

absolute pos
↳ nearest pos ancestor

How to handle layering.
Use Z-index to decide

Konsi layer kidhr aage

H.W } .container
- z-index: -1; \Rightarrow req? yes or no? reason.

3D - Transforms

To work on Z-axis, try perspective on explicitly
it activates Z-axis

Inline element (like span) par translate nhi chalta

transform: translate(100px, 50px);

transform: rotate(-45deg); + \rightarrow clockwise
- \rightarrow anticlockwise

transform: scale(2.5, 3);
horizontal scaling vertical stretching
// shorthand notation

transform: scaleX(2);

transform: scaleY(3.5);

transform: skew(45deg); tilt
- - (-45deg)
- - (180deg)

transform: translateZ(-10px);

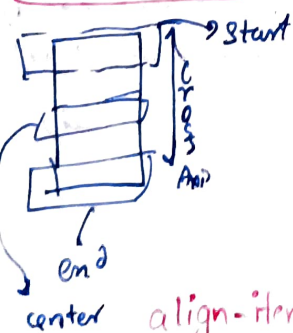
pic andar
screen ke
jayege,
dur jayege

flex-flow: (row), (wrap-reverse);
 ↓ ↓
 flex-
 dir flex-wrap

justify-content :

- flex-start
- center
- flex-end
- space-around
- space-between
- space-evenly
- baseline

align-items : aligns ^{flex-}items on CROSS AXIS (↓)



flex-direction: column

flex-start

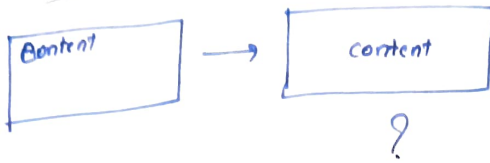
flex-center

flex-end

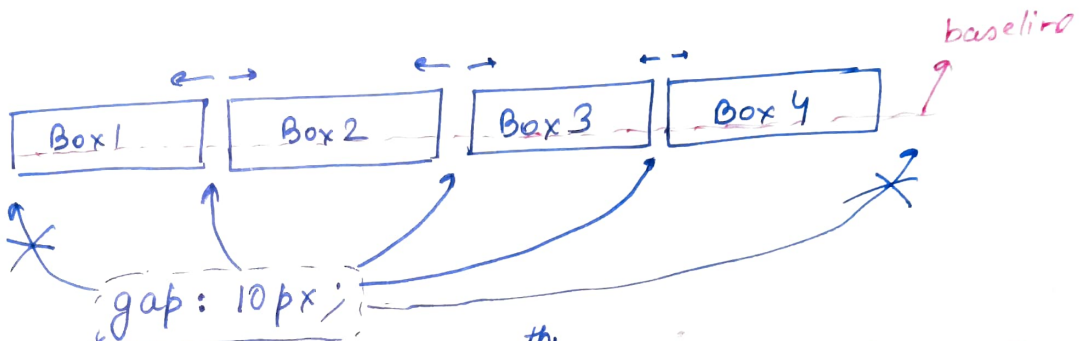
align-items

stretch : items pura stretch hoga & takes height of parent.
Hems r stretched 2 fit the container
It is By default value of Align Item.

align-items → *baseline* // all content comes on single/same horizontal line



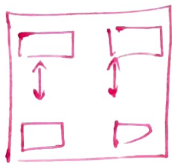
make
`justify-content: center;`
`align-items: center;`



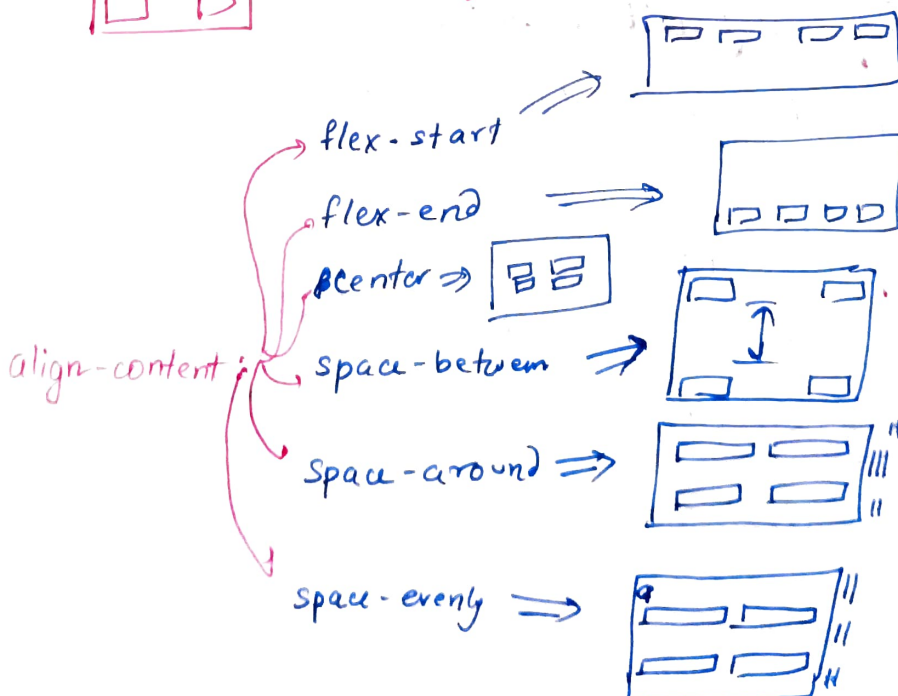
introduces gap b/w ^{the} boxes only. & not around.

shorthand notation : `row-gap: -px;`
`column-gap: -px;`

align-content : handle spacing between multiple rows



this space handled by align-content



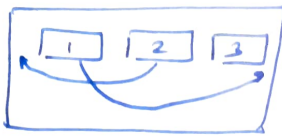
Flex-item-properties

order

box1

```
{ order: 9;
```

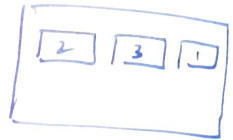
```
}
```



1 : order \Rightarrow 9

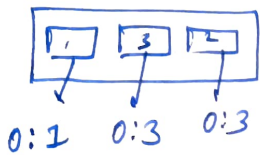
2 : order \Rightarrow 3

3 : order \Rightarrow 5



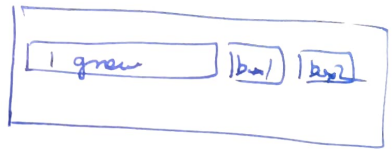
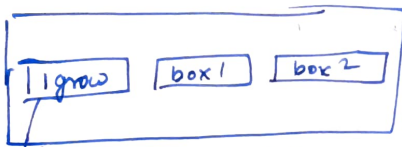
(order is calculated relatively kuch bhi no-de do)

For same order, next pos par place hoge



flex-grow : Default value: '0'

* -ve values invalid



flex-grow : 2

Available space is used to grow the flex-items.

flex-shrink : Default value: '1'

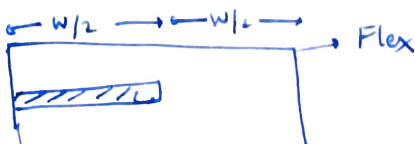
(set shrinking speed of a flex-item)

flex-basis : — px;

similar to width. but it provides responsiveness
overflow of content doesn't hide / trim.

(width se overflow content hides out)

flex-basis : 50%;



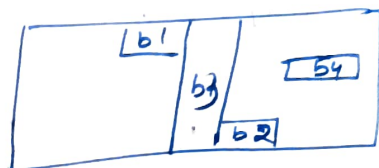
flex: shorthand notation to write in single line.

order,
flex-grow,
flex-shrink,
flex-basis

Eg: flex: 3 4 2 120px; // readability wise not very good

align-self: (aligns individual item)

#b3 { align-self: stretch; }
#b1 { align-self: flex-start; }
#b2 { align-self: flex-end; }
#b4 { align-self: center; }



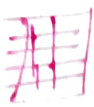
object-fit: cover; // fit image in container.
specifies how `` or `<video>` should be resized to fit its container.

transform-style: preserve-3d; // sets whether children of an element are positioned in 3D space or flattened in plane of element.

Eg: .container { transform-style: preserve-3d; }
container ke children mein ~~z-index~~ ^{z-index} set karo to add Layering & preserve-3D

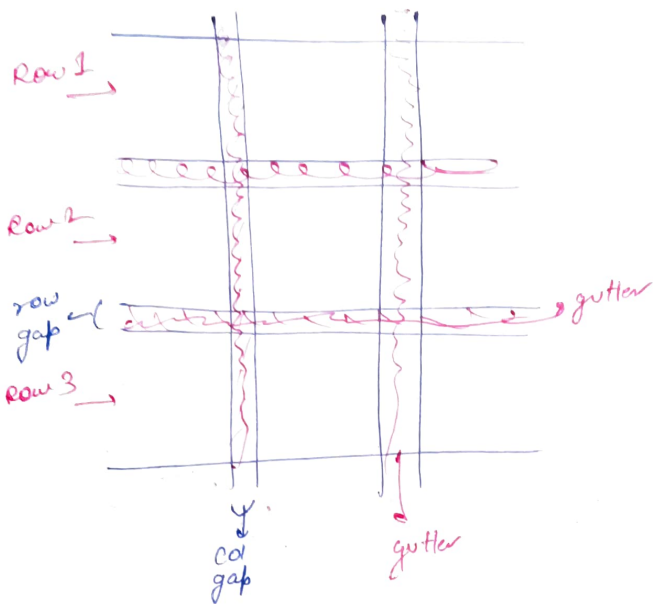
* background-attachment: fixed;
background img do not scrolls

* transform: translateZ(-40px) scale(3);
on z axis, goes 40px deep inside screen.
enlarges

Grid?  , arrows, marichandani, camera, grid lines? Excel sheet
Tiles, et.

CSS Grid? use 2D layout of rows & col using code.

* gaps b/w rows & col \Rightarrow gutter



grid market mein aya the, flex box hatane keliye, but ho nhi paya. duniya
hila denge

Adv. developer use (Grid + Flex box) : $\left\{ \begin{array}{l} \text{positioning \& alignment} \\ \text{\&} \\ \text{Layout dealt by Grid.} \end{array} \right.$ dealt by Flex box

Flexbox is how content flows.

Grid is simple layout

Flexbox is specific to positioning of content.
 Grid is specific to Layout Creation.

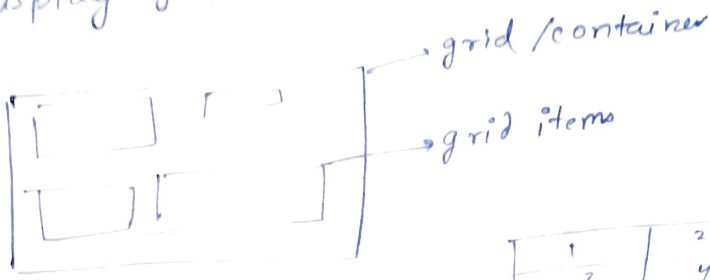
Flexbox offers 1D Layout. (\leftrightarrow & \updownarrow) (mainaxis & cross axis
at a time either of one direction)
 Grid offers 2D " ($\begin{smallmatrix} \leftrightarrow \\ \updownarrow \end{smallmatrix}$)

In flexbox : `display: flex;`

In CSS Grid : `" : grid,`

Flexible grids

display: grid; // now apply grid properties



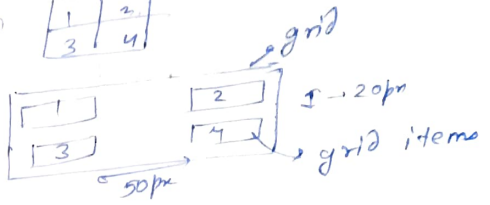
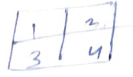
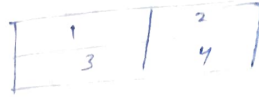
display: grid;

display: inline-grid;

column-gap: 50px;

row-gap: 20px;

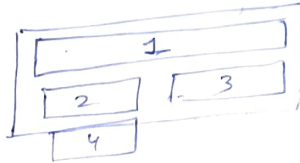
gap <= column-gap;
row-gap;



• item 1

{ grid-column-start: 1;
grid-column-end: 2;

? similarly, grid-row-start: 1;
grid-row-end: 2;



H.W
grid-row
grid-col

• item 2

grid-area: 2 / 1 / span 2 / span 3;

start on row 2

start on col 1

rowspan value = 2

colspan value = 3

grid-area

shorthand notation

grid-row-start;

grid-column-start;

grid-row-end;

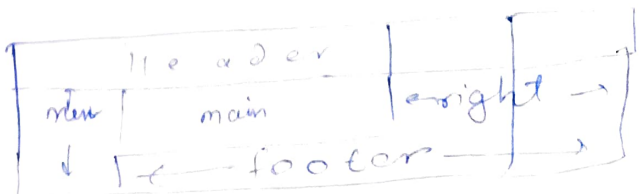
grid-column-end;

grid-template-areas:

'header header header ...'

'menu main main right right'

'menu footer footer footer footer'



grid-template-rows: repeat(3, 1fr);

fraction of available
space in grid container

Some more properties

⇒ justify-content:

- start
- end
- center
- space-between
- space-evenly
- space-around

⇒ align-content

⇒ justify-self

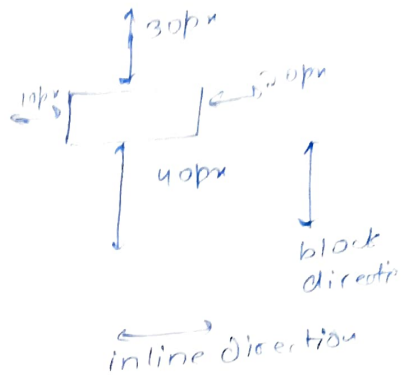
⇒ align-items

⇒ align-self

⇒ place-items

⇒ place-self

margin - inline - start : 10px
margin - inline - end : 20px
margin - block - start : 30px
margin - block - end : 40px



Observations from Image Gallery Project (mins)

- ⇒ Text @ centre → My Gallery
- ⇒ Bohut sari images hai, inke bich mein gaps hai.
- ⇒ Images are grayscale, b/w white dikh rhi hai. Hover → color changes
- ⇒ Text @ centre ⇒ text-align: center; can be used.
 ↳ h2 tag ✓ or h1 tag ✓
- ⇒ image ke upar text bhi aa rha hai, we can use figure, figurecaption.
- ⇒ abhi image grayscale hui padi hai ⇒ matlab filter lagai hai
- ⇒ on hover ⇒ shadow ban gyi ✓
 text appears on hovering ✓ (opacity: 1)
 " disappears on not " (opacity: 0)
 filter hat jata hai, black gradient top to bottom on image
- ⇒ gap b/w images ✓ } gap ✓
 or
 space-between ✓
- ⇒ same images are centred. (left-margin = right margin)
- ⇒ images are rounded. (border-radius ✓)
- ⇒ On hover ⇒ transition takes time.
 ⇒ image ka size ↑ ⇒ scale.
 ⇒ text ka size ↑ ⇒ scale
 ⇒ text overlaps ✓
 ⇒ font-family ✓
 ⇒ max-width ✓
 ⇒ wrapper ✓
 ⇒ favicon ✓
 ⇒ overflow x. hidden
 ⇒ Title ✓
- ⇒ grid ✓
 flexbox ✓ bot se bana sakte hai
- ⇒ bgcolor bhi set hua pada hai

HTML wise

heading h1 ✓

then

container ke andar bohot sari images
each image = figure tag + figcaption tag

bas ✓

Top level container @ centre,

margin: 0 auto;

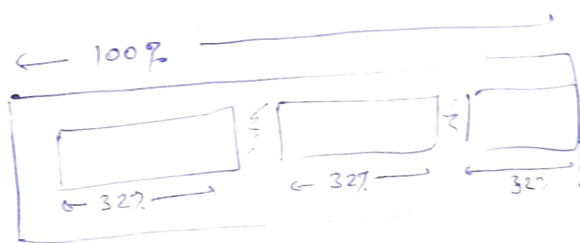
left margin = right margin

P: 20px; (already given)

Why box-sizing: border-box?

done bec when I add padding & border
in code, then content's size will ↓
but box ka size do not change

Box size remains same



remaining 4% used
in spacing

when we create flex items, its width are fit to content
but height part le leta hai.