

Ch000 - 10

CSS Continue

ODI Process

- ↳ Company Provides you E-Shop
- ↳ जो Company Stock के बनी होती है
- 100% में से के $\frac{5\%}{1}$
- इस 5%. के 100%
Gridbox का लिया जाता है
- * Create own SaaS Product and Sale it
- * and Know Your Potential

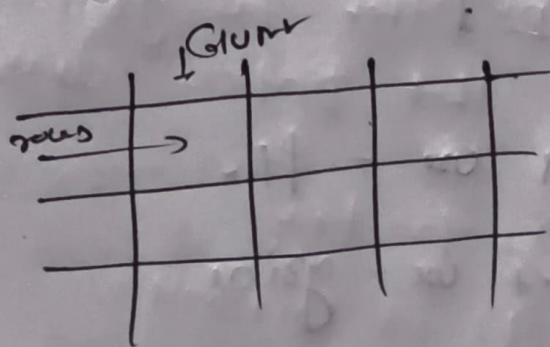
flexbox

→ Use when alignment elements in main axis

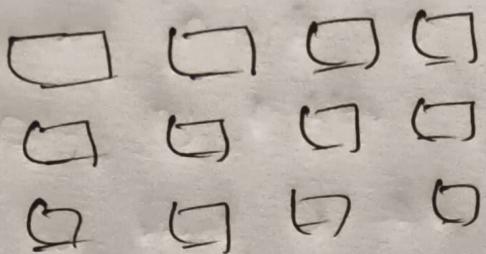
→ 1D

Grid

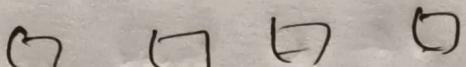
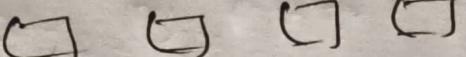
→ made of rows and Columns



Hollow (Practical ex. Gr. of gear)

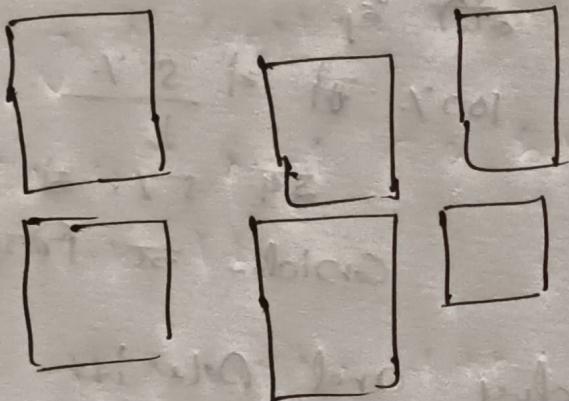


Ans. 220

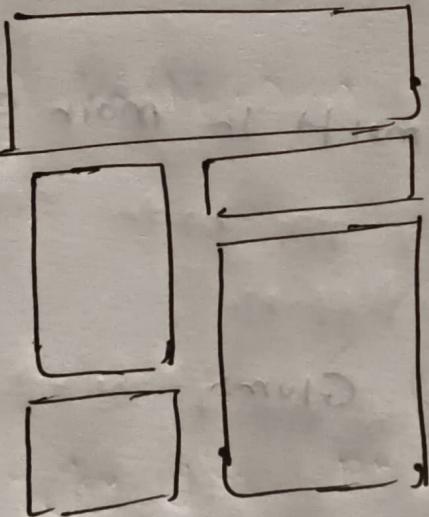


Ans. 200

Pittement



Borda Grid



* One arbor alignment use - flr

* full arbor alignment use - gear

HTML

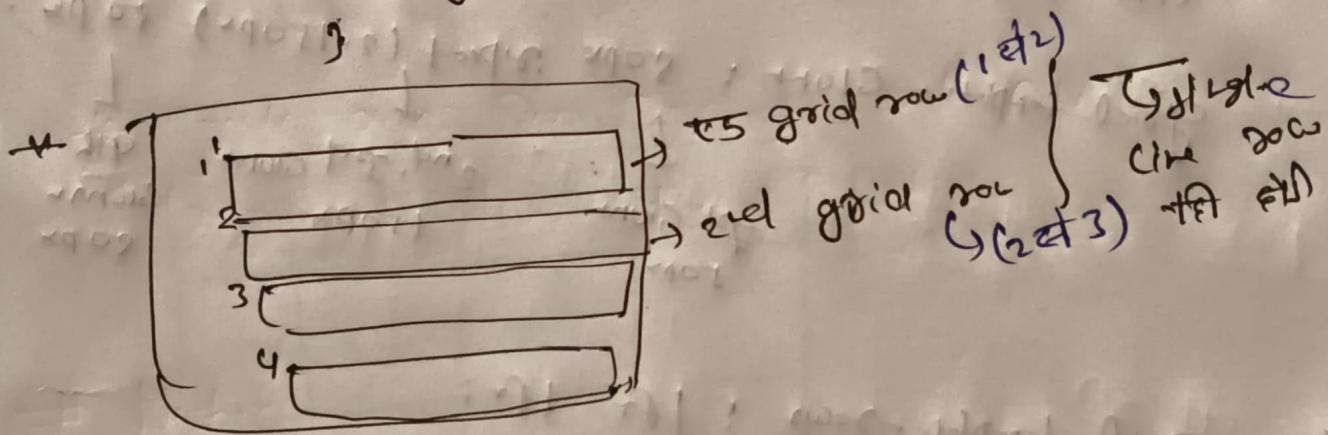
```
<div id="parent">
  <div class="child child-1">1</div>
  <div> - <div>2</div>
  <div class="child child-2">3</div>
  <div class="child child-3">4</div>
</div>
```

CSS

* grid apply on Parent

Parent {

display: grid



+ # Parent {

grid-template-columns: 10px 10px 10px;

grid-template-rows: 10px 10px 10px 10px;

→ it is just declare a template but not
really declare empty boxes

so you need 10px box then you not
need to create or write

grid-template-rows: 10px 10px ...

Q

→ so we need a value which
solves my problem

→ we need 4 box and each one 100px

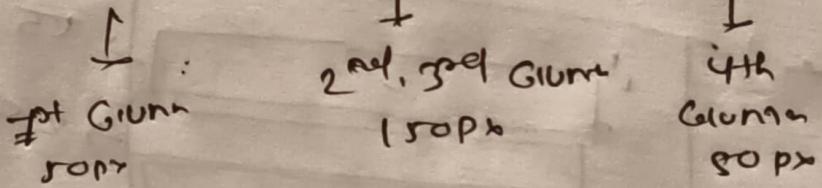
grid-template - rows : repeat (4, 100px)

grid-template - columns : repeat (4, 100px)

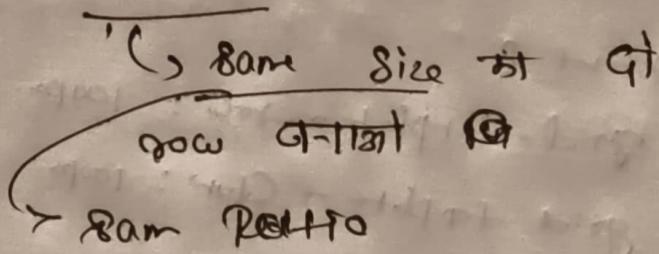
→ but we need each box in different size.

(like this)

grid-template - columns : 50px repeat (2, 150px) 50px



→ Grid-template - rows : 1fr 1fr;



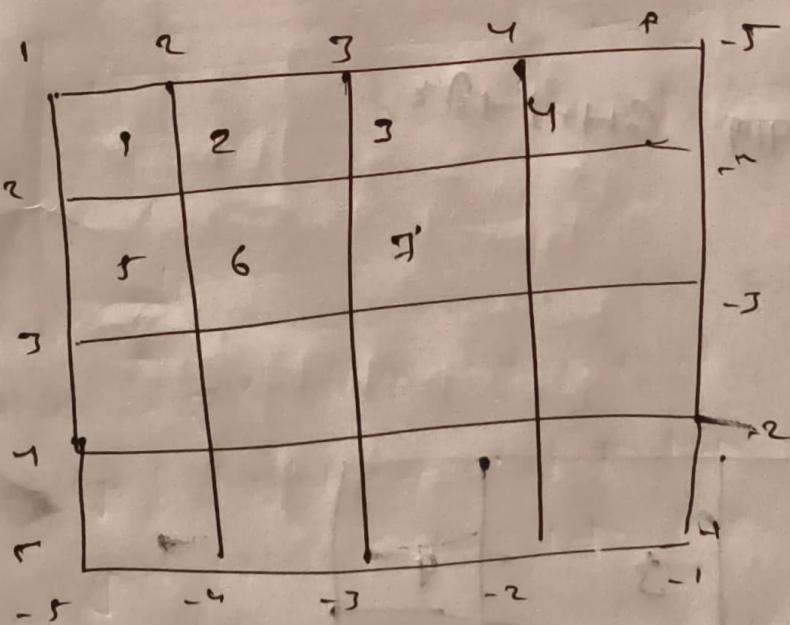
→ enter your given px then the grid will
stretch with that particular width px

→ you can use $\text{fr} \rightarrow \text{fr}$ for

6 q+ is automatic
adjustable

- * grid-template - Glur : 1fr;
 - ↳ 1 Column of 80% Single Ratio
 - ↳ 20%
- * grid-template - Glur : 100px 1fr;
 - ↳ 1st Column 100px and
 - 2nd Glur captures remaining
 - Space : 0

- * gap
 - ↳ space b/w children element



- * * whole child at position 8x1

child - 6 {

grid - row - Start : 4 ;

grid - row - end : 5 ;

grid - Glur - Start : 3 ;

grid - Glur - end : 3 ;

~~grid-area~~

child - 6 :

grid-area : - 2 / - 1 ;

grid-column : 3 / 4 ;

grid-area : row start | col-start { second },
grid-area : row end | col-end ;

grid-area : - 2 / 3 | - 3 / 4 ;

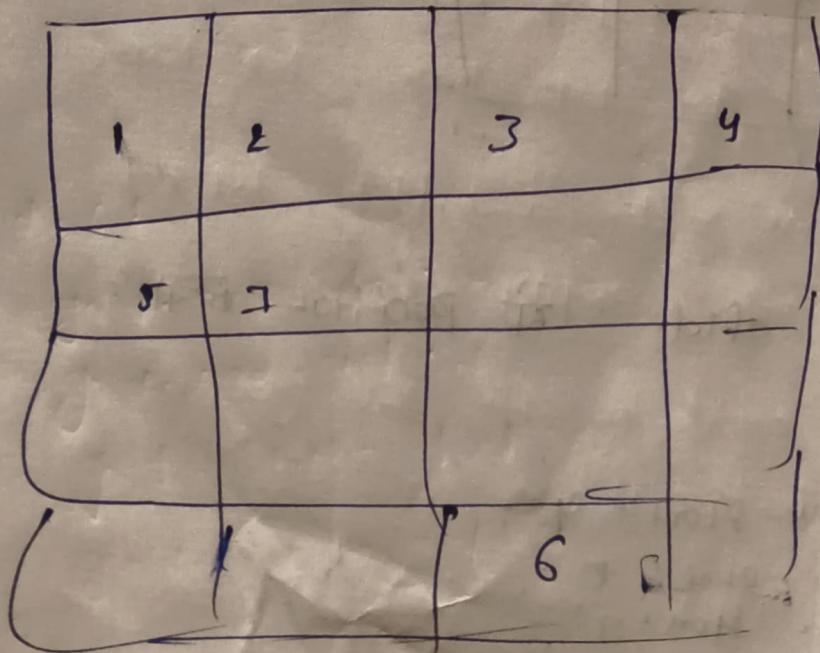
↳ short hand notation

class . and & #

Note ~~grid-area~~ A class . and &

&

Note Grid के पाल Control attr.

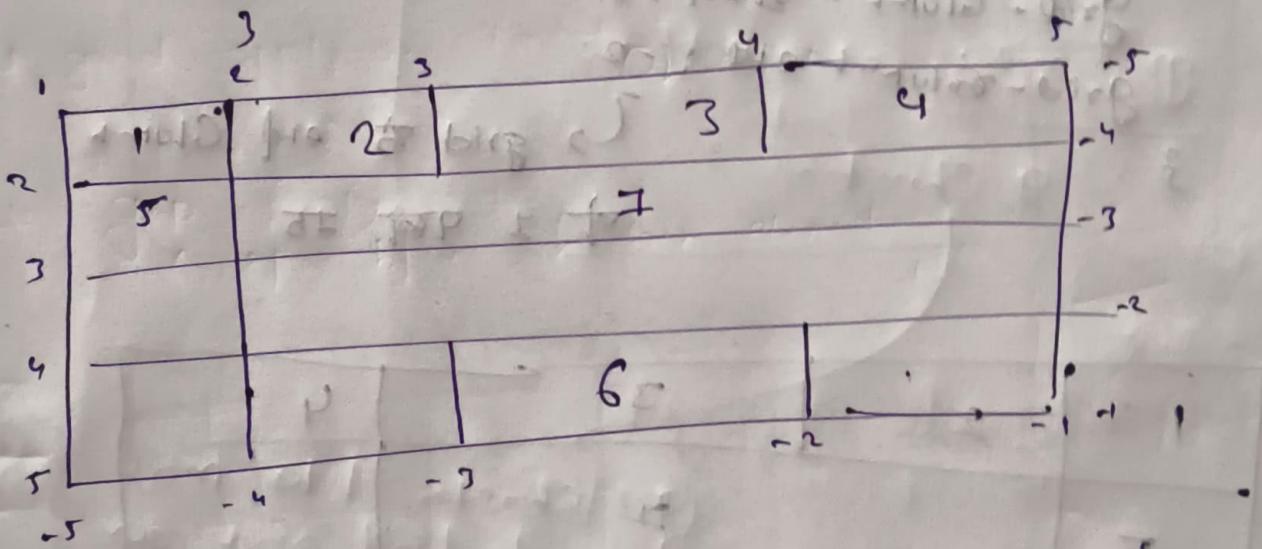


Now let ~~the~~ ~~the~~ Chol Client aint put
alway of the roof
then

· child = ?

grid - Glum - Sto : 2;

gric0-Glunt - evol : 5:



to get to occur i increase the number of Glue
then : always put (E) child element , key you
start from the down and end to those grid
by out

Parent 1

→ borders! or no solid red:

dippy: gril:

grid transfer - Glumc : 100% 1fr 2fr 1fr 1fr

grid-template-rows : repeat(4, 100px)

so if you want you ref. to give
manually value of any element either always
to the end of grid then you need to

choose one value

• Child - 7 {

grid - Glumr - Start : 2;

grid - Glumr - End : -2;

}

↑ grid के end Glumr

से 1 पस्त रह

1	2	3	4
5		2	
6			7

• Child - 6 {

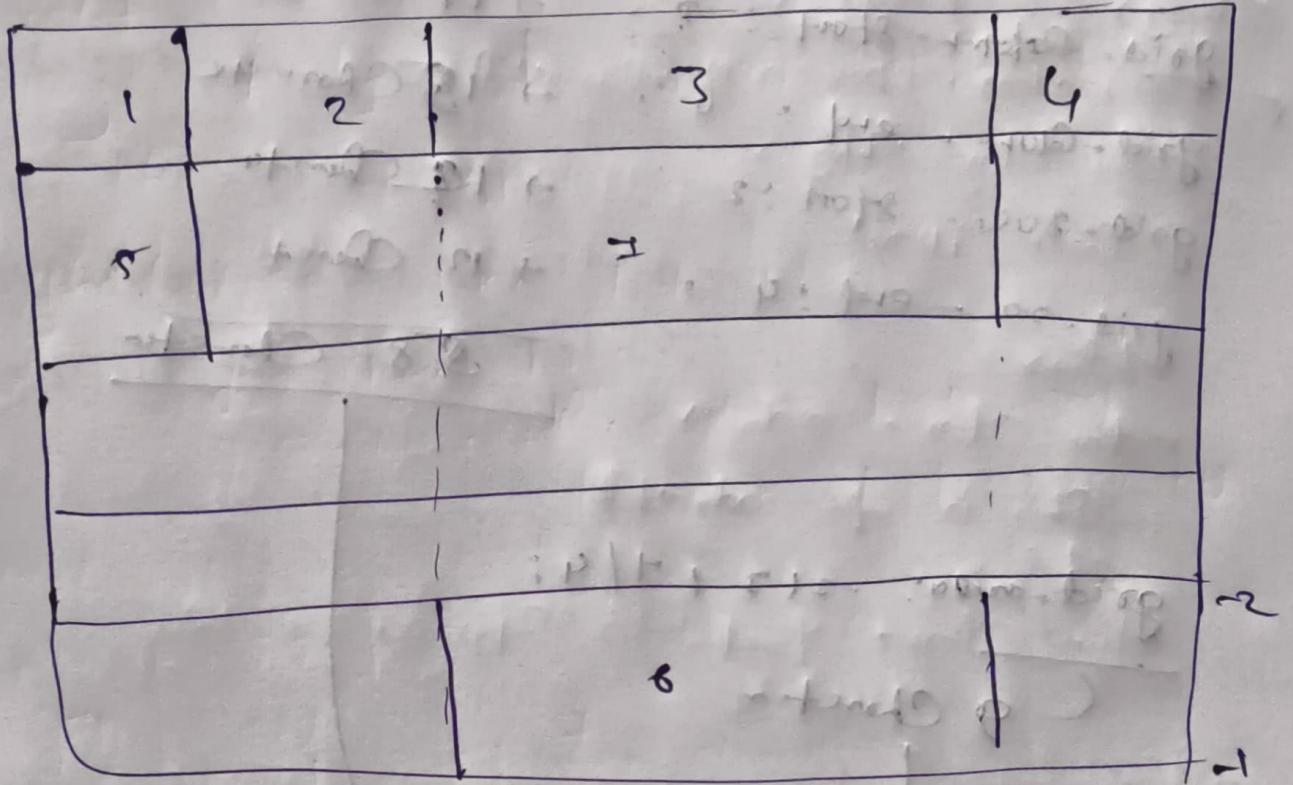
grid - row - Start : -2;

grid - row - End : -1)

grid - Glumr - Start : 3;

grid - Glumr - End : 4;

,



Short - Hand Notation

grid-row-start : -2; } grid-row : Start / End;
 grid-row-end : -1; }

grid-child-start : 3; } grid-Child : Start / End;
 grid-child-end : 4; }

grid-area : row-start / Col-start;

grid-area : row-end / Col-end;

grid-area : -2 / 3 / -1 / 4;
 PS ↓ ↓ ↓
 CS RF RF

grid - feature - start : 5 ; \rightarrow 17 Character
 grid - Glut - end : 6 ; \rightarrow 15 Character
 grid - new - start : 2 ; \rightarrow 16 Character
 grid - sa - end : 4 ; \rightarrow 12 Character

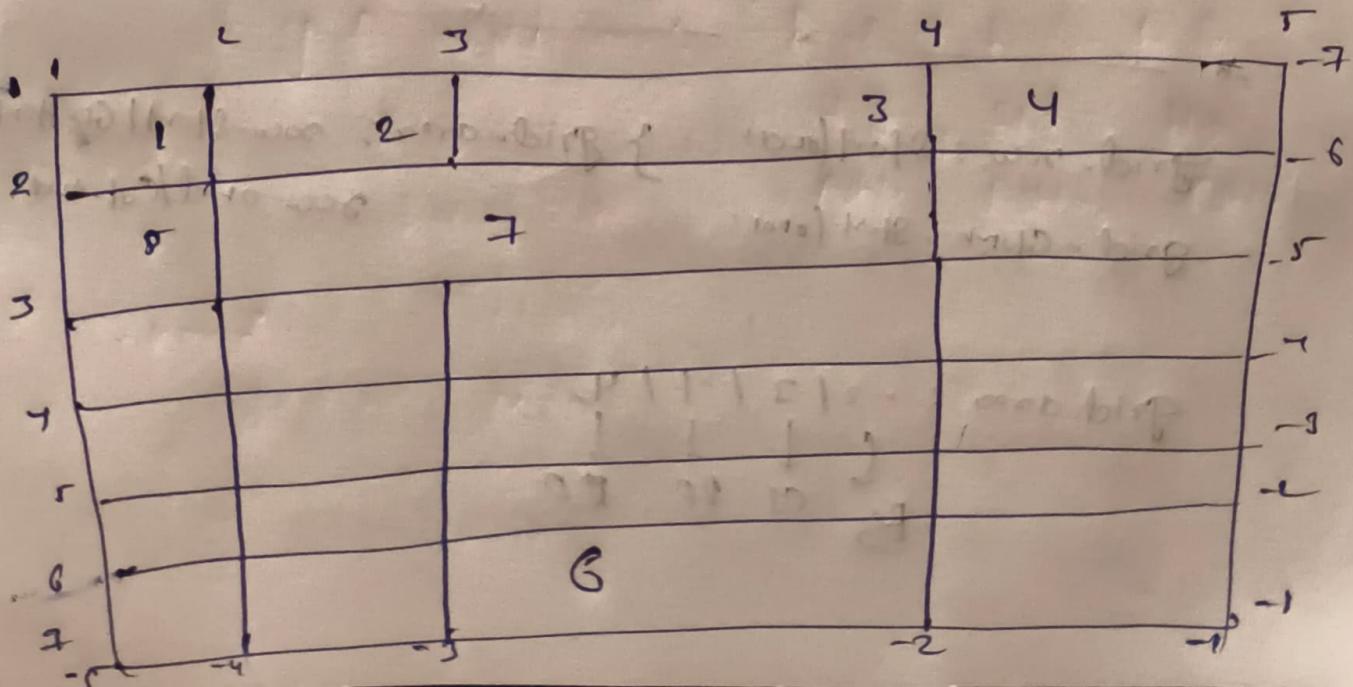
88 Character

grid-order: -2 3 1 7 4 ;

9 Character

(88) Character takes more time to search our browser

but afterwards (9) Character is too fast.



but qd i want (6) child achieve this
 column , so in that case use on (span)
 "bold" - add value
 "width" - add
 .child-6 {
 grid-area: -2 3 1 -1 1 5;
 ↗ grid-area: -2 3 1 -1 1 span 2;
 ↗ qd mean from blurb
 ↗ go two Open
 ↗ Take off
 ↗ 39c

→ ∞ — ∞ — ∞ —
 .child-6 {
 grid-area: -2 3 1 span 1 / span 2;

Note Grid का तरीका footer वाले use होते हैं
 फिर ऐसी,
 ↗ grid-template-areas

```
<body id='Parent'>  
<div class="nav">NavBar</div>  
<div class="blog">  
    <h1>Input</h1>  
</div>  
<div class="ads">  
    This is for ads  
</div>  
<div class="Footer">Copyright 2020</div>
```

Style.css

```
body {  
    background-color: #1E1E1E;  
    color: white;  
    font-family: sans-serif;  
}
```

```
#Parent {  
    display: grid;
```

```
grid-template-areas: "header header"  
                    "blog blog";
```

```
header {  
    grid-area: header;  
    background-color: black;  
    color: white;  
    padding: 10px;  
}
```

```
blog {  
    grid-area: blog;  
    background-color: black;  
    color: white;  
    padding: 10px;  
}
```

```
.nav {
```

```
    grid-area: header;
```

```
,
```

```
.blog {
```

```
    grid-area: blog;
```

```
,
```

.cols

grid-area: side;

3

tailwind

play.tailwind.com

```
<div class="bg-zinc-950 text-zinc-100 min-h-screen flex justify-center items-center">  
  <h1 class="text-4xl font-extrabold">Cras ipsum animi!  
</div>
```

breakpoint

sm	40rem
md	48rem
lg	64rem
xl	80rem
xxl	96rem

```
<div class="bg-zinc-950 flex flex-col items-center justify-center min-h-screen gap-10">
```

```
  <div class="w-fit max-w-md border-white">
```

</div>

wid-fit to fit flex-item lena ~~fit~~ fit

main track main width ($md = 48\text{rem} = 48 \times 16$)

रहेंगे। मतवाव इसके बारे में

width नहीं करना था

```
<form action = " " class = " ">

<div>
  <label class = "mb-1 block text-DT text-align-left">
    for = "email" > Email </label>
    <input class = "focus:outline-w-full rounded-lg border border-zinc-400 px-4 py-2 type = "email"
      name = " " placeholder = "Enter your email" id = " " />
  </div>

  <div>
    <label class = "mb-1 block text-sm text-align-left mt-4" for = "email" > Password </label>
    <input class = "focus:outline-w-full rounded-lg border border-zinc-400 px-4 py-2
      focus: border-orange-500 focus:zinc-2
      focus: outline-orange-500" type = "password" name = " " placeholder = " " id = " " />
  </div>
```

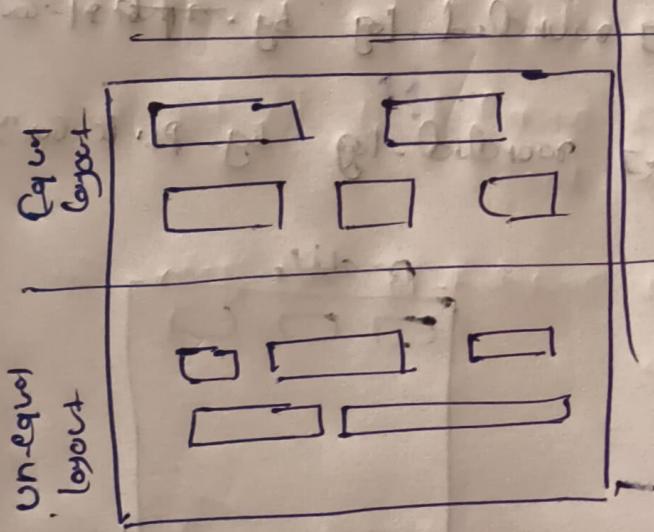
```
<div class = "flex main-h-screen items-center justify-center bg-black-200-950
  flex - zine - 100">
  <div>
```

Responsive problem with how we use CSS

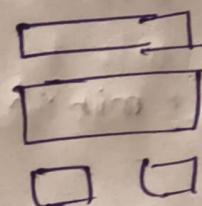
Breakpoint →

→ if CSS set to work on big screen
→ break point

Web



mobile



one level
layout

Following ~~part~~ class

→ bg-orange-500

sm: bg-orange-500 → this class act from small & above

md: bg-orange-500 → this class act from medium & above

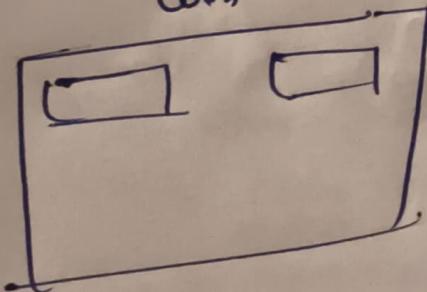
① <div class="grid sm:grid-10-2 gap-4 m-4">

<div class="min-h-[100px] rounded-lg bg-amber-600"></div>

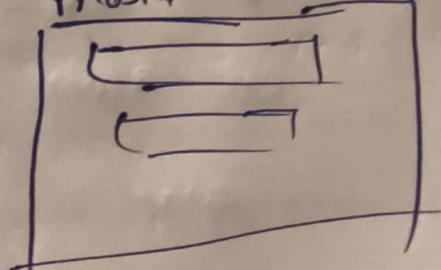
<div class="min-h-[100px] rounded-lg bg-teal-600"></div>

</div>

Web

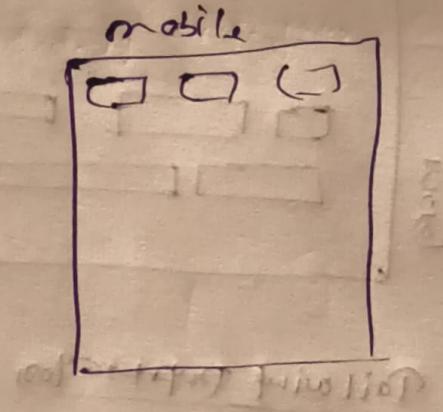
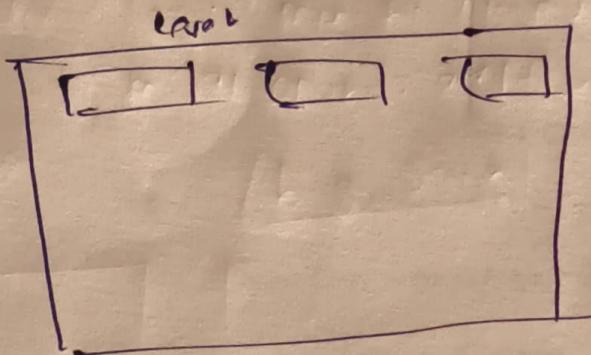


mobile

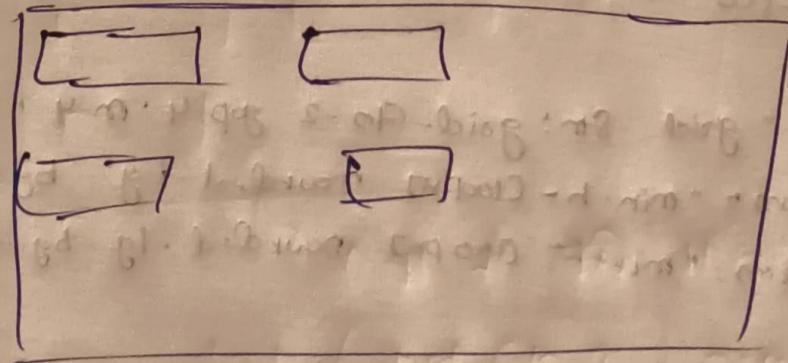


If we want in mobile and big screen child
in 3 @ diff. Column

```
<div class="grid sm:grid-cols-3 gap-4 m-4 grid-cols-3">  
<div class="min-h-[100px] rounded-lg bg-amber-600">  
  </div>  
<div class="min-h-[100px] rounded-lg bg-#f0f0f0">  
  </div>  
<div class="min-h-[100px] rounded-lg bg-pink-600">  
  </div>
```



grid-cols-2



<div class="grid sm:grid-glo-4 gap-4 m-4 grid-glo-2">

I
it small device पर
उसे बजा के लिए

I
ये mobile के
लिए होना है
(-फुट में व).
के लिए यह
define की

चिपा अपनाविए

sm:grid-glo-4

dev. device के

ये लगा जाए

grid grid

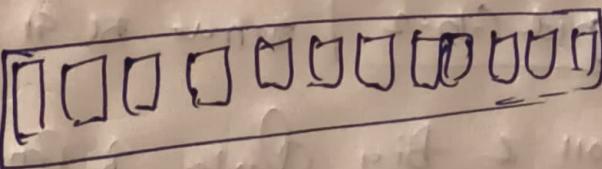
Note :- sm
↳ 40rem (640px)

↳ 640px आ इसके बजे

This is our first Column Grid

Now turn to our grid Grid

↳ Bootstrap Introduce, your Complete Stack 12 parts
menu 12 Columns

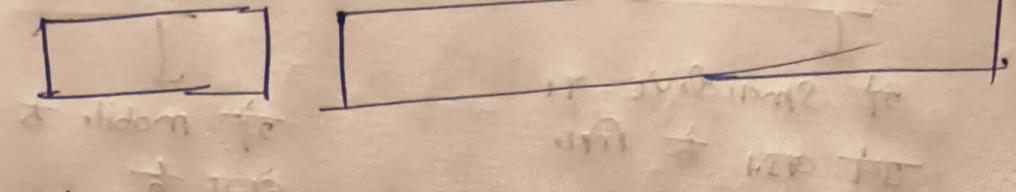


<div class="grid sm:grid-glo-12 gap-4 m-4 grid-glo-2">

<div class="min-h-[100px] rounded-lg bg-amber-600 sm:gl-0px">
</div>

<div class="min-h-[100px] rounded-lg bg-teal-800 sm:gl-0px">
</div>

</div>



<div class: "grid sm: grid-1g-2 gap-2 m-4 grid-1g-2">

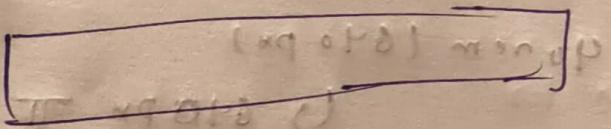
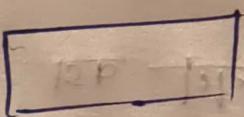
<div class: "min-h-[100px] rounded-lg bg-amber-600 sm: gl-opar-3"></div>

<div class: "min-h-[100px] rounded-lg bg-teal-600 sm: gl-opar-6"></div>

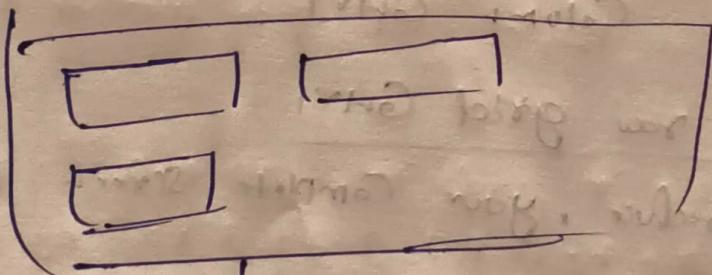
<div class: "min-h-[100px] rounded-lg bg-green-600 sm: gl-opar-3"></div>

</div>

big screen



Small Screen



लेटा तो नहीं आविष्कार करें

जैसे [grid-1g-2], remove कर दो

अब small & big screen के बीच

बोता

Note In the small screen Right side
Position ~~is~~ should be removed

```
<div class="grid sm:grid-col-12 gap-2 m-4">  
  <div class="min-h-[looppx] rounded-lg bg-amber-500  
    sm:gl-span-3></div>  
  <div class="min-h-[looppx] rounded-lg bg-teal-600  
    sm:gl-span-6></div>  
<div class="min-h-[looppx] rounded-lg bg-green-600  
    sm:gl-span-3 hidden"></div>  
</div>
```

→ ये तीर्थ child element small device, mid.
device, big device के लिए हैं
hidden के लिए

, ~~<div class="grid sm:grid-col-12 gap-2 m-4">~~
~~<div class="min-h-[looppx] rounded-lg bg-green-600
sm:gl-span-3 hidden sm:block"></div>~~

6

→ small या sm. device
visible ("block होता)
but small के छोटे नहीं hidden
हो जाता

(display: none) के के element और
class के नहीं हैं, इसके

→ hidden के unhidden के लिए CC Non के unhidden