3) Flex - flow
4) Justify - content
5) align - items
6) Align - content

~~Flex - Item properties~~ :-

✓ Flex-container property is used for parent element

✓ flex - item property is used for nested / Child Flex items.

Flex container properties :-

1) Flex-direction

- Flex-direction : raw ;
- Flex-direction : raw-reverse ;
- Flex=direction : column ;
- Flex-direction : column-reverse ;

2) Flex-wrap

- Flex-wrap : wrap
- Flex-wrap : no-wrap
- Flex-wrap : wrap-reverse.

3) Flex-Flaw (Shorthand notation)

↳ we can add Flex direction & Flex wrap in one line.

Flex-flaw : raw wrap

4) Justify content

↳ align content in main axis

- justify-content : flex-start (left end)
- justify-content : flex-end (Right end)
- justify-content : center
- justify content : space-around ; (space b/w items equal spacing) ← no space equal from left & right
- justify content : space-evenly (equal spacing)

5) **Align-items** :- (alignment in cross axis) vertically.

- align-items: Flex-Start,
- align-items: Flex-end;
- align-items: center;
- ✓ align-items: Stretch; (default)
- align-items: baseline;

——— x ——— x ———

✓ Gapping b/w elements / boxes.
use.

gap : 10px;

gap → Shorthand notation
↳ row-gap
↳ column gap.

——— x ——— x ———

6) **Align-content** :-

↳ handeling Spacing b/w row of boxes.

- align-content : Flex-Start,
- align-content : Flex-end;
- align-content : center.
- align-content : Space-between;
- align-content : Space-around;
- align-content : Space-evenly:

Flex - item properties :-

1) Order

2) Flex - grow → equal width ✓ & giving Spaces to element ✓

3) Flex - shrink → Shrink Speed (Squeezing)

4) Flex - basis → used for giving width.

5) Flex → all above property in Single line

6) Align-self.

1) Order :-

put in order with values.

```
#box1 {

      order : 3;

   }
```

2) Flex-grow

Flex-grow : 1;

This property is for giving space to element.

3) ~~Flow~~ Flex - Shrink (by default - 1)

Shrink Speed when squeezed the layout.

4) Flex-basis :-

```
Flex-basis: 100px;
```

we can also use %

a particular item will take the % width from total ✓

Width	v/s Flex-basis
The overflow content gets hidden	the width increases with content.

Flex-basis makes it Responsive

5) Flex :-

use to write all properties in one single line.

(order, Flex-grow, Flex shrink & Flex-basis)

↓

all in one

```
Flex : 3 4 2 120px;
```

6) Align-self :-

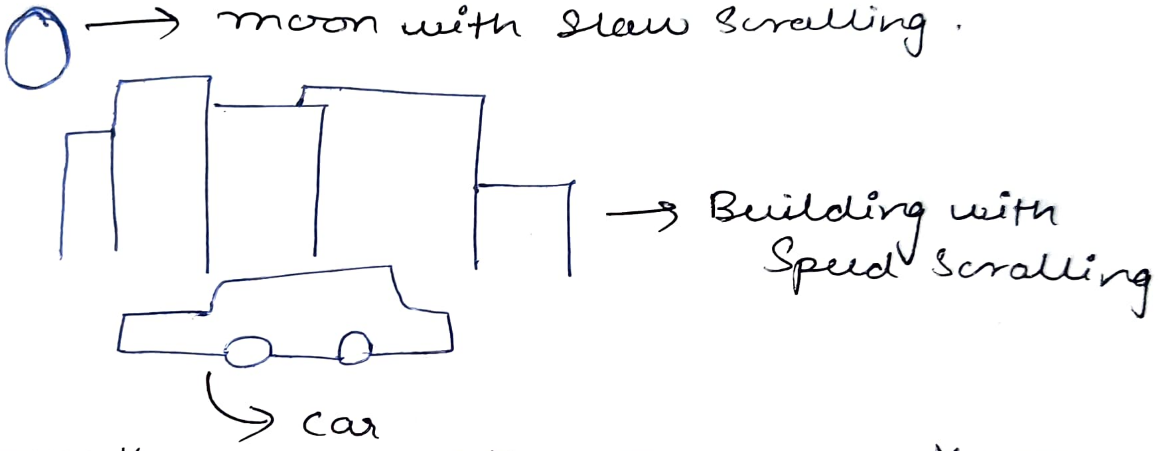
Self aligning ✓

```
align-self : Stretch;
             Flex-end;
             ~~Flex-end~~
             Flex-Start;
             Center;
```

For each & every element;

Parallex - website :-

parallex Effect?
↓
In a Single Frame when Speed of different objects is different, that effect is parallex effect.



→ First we overlap our images

→ Then, Foreground image near,
Background image Far (3-D effect)

→ Then when we will Scroll
one will go fast & one
will move slow.