

Grid Sizing & layout - Fixed Sizing

* grid-template-rows: 100px 200px ;
↓ no comma.

→ We can set like above but it will not be responsive

→ We can set rem also but it will not be responsive and it is fixed.

Short-hand:

grid-template: 100px 200px / 400px 300px ;
| |
| |
| Rows | Columns .

24

Wednesday

144.221

MAY
2023
Week 21

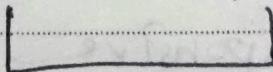
Auto-Sizing:

Column

- * grid-template: 100px auto;
- (*) It automatically takes the space (100%).
- (*) It is responsive.
- (*) Catch: Auto does not work 100% for the rows. Instead rows takes up to content height.

Ratio Sizing (fractions):

→ grid-template: 1fr 2fr :



1 : 2 ratio

(*) Responsive

(*) Always maintains the ratio no matter how big the content is.

Min - max Sizing :

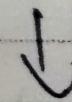
→ grid-template-columns : 100px minmax(400px, 800px)

* When there is enough space extend the second column width upto 800px. but while shrinking shrink upto 400px. below that content will not be shrunk.

* Highly responsive -

Repeat property :

- grid-template-rows : repeat(2, 100px);



=

→ Count

- grid-template-rows : repeat(2, 100px);

* When using the repeat if we the given count to repeat the value we have given

26

Friday

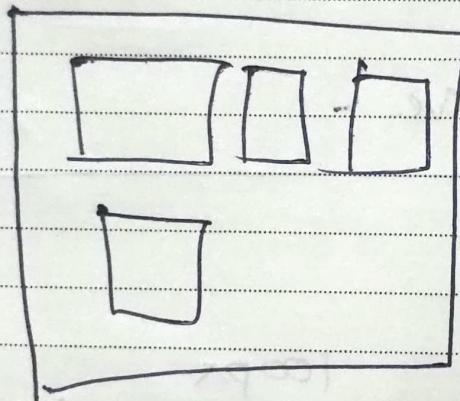
146-219

MAY
2023
Week 21

Extra column or row $\div (\text{items} \geq 4)$

- grid-template-columns: repeat(3, 200px);
- grid-template-rows: repeat(2, 200px);

* there are only 4 items by we defined (2×3) grid which give extra space.



Catch

* In this scenario
grid-auto will
not work

* It will not add anything to the grid to fill up the extra space.

less column or row $\div (\text{items} \geq 5)$

- grid-template: 100px 200px / 100px 200px;
- * Here we created (2×2) grid but there 5 items.

MAY
2023

Week 21

27

Saturday

(*) The extra item will be added

(*) Catch : The extra item will be matched to the column size given. but the new size will be based on the content height.



The above can be resolved by :

→ grid-auto-row : 300px ;

* it will add a row of already existing column and change its row height from content height to the new height we given.

28

Sunday

Week 22

148-217

Summary :-

Fixed Sizing - ((px) or rem)

Auto Sizing → Auto

Ratio Sizing → fr

Repeat

Extra column or row (less items)
less column or row (more item)

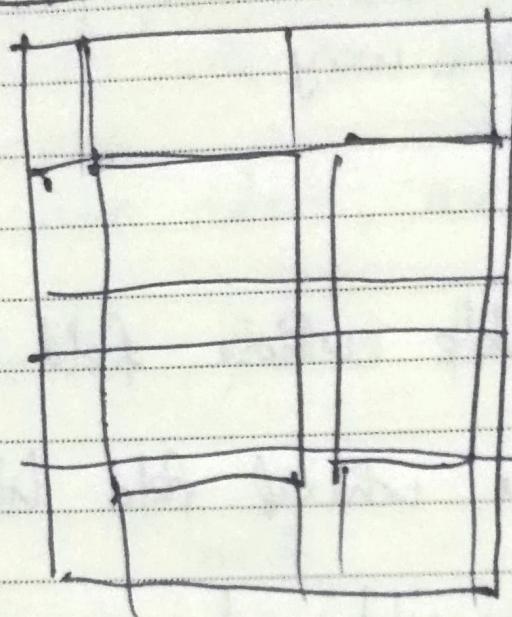
MAY S M T W T F S S M T W T F S S M T W T F S S M T W T F S

05 - 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 - - -

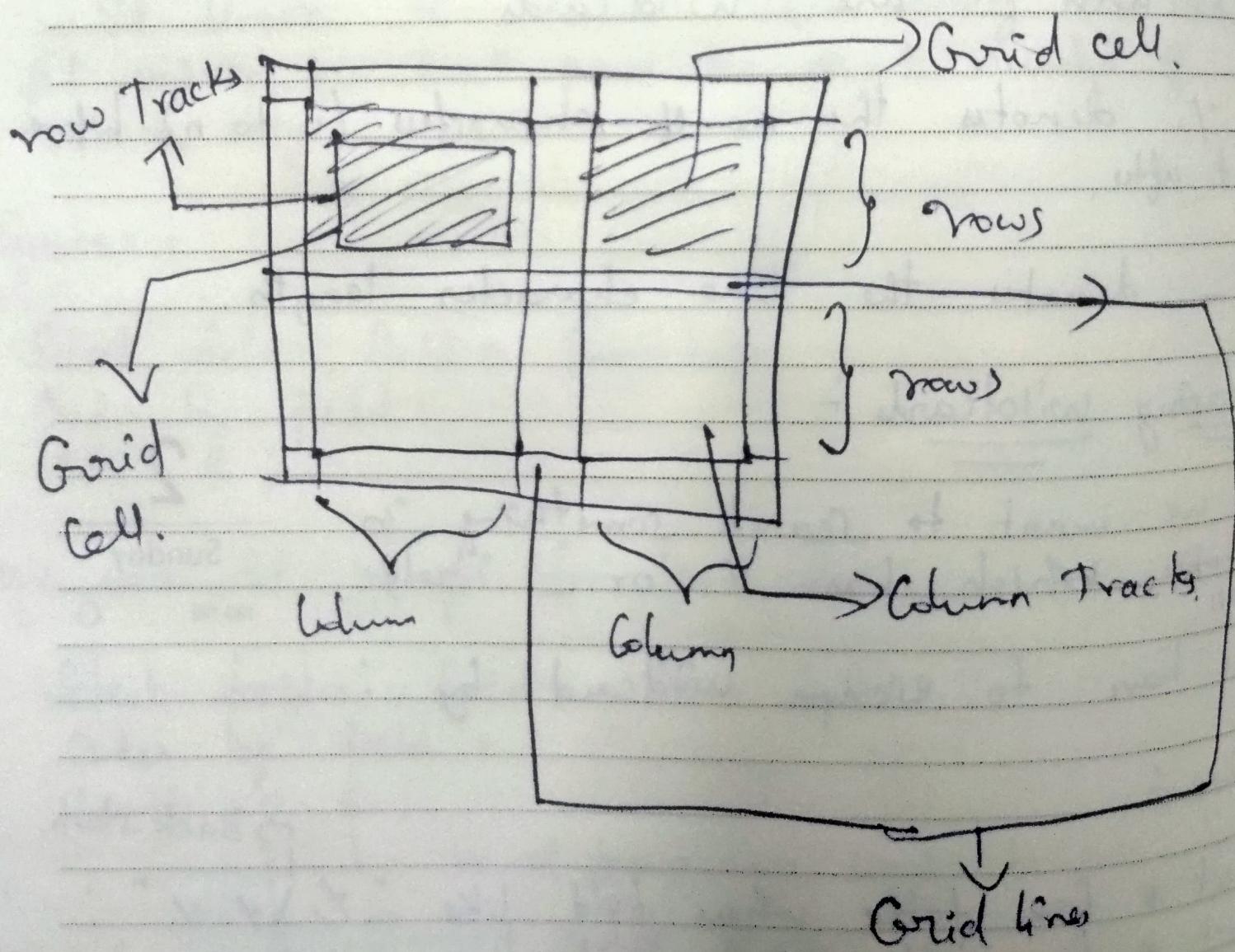
grid Auto rows



Grid - placement:



→ Grid contains.
(div)



All terms :-

- Container
- items
- tracks (Row and column tracks)
- Cells
- Grid lines (lines separating tracks)
 - We can control only by gap property.

Created by
grid-template
property

Grid - Column - Property :-

{ Grid - Column : Span 2 }

Take the column track from the default position of the column track span or extend by 2 cells.

This is a short hand property

Extended Version :-

{ Grid-column : Span 2 }

or

→ Grid-column - start : 2

→ Grid-column - end : 3

* 2, 3 are the cells order.

* To get to span or extend to last item always use -1

* To start from the first item always use positive integer.

* We can reverse the extend or span also.

* Similarly we use :

{ Grid-row : Span 2 }

→ Grid - row - start : 2

→ Grid - row - end : 3

Grid-area :

Ex: { grid-area : 2 / 1 / 2 / -1 }

* Grid - area is a short hand of 4 properties.

{ Grid-area : 2 / 1 / 2 / -1 } =

{ Grid - row - start : 2 ←

Grid - column - start : 1 ←

Grid - row - end : 2 ←

Grid - Column - end : -1 } ←

Grid - Overlapping :

* We can overlap grid-items over one another which is a very cool in grid.

Increasing Transparency Using Hex code :

1 2 3 4 5 6 80

It is the
RGB color.

80 here means
50% Transparency

Algorithm to write grid :

- 1) check How many divs need
- 2) check How many rows and columns
- 3) We Span and Grid area accordingly
- 4) We inspect and check the cells properly.
- 5) Repeat by doing complex layouts.