CSS grid

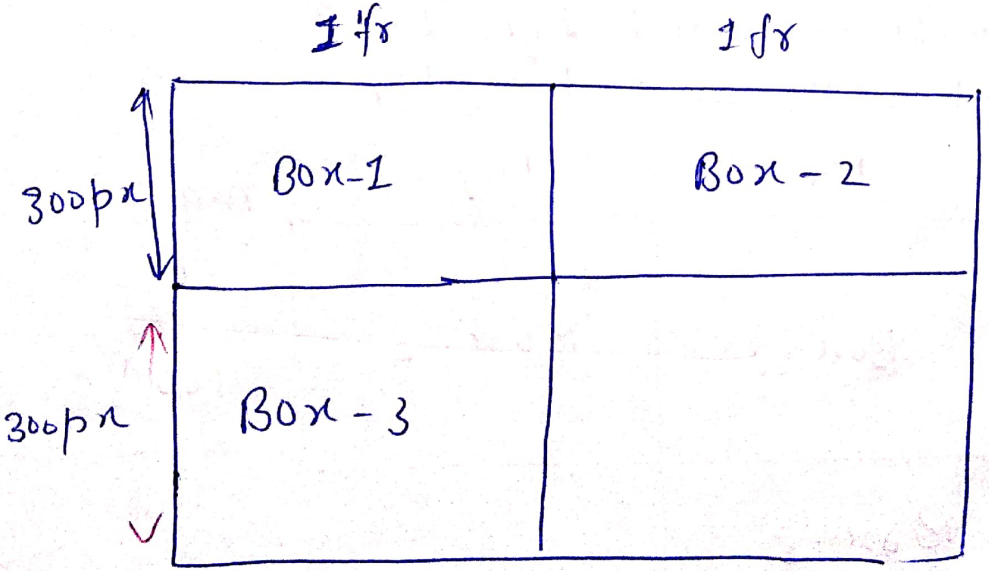
```
.container {
    display: grid;
    grid-template-columns : 1fr 1fr
    grid-template-rows    : 300px 300px
}
```

column-width

1st row-height

2nd row height

Let assum there are three boxes

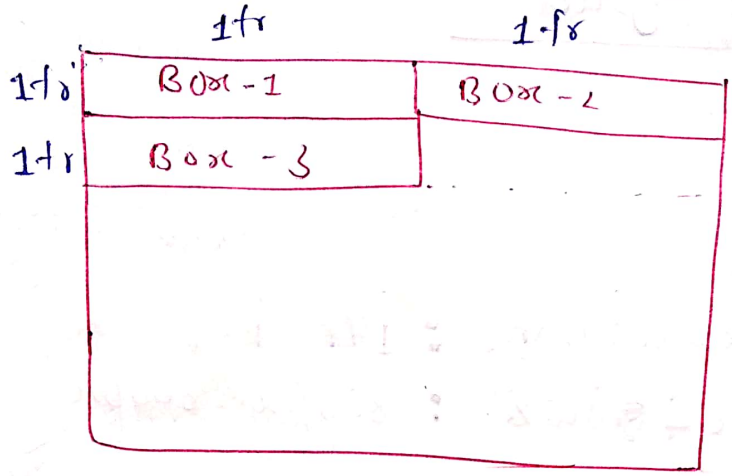


if all the fr are equal then it will take equal space that avairable in a singl row

#2

```
main {
    display : grid;
    grid-template-column : 1fr  1fr
    grid-template-rows :  1fr  1fr
}
```

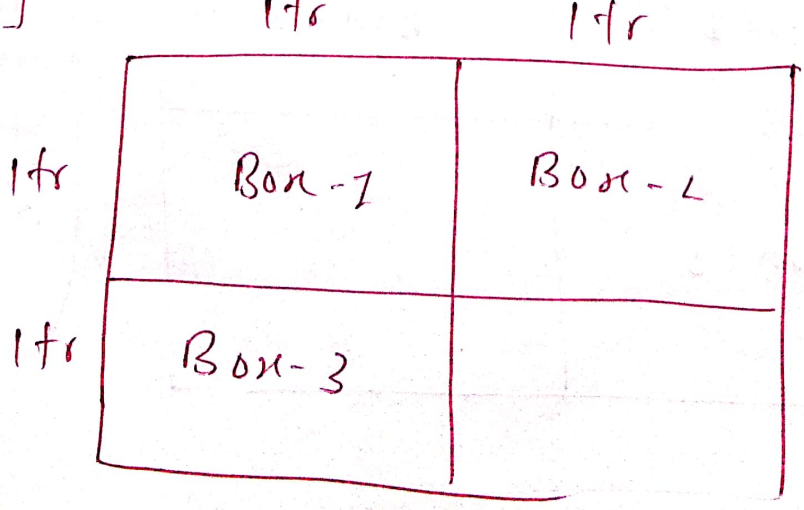
Let assume we have only three box in main



Because its column width and row width is in fr it only takes the available space for the content required inside the Boxes

But if we set the height for main container now

```
main {height: 100vh;
    display : grid;
    grid-template-column : 1fr  1fr
    grid-template-row :  1fr  1fr
}
```



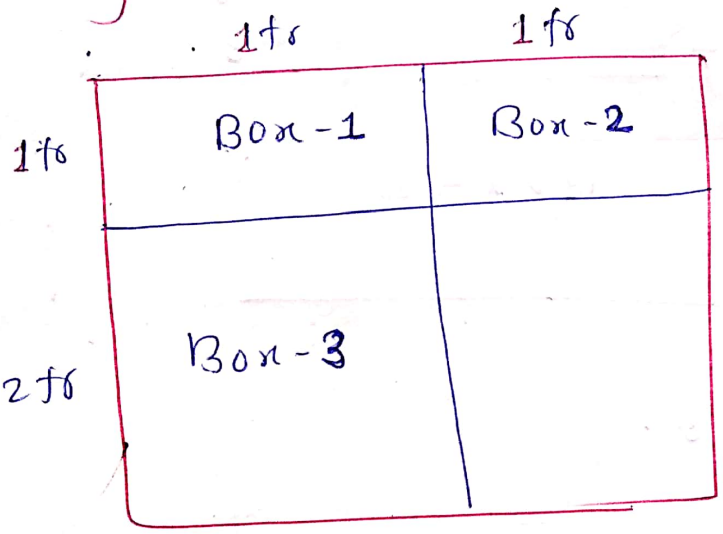
Grid-template (Shorthand)

~~grid-column-start~~

```
main {
    height : 100vh ;
    display : grid ;
    grid-template : 1fr 2fr / 1fr 1fr
}
```

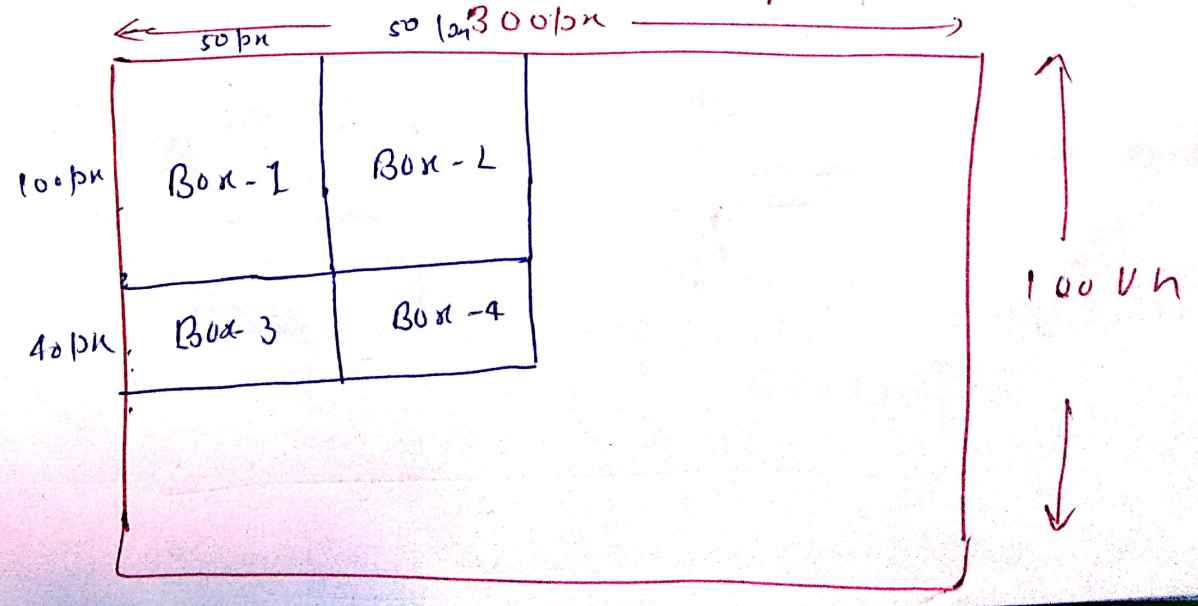
first-row 2nd-row 1st-column 2nd-column

Second-colum




```
main {
    height : 100vh ;
    display : grid ;
    grid-templates : 50px 50px / 100px 40px
}
```

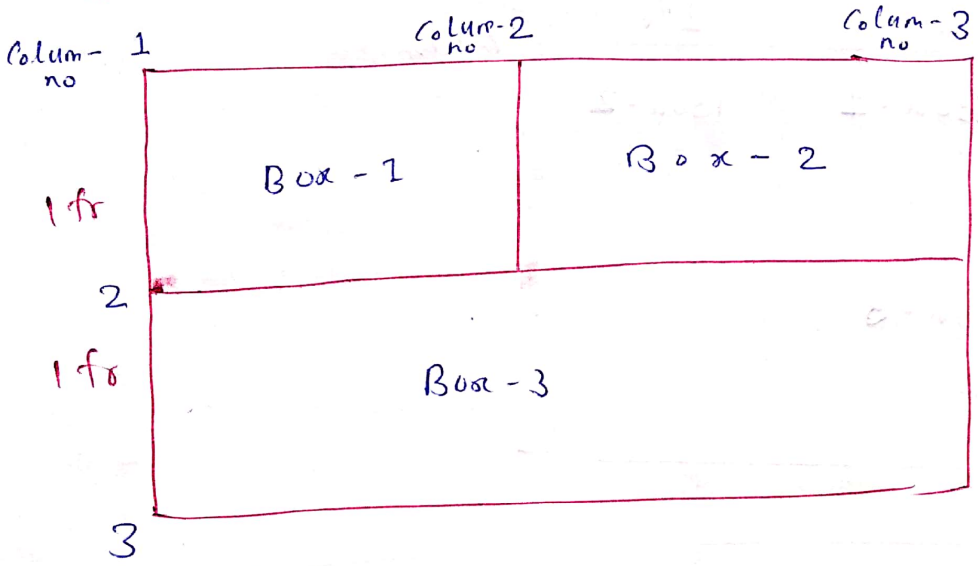
row



grid-column-start / grid-column #4

- It is property for item, to manage the start & end position of colum

```
main {
    height : 100vh;
    display : grid;
    grid-template : 1fr 1fr / 1fr 1fr
                    2 rows    2 colum

.box 3 {
    grid-colum-start : 1;
    grid-colum-end : 2;
}
}
```



Shorthand :

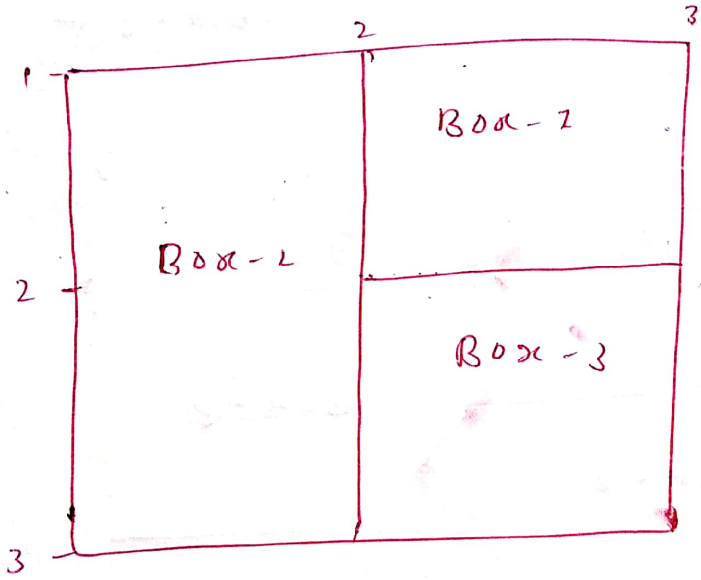
```
grid-colum : 1/3;
```

Grid-row-start | Grid-row #5
Grid-row-end

main{
height: 100vh;
display: grid;
grid-template: 1fr 1fr / 1fr 1fr
}

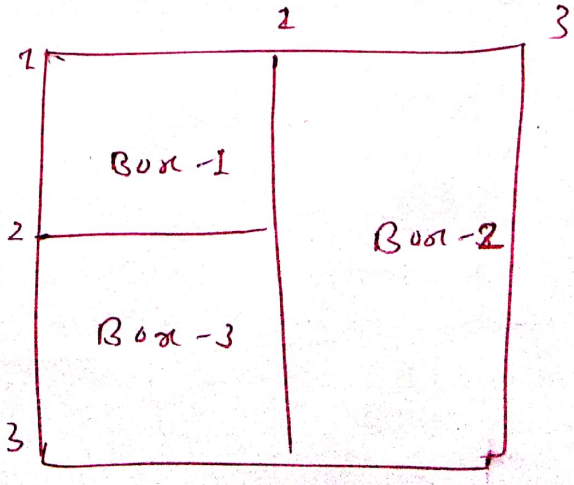
.Box-2 {
grid-row: 1/3
}

after applying this there will be some issue



to fix this we have to apply the property in box-1 & box-3 also

.Box-1{
grid-colum: 1/2
grow-row: 1/2
}

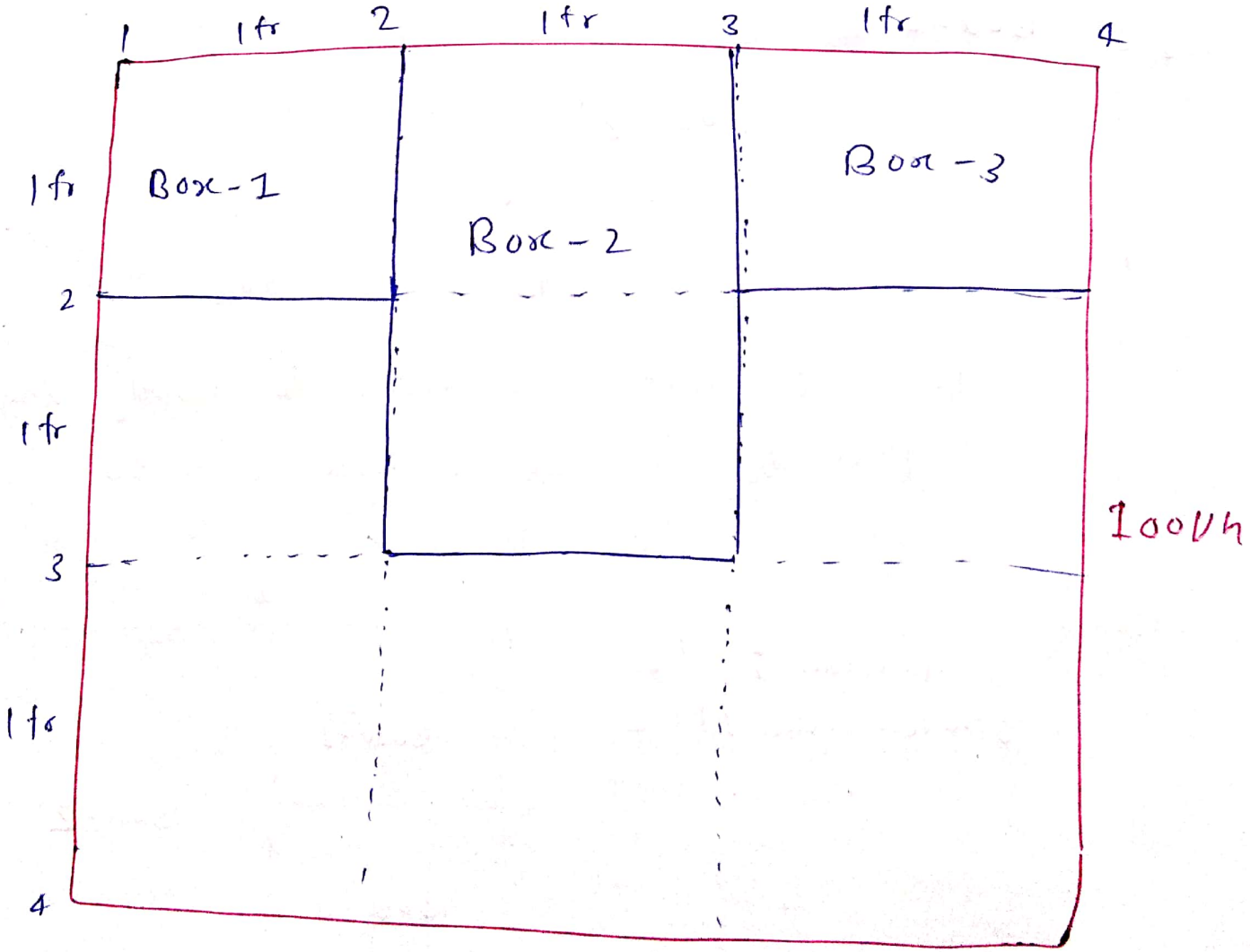


Another example for grid with 3 x 3 grid #6

```
main {
    height: 100vh;
    display: grid;
    grid-template-row: 1fr 1fr 1fr
    grid-template-column: 1fr 1fr 1fr
}
```

```
.box-2 {
    grid-row: 1/3
}

.box-1 {
    grid-column: 1/2
    grid-row: 1/2
}

.box-3 {

}
```

The layout will be

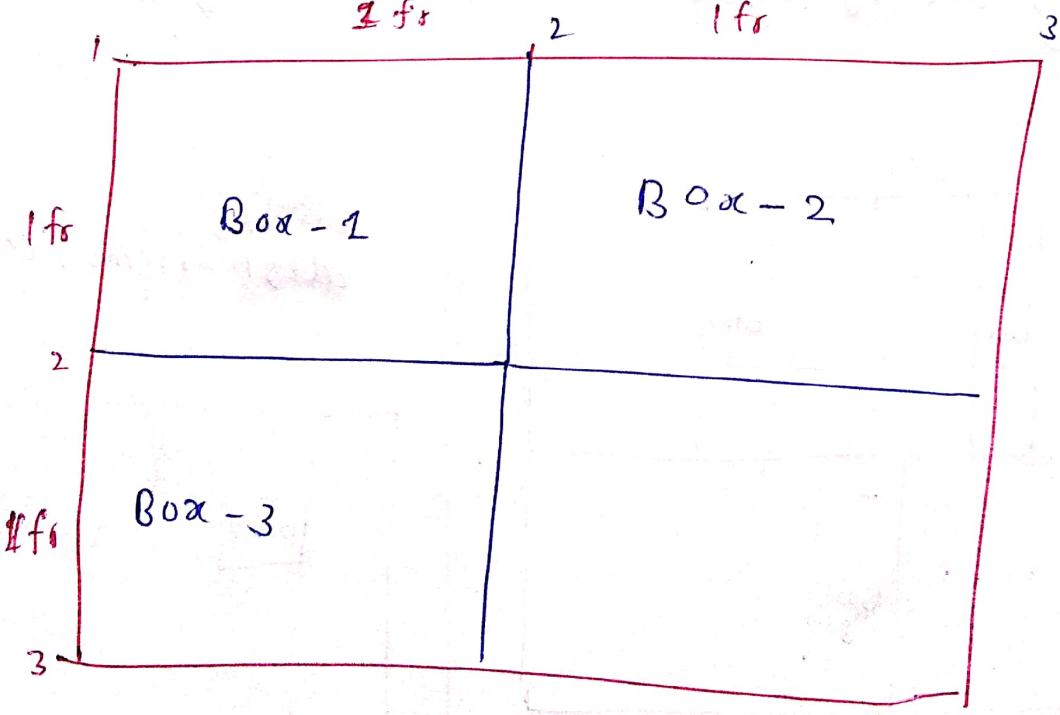


Grid-template-area. #7

```
main{
    hight: 100vh;
    display: grid;
    grid-template-colum: 1fr 1fr
    grid-template-row: 1fr 1fr

    grid-template-areas:
        " Box-1  box-2 "
        " box-3   .  " ;
}
```

To leave the empty space just put the .dot (.) in row*colum. box

```
.box-1 {
    grid-area: box1;
}

.box-2 {
    grid-area: box2;
}

.box-3 {
    grid-area: box3;
}
```



If you want box-3 from colum-no-2 to colum-no-3 just put the 2nd row & 2nd colum is box-3

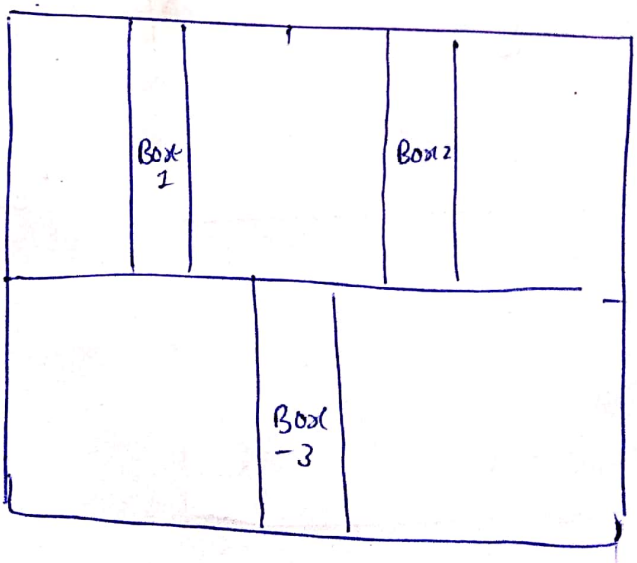
#8

- We can move element onside all the boxes like we can do in flex box, justify ~~content~~ etc
- But instead we use justify items for horizontal alignment.
- align-items for vertical alignment.

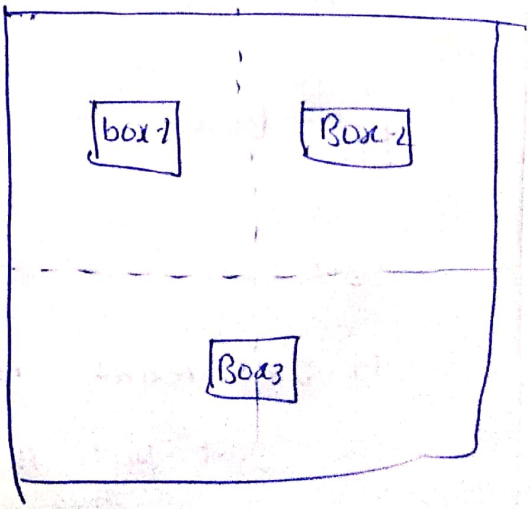
ex main {

display: grid
height: 100vh
grid-template-row = 1fr 1fr
" " -colum = 1fr 1fr
grid-template-area:
"box1 box2"
"box3 box3";
} justify-items: center

.box-1 {
grid-area: box1;
}
.box-2 {
grid-area: box2
}
.box-3 {
grid-area: box3;



~~justify~~
align-items: center;

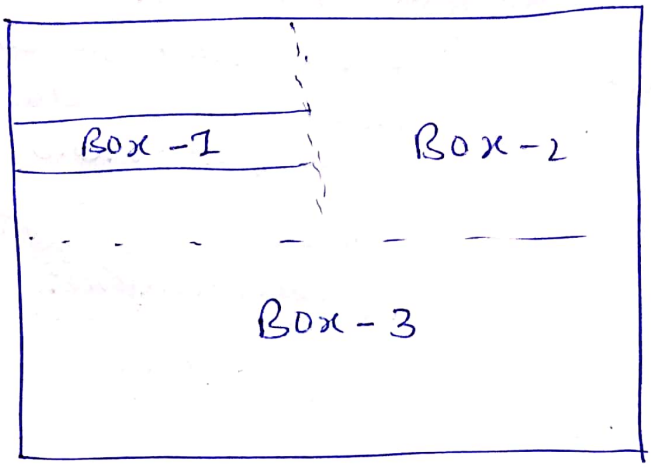


- **Align-self :** It is property of child items #9
  - we can align item by self alignment.
  - It align item vertically.

align-self : center

.Box-1{

Align-self: cent

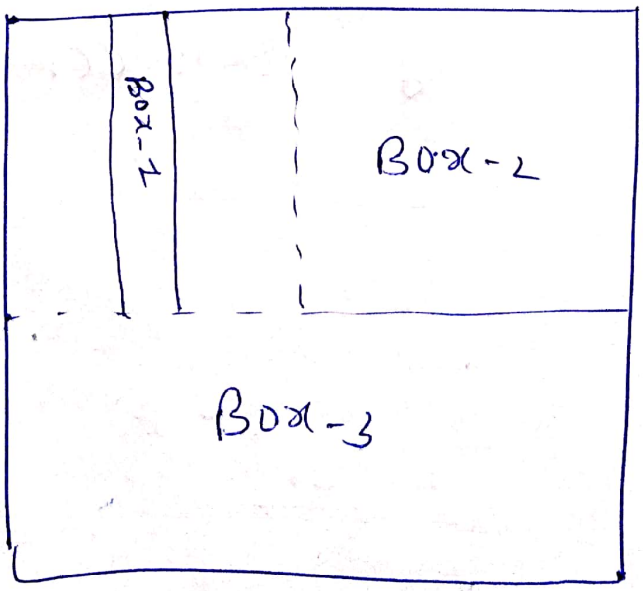


- **Justify-self:** It is also the property of child element.
  - It align item horizontaly.

.box-1{

Justify-self: center;

}



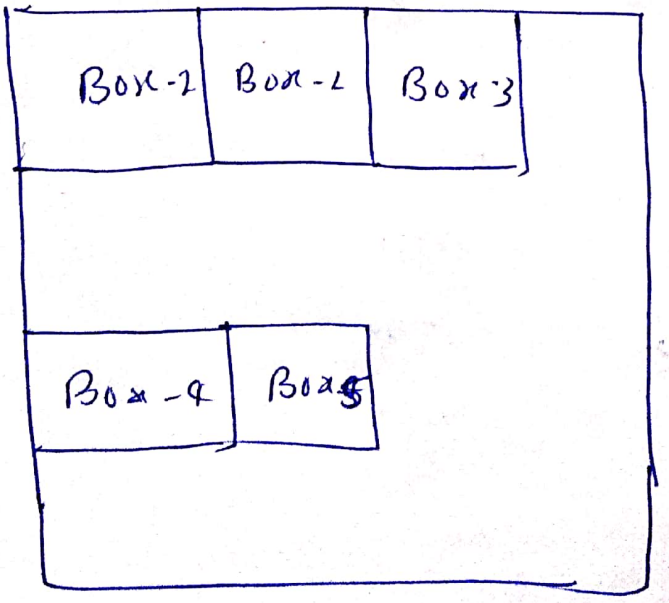
Flex-Box

#10

Align-Content :- We use align content when we use wrap (flex-wrap: wrap)
- It works as vertical alignment but whole content with wrap.

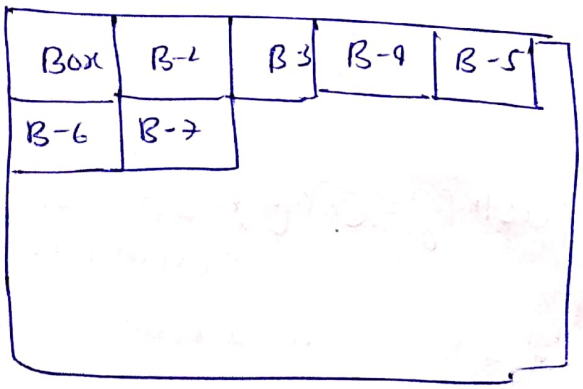
- flex wrap without align-content

```
.container {
    height: 100vh;
    display: flex;
    flex-wrap: wrap;
}

.box-1 {
    width: 200px;
    height: 200px;
}

.box-2 {
    Same as box-1
}

.box-3 { Same as above }
```



If we want to wrap the box without leaving the space after first row

use
align-content: flex-start;
in container



align-content: center;