

**JOIN US & LEARN
WEB DEVELOPMENT**

MASTER THE **CSS** **GRID**

Grid Layout
Rows & Column
Repeat()
Column & row gap()
Grid gap
Grid Lines
How to use?



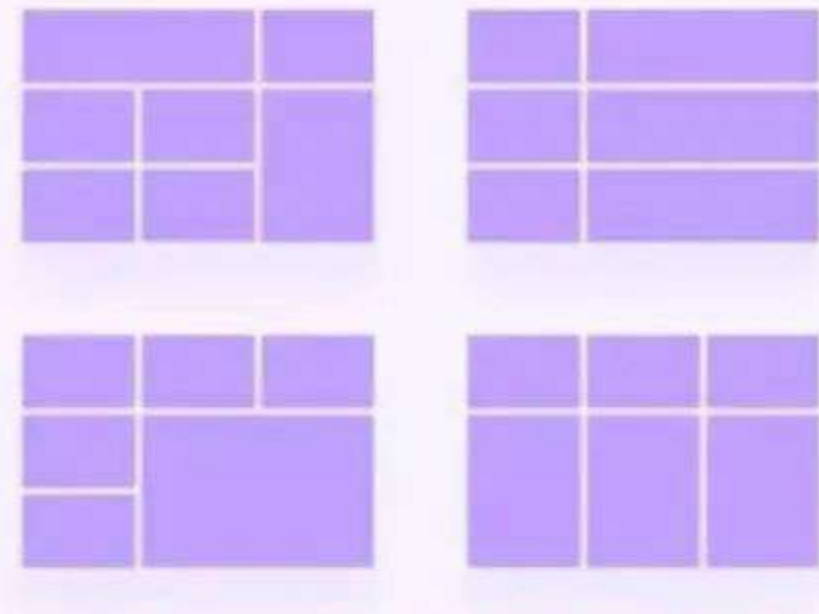
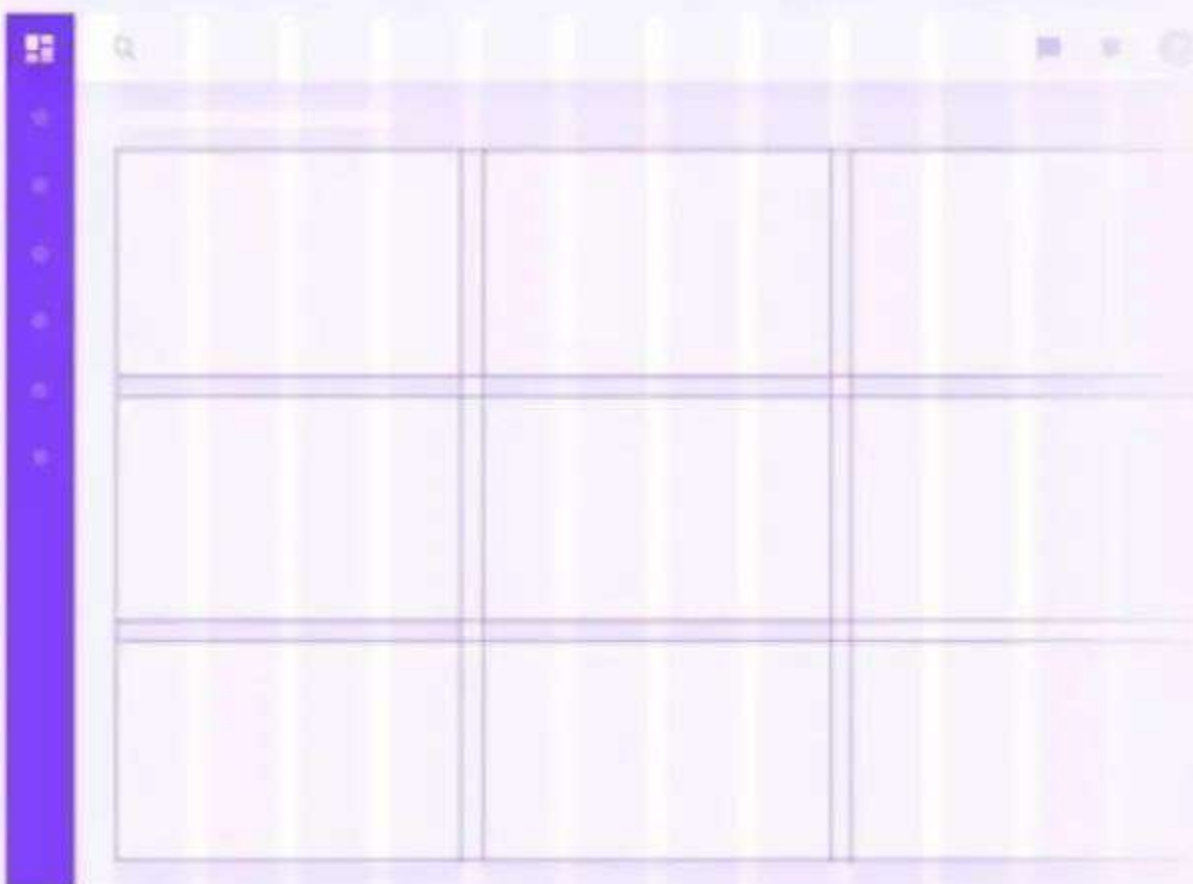
Learn all in one place



GRID LAYOUT

The **CSS Grid** Layout Module offers a grid-based layout system, with **rows & columns**, making it easier to design web pages without having to use **floats & positioning**.

EXAMPLE OF GRID LAYOUT :-



Understanding Rows & Column

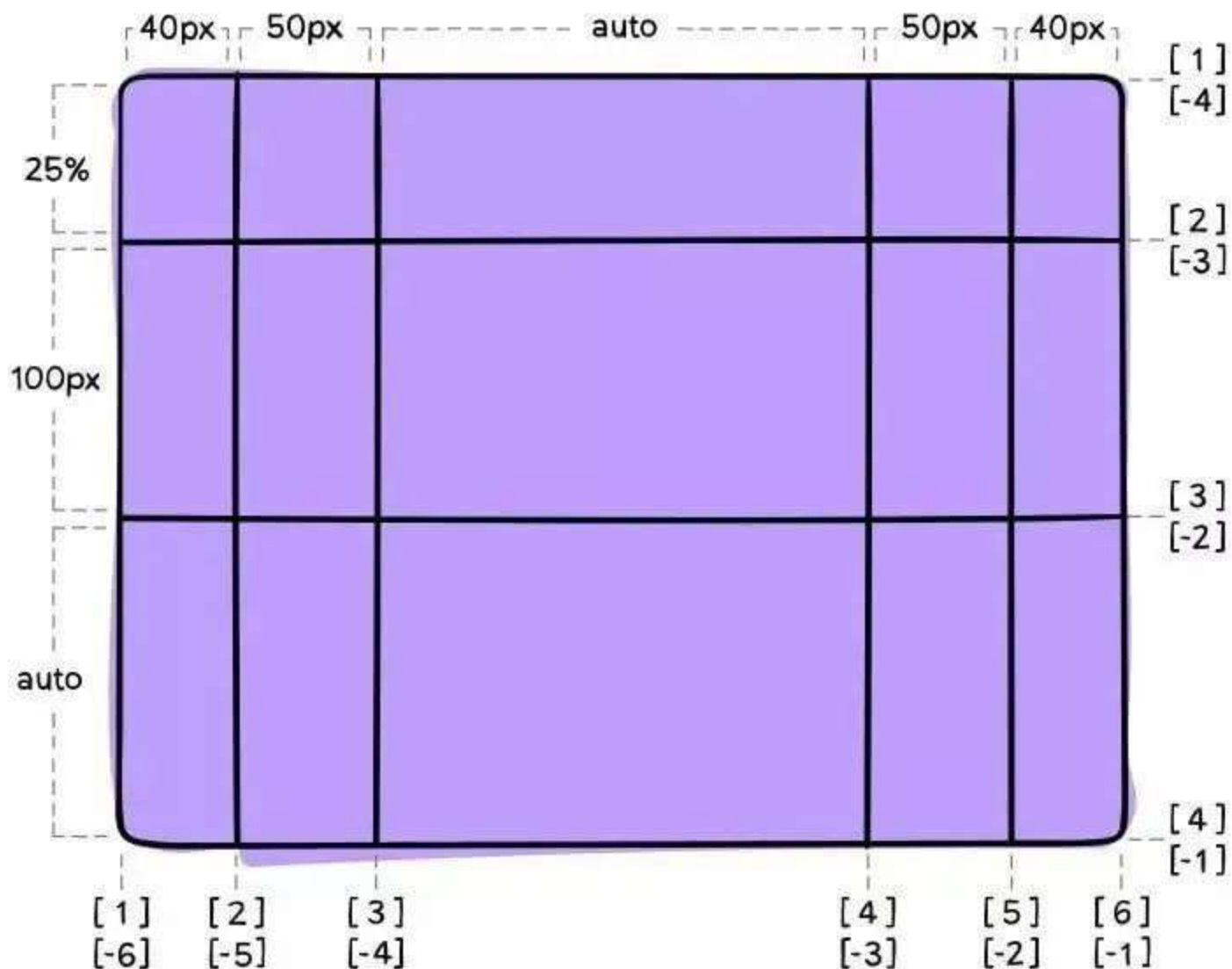


ROWS & COLUMNS



```
.container {
  grid-template-columns: 40px 50px auto 50px 40px;
  grid-template-rows: 25% 100px auto;
}
```

// SEE THE DIAGRAM BELOW TO UNDERSTAND IT



Repeat() method



REPEAT()

Using normal **rows** and **column** for grid



```
.container {
  display: grid;
  grid-template-columns: 1fr 2fr 1fr 2fr 1fr 2fr;
  grid-template-rows: 100px auto 20% 100px auto 20px;
}
```

We can use the **repeat()** notation like this *instead to simplify* :-



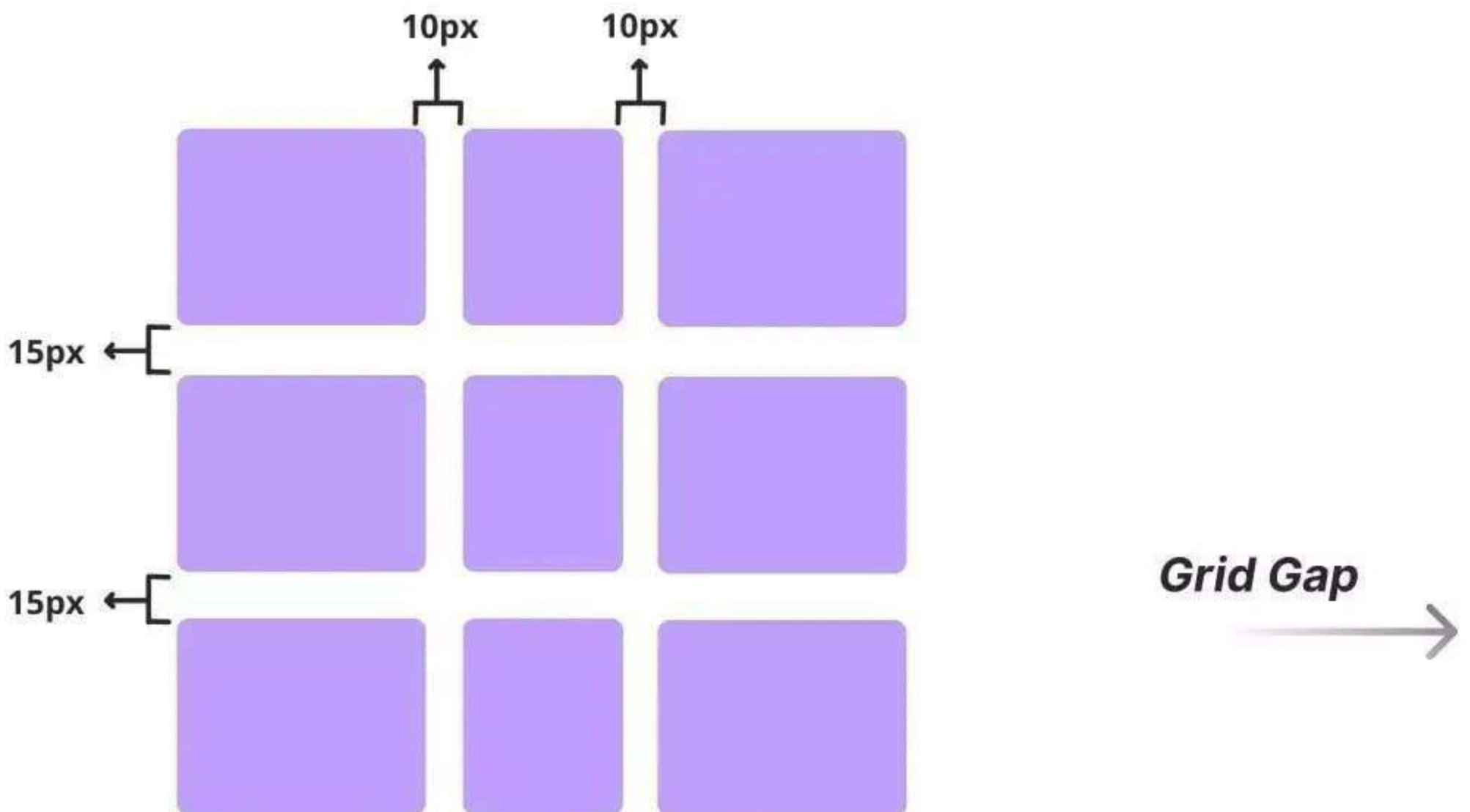
```
.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr 2fr);
  grid-template-rows: repeat(2, 100px auto 20%);
}
```

Column Gap & Row Gap



COLUMN GAP & ROW GAP

```
.container {  
  grid-template-columns: 100px 50px 100px;  
  grid-template-rows: 80px auto 80px;  
  column-gap: 10px;  
  row-gap: 15px;  
}  
// SEE OUTPUT BELOW
```



GRID GAP

A shorthand for **row-gap** and **column-gap** is **grid-gap**.



```
.container {  
  grid-template-columns: 100px 50px 100px;  
  grid-template-rows: 80px auto 80px;  
  column-gap: 10px;  
  row-gap: 15px;  
}
```



```
.container {  
  grid-template-columns: 100px 50px 100px;  
  grid-template-rows: 80px auto 80px;  
  gap: 15px 10px;  
  /*gap: <grid-row-gap> <grid-column-gap>;*/  
}
```

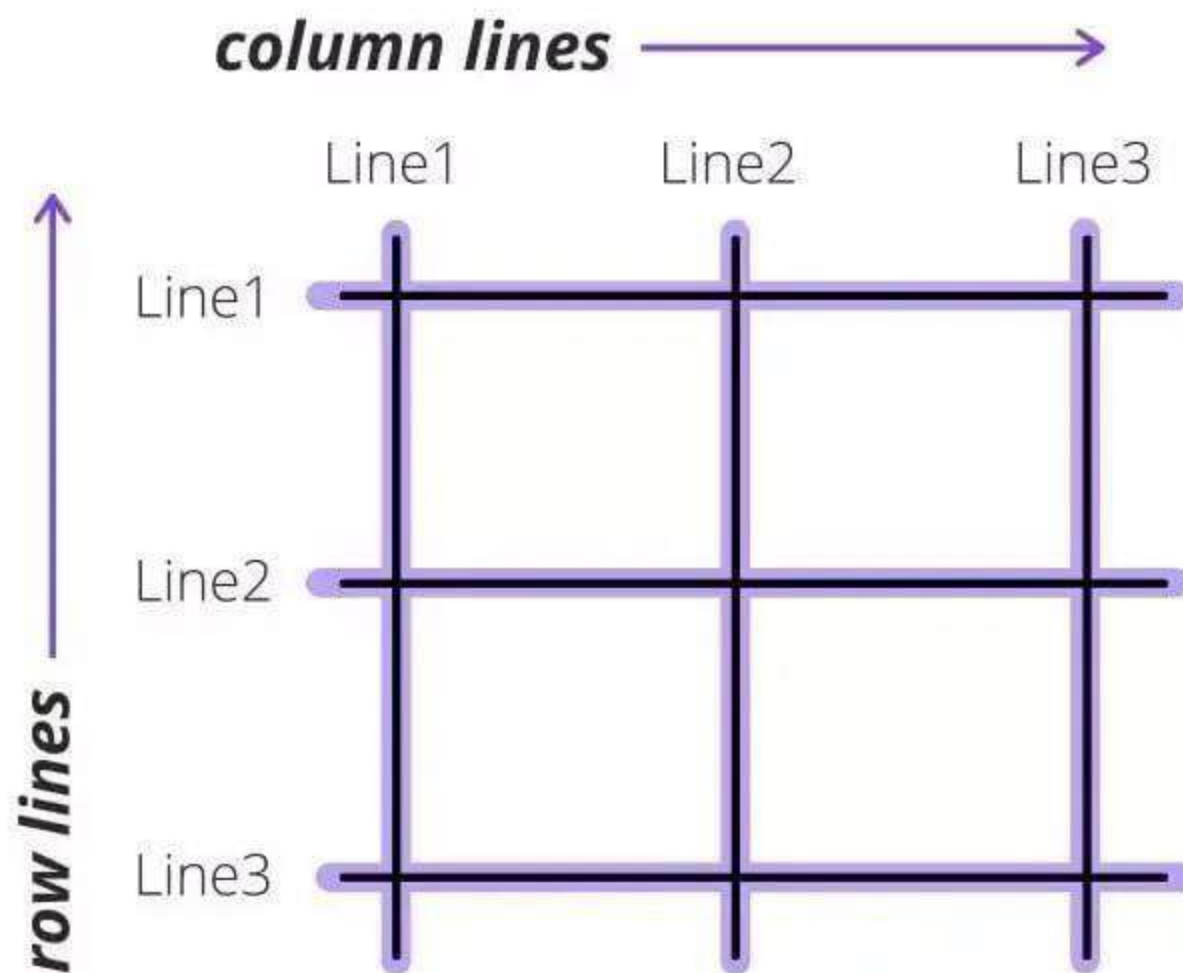
Understanding Grid Lines



GRID LINES

The lines b/w **columns** are called **column lines**.

The lines b/w **rows** are called **row lines**.



***" Refer to line numbers when placing a grid item
in a grid container "***

How to actually use these?



GRID LINES: BUT HOW TO USE?

Place a **grid item** at **column line 1**, and let it end on **column line 3** :-



```
.item1 {  
  grid-column-start: 1;  
  grid-column-end: 3;  
}
```

Place a **grid item** at **row line 1**, and let it end on **row line 3** :-

```
.item1 {  
  grid-row-start: 1;  
  grid-row-end: 3;  
}
```



What next now?



**Was it
helpful?
follow for more!**



Muhammad Zubair
@iamzubairarain



Did you know? I'm a freelancer too!  Visit my Upwork profile:
[<https://www.upwork.com/freelancers/~014ffd655fc0791835>]