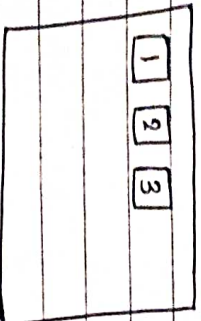


⇒ Flex Container Properties

* display: flex

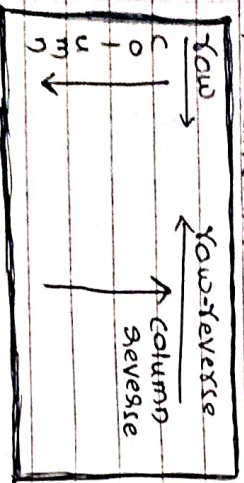
makes container as flex & items init as flex items

* flex items

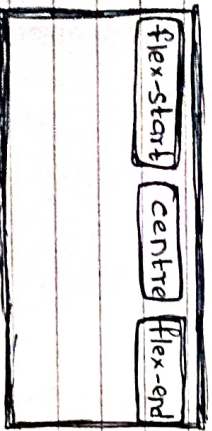


flex-direction: row
flex default property
top left.

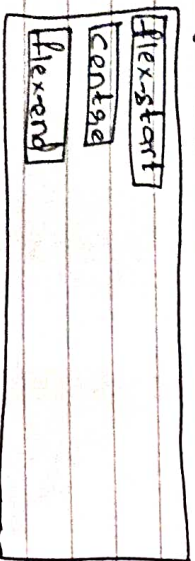
* flex-direction:



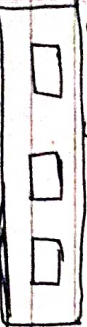
* justify-content:



* align-items: vertically centres



* justify-content:



space-between

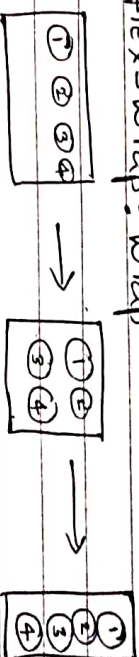


space-between

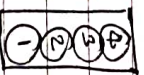
* flex-wrap:

by default flex-wrap: no-wrap
this makes website unresponsive when browser window shrinks.

flex-wrap: wrap



flex-wrap: wrap-reverse (opposite behaviour to above)

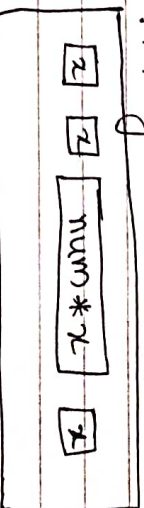


⇒ FlexItem Properties

* order: AnyBigNum

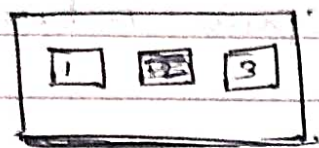
this will set that flex item at the end if its order is greater than other flex items.

* flex-grow: num

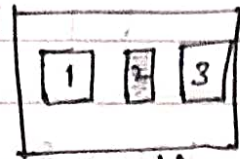


* flex-shrink: num

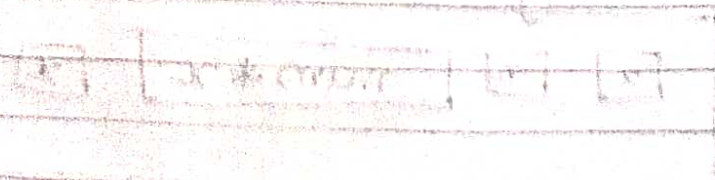
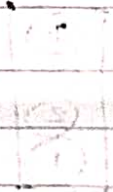
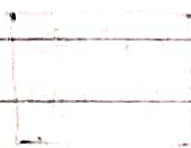
shrinks that flex item when window shrinks to a certain limit and if window is normal then that flex item will be in normal size but shrinks only when window shrinks.



big window

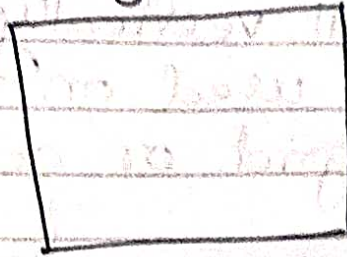


small window

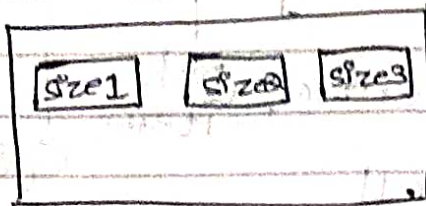


Grid CSS

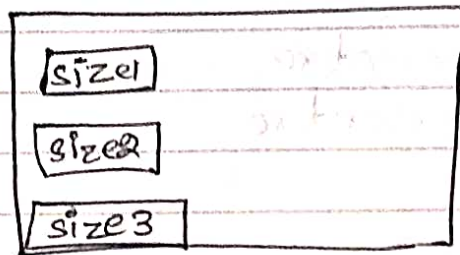
* `display: grid`



* `grid-template-rows: size1 size2 size3`



* `grid-template-columns: size1 size2 size3`



* `row-gap: size`

row gap brings gap b/w rows.

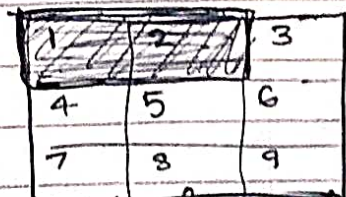
* `column-gap: size`

column gap brings gap b/w columns.

→ You can use `1fr` in place of size
`1fr` means 1 fraction of total width available.

* `grid-column-start: 1`

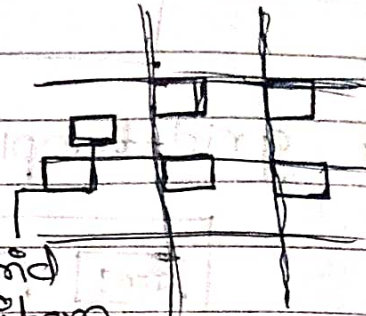
`grid-column-end: 3`



spans from column 1 to till before column 3.

220bnp1

- * align-items: centre
aligns grid cell vertically centre
and can be used on individual
cell of the grid or container
grid.



- * Justify-self: centre. ^{grid} item

aligns grid cell
horizontally centre.

Grid layout.

- * align-self: centre
- * place-self: centre

