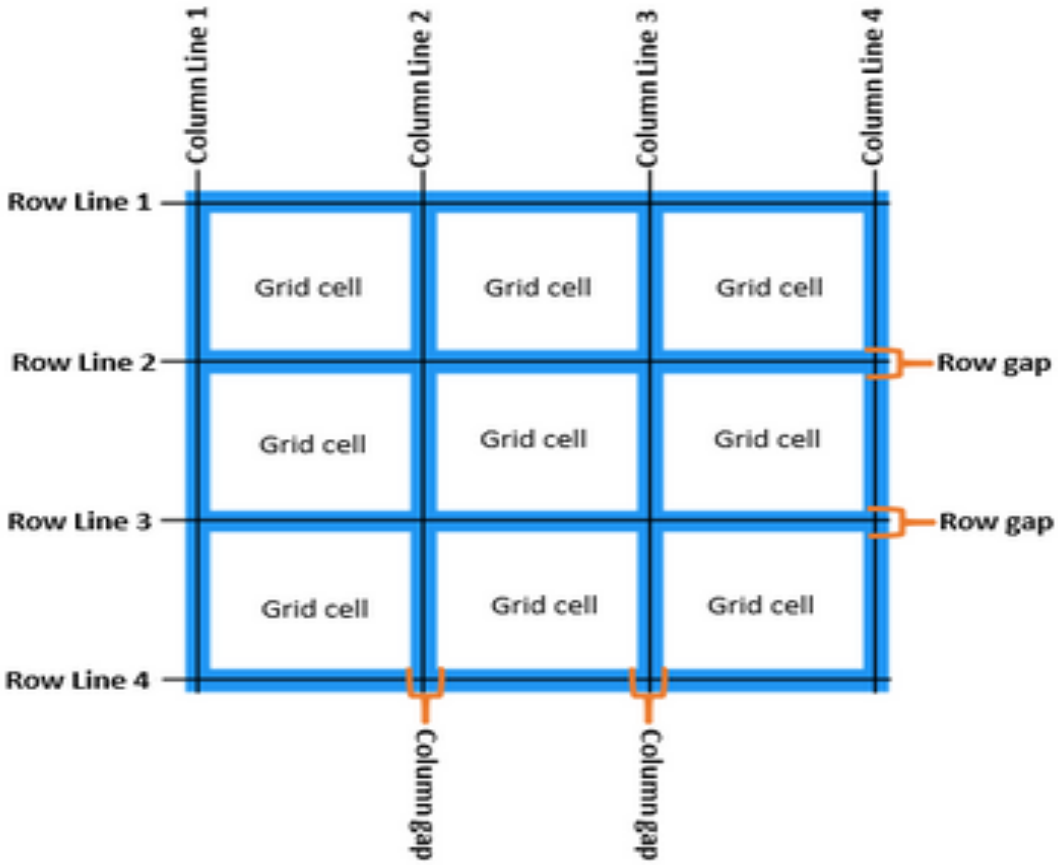
CSS Grid

- A Grid is a collection of horizontal and vertical lines, creating a pattern against which we can line up our elements
- It provides greater consistency on our website
- It is 2D layout model. Content is laid out in rows and columns.
- Grid has rows, columns and gap between each row and column. The gaps are commonly referred as gutters



Grid Vs FlexBox

Grid

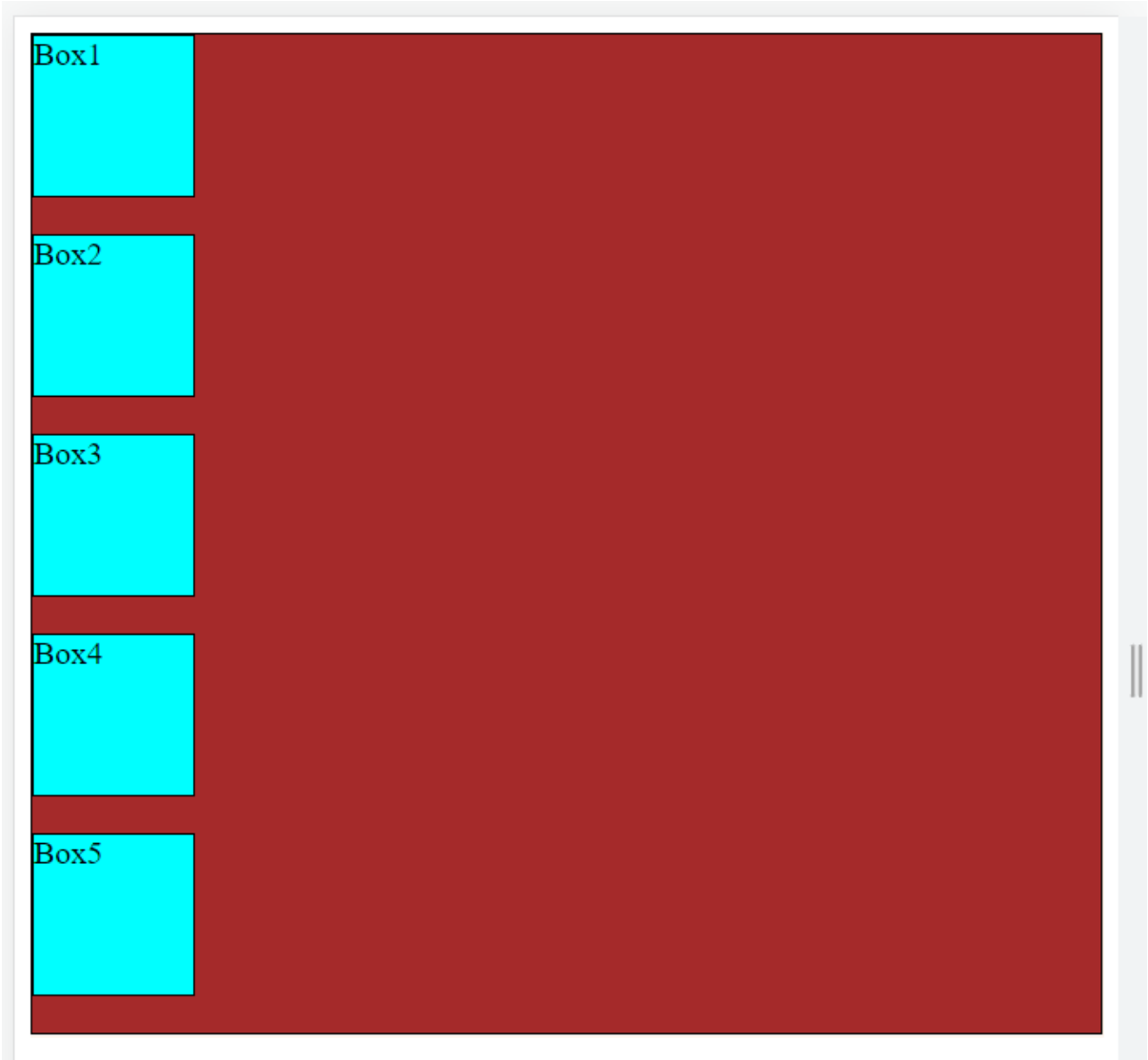
- It is 2D layout model
- It is use for layout creation

FlexBox

- It is 1D layout model.
- It is use to show how to position content / how content flows.

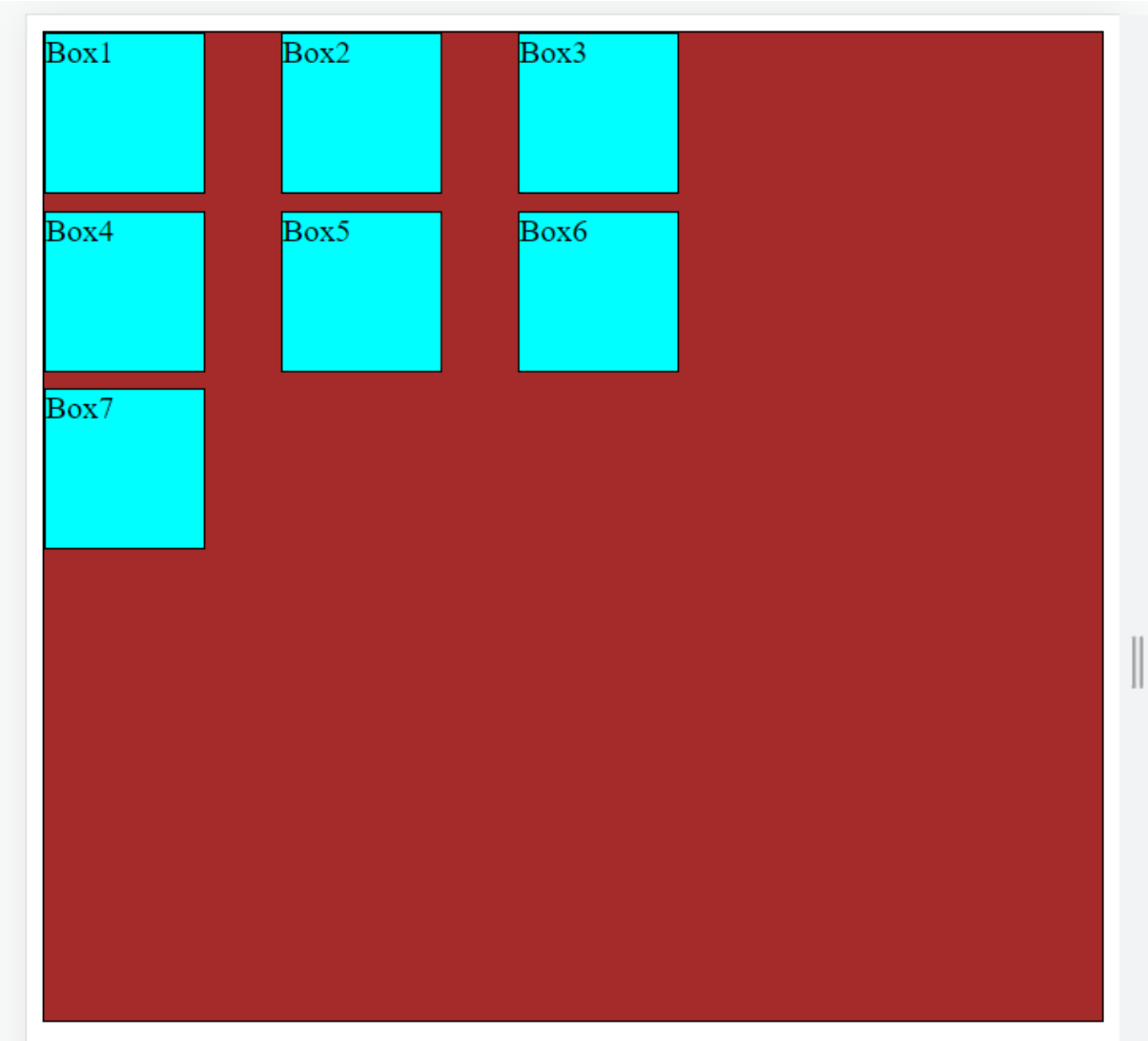
We can make a container as grid by using **display: grid;**

```
.container{
    background-color: brown;
    height: 500px;
    border: 1px solid black;
    display: grid;
}
.box{
    background-color: aqua;
    border: 1px solid black;
    width: 80px;
    height: 80px;
}
```



Grid Container Properties

```
grid-template-columns: 120px 120px 100px;
grid-template-rows: 90px 90px 90px 90px;
```



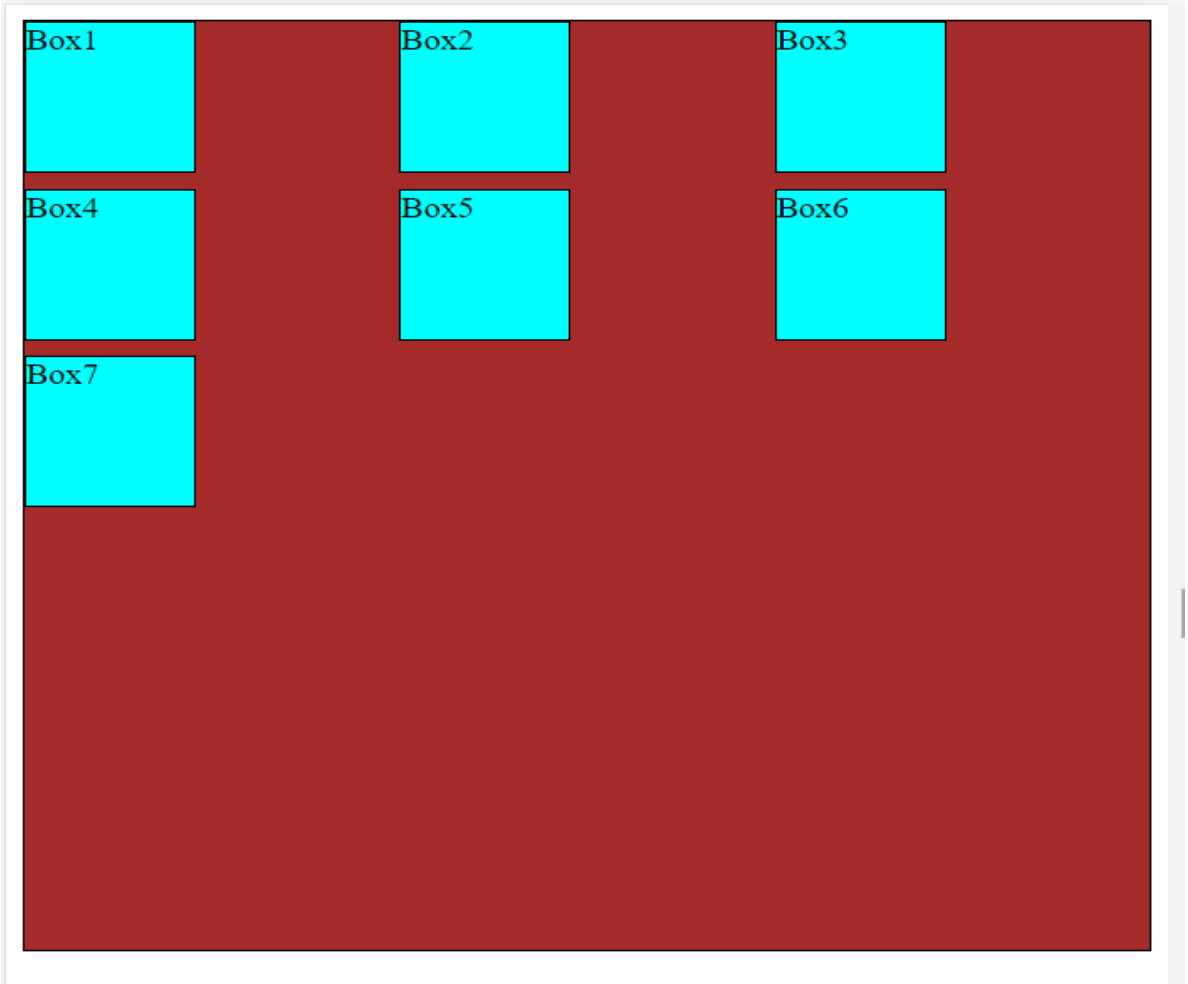
How to create flexible Grid?

We can use flexible unit “fr” to create flexible grid. It is fractional unit and **1fr means 1 part of available space**

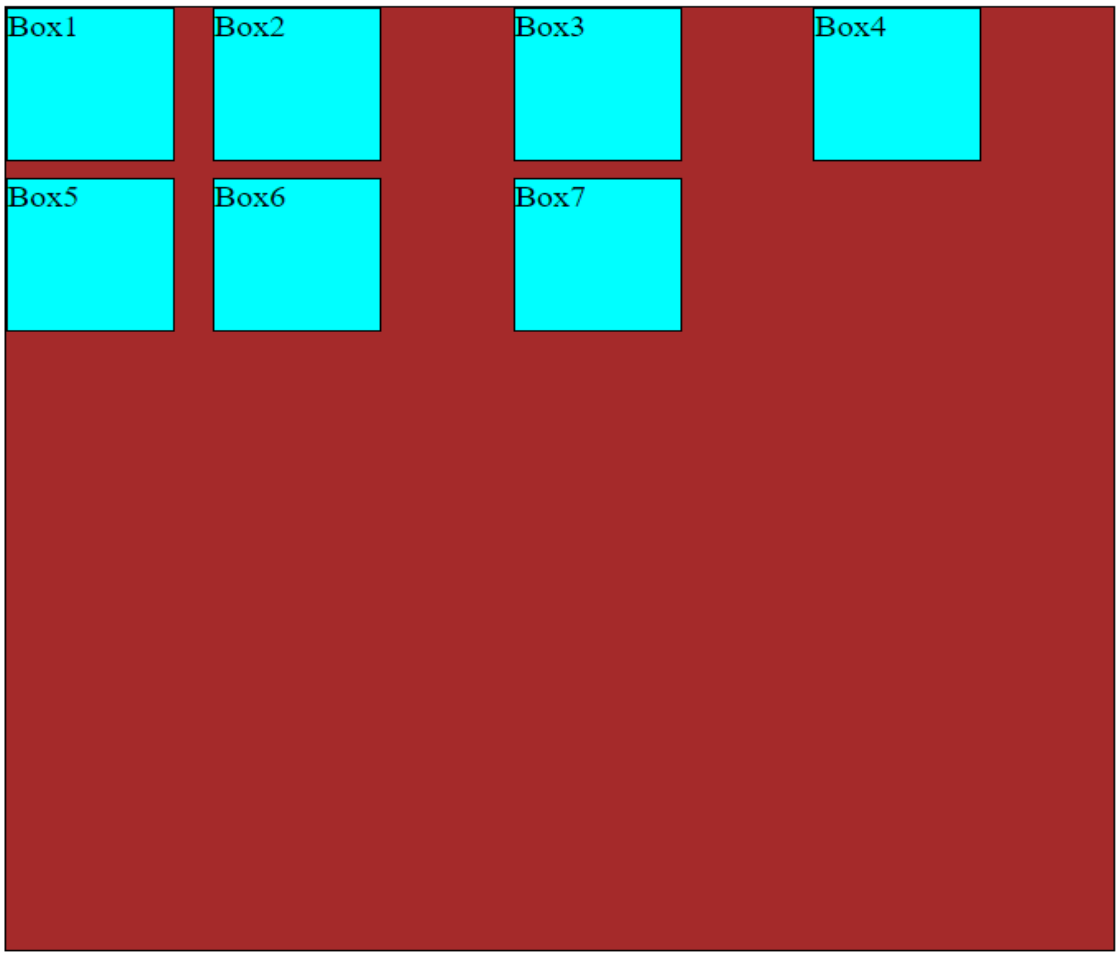
```
grid-template-columns: 1fr 1fr 1fr; //width will divide in 3 equal columns
```

OR

```
grid-template-columns: repeat(3, 1fr); //width will divide in 3 equal columns
```

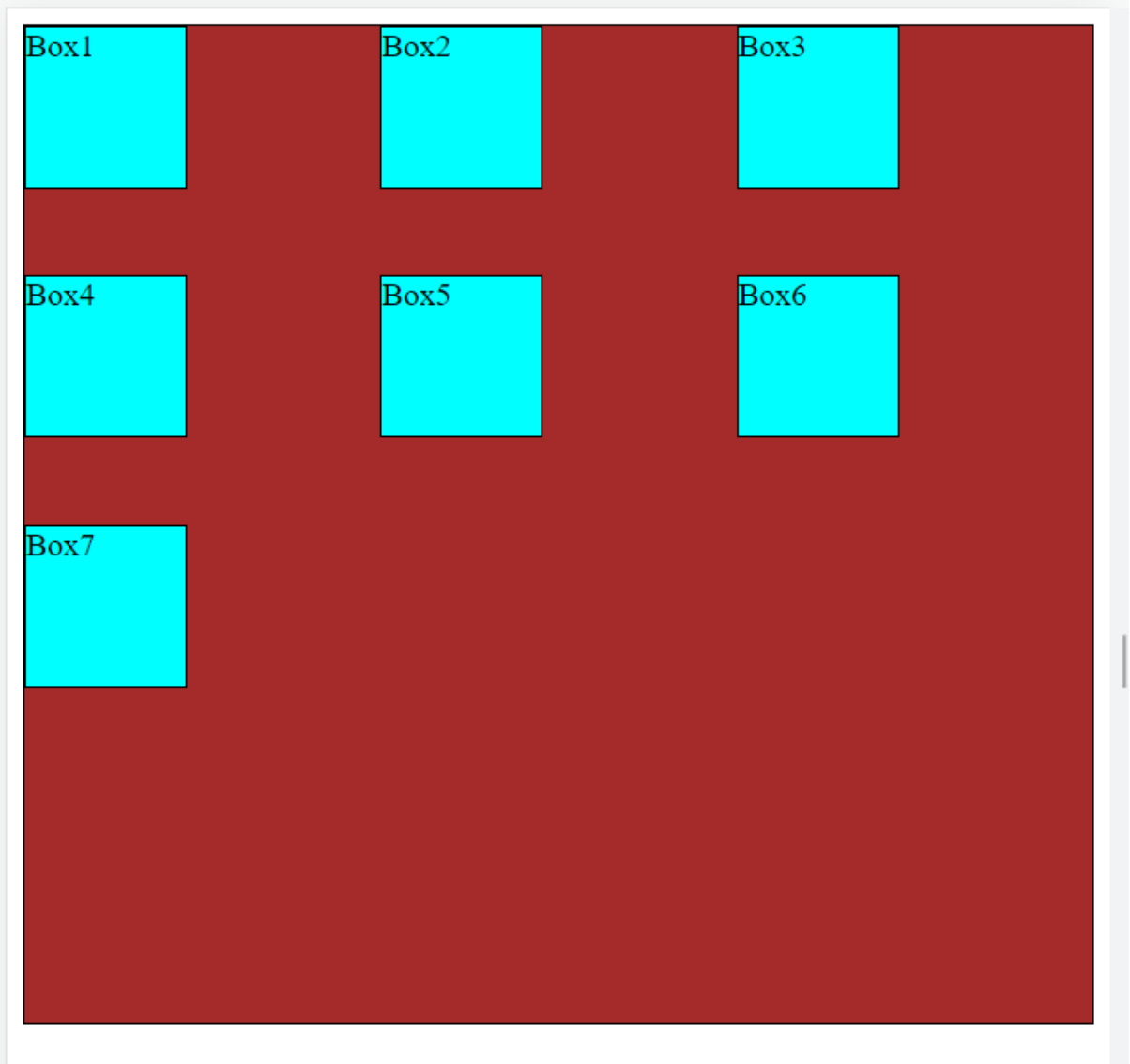

Box1
Box2
Box3
Box4
Box5
Box6
Box7

```
grid-template-columns: 100px repeat(3, 1fr); //first column will be of 100px and other will be divided in equal parts
```




```
grid-template-columns: 100px repeat(2, 1fr) 90px; //first col will be 100px, last col 90px and 2 columns middle having same width divided
```

```
 grid-template-columns: repeat(3, 1fr); //three cols having same width
 grid-template-rows: repeat(4, 1fr); //four rows having same width
```



gap – adds gap between rows and columns

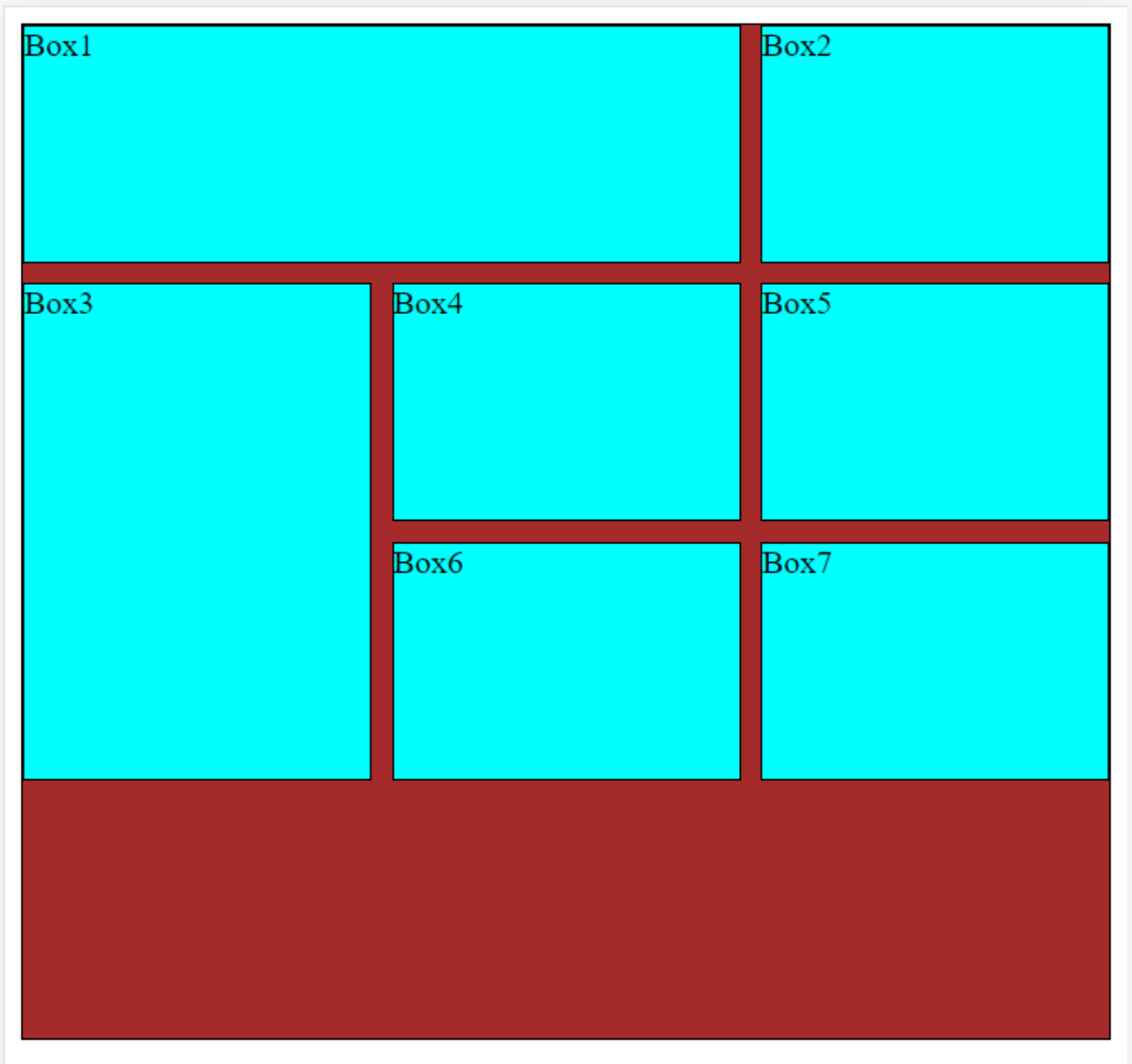
```
gap :10px; // add 10px gap
```

Grid Item Properties

```
#box1{
    grid-column-start: 1;
    grid-column-end: 3 ;
}
Note : Make sure you have set width explicitly for box
```

In above case, In case of overflow of number of boxes, grid layout adjust their height and add new row and items on that new row.

```
#box3{
    grid-row-start: 2;
    grid-row-end: 4;
}
//Make sure you haven’t set height explicitly
```

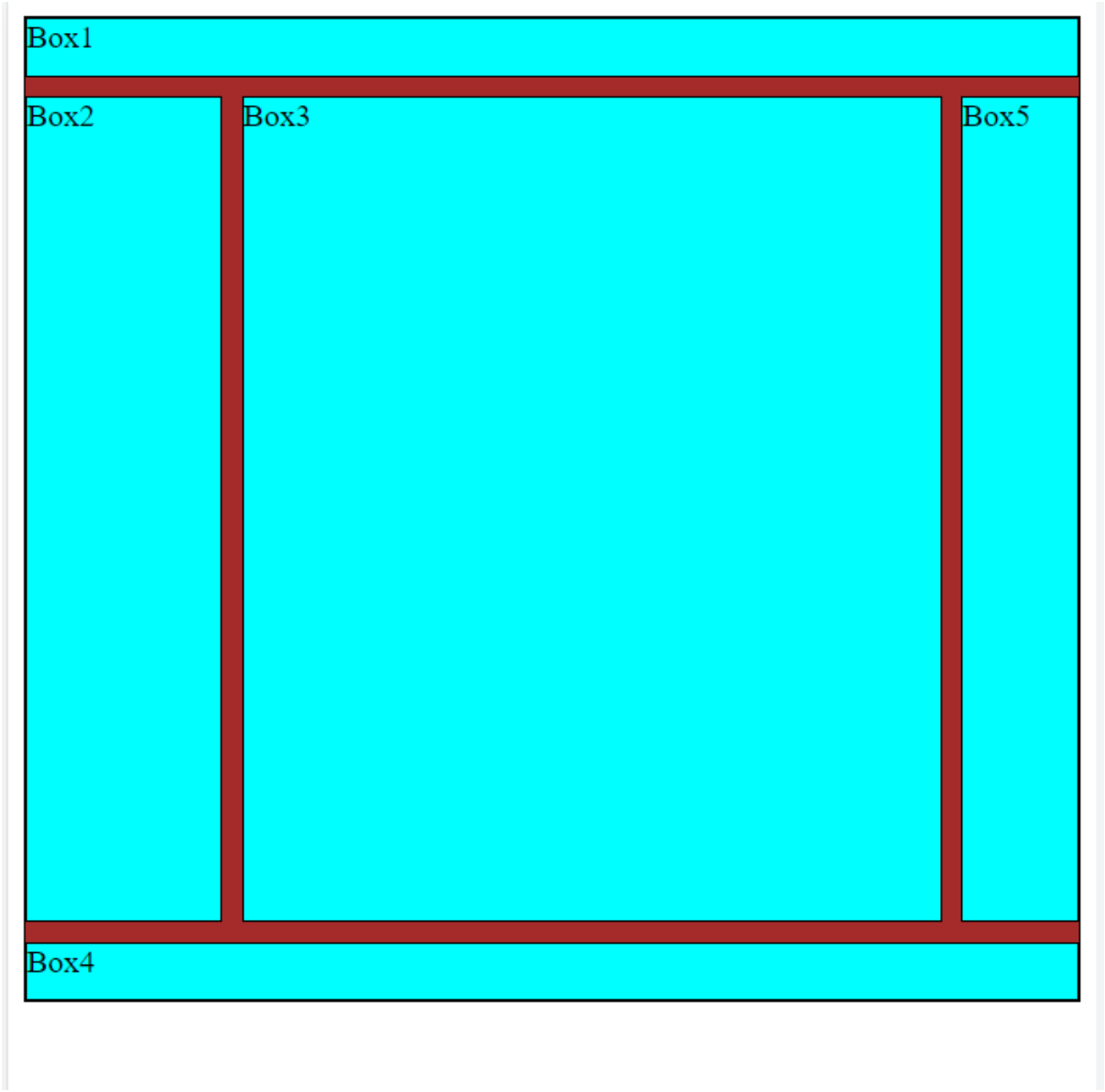


Creating Simple Layout

```
.container{
    display: grid;
    grid-template-columns: 100px repeat(2, 1fr) 60px;
    grid-template-rows: 30px 1fr 30px;
    gap :10px;
}
```

```
#box1, #box4{
    grid-column-start: 1;
    grid-column-end: 5;
}
#box2{
    grid-row-start: 2;
    grid-row-end: 3;
}
```

```
#box3{
    grid-column-start: 2;
    grid-column-end: 4;
    grid-row-start: 2;
    grid-row-end: 3;
}
```



Shorthand Properties:

1) **grid-row : grid-row-start / grid row end ;**
   It is shorthand property for grid-row-start and grid-row-end.
   Grid-row-start specifies on which row to start displaying the item.
   Grid-row-end specifies on which row line to stop displaying the item, or how many rows to span

Example:

```
 .container{
      grid-template-rows: repeat(4, 1fr); //creating 4 rows
      grid-template-columns: repeat(3, 1fr); //creating 2 cols
      gap: 5px;
}
```

```
#box1{
    /* box1 will start from row2 and will span across 3 rows */
    grid-row: 2 / span 3;
}
```

2) **grid-column- grid-column-start / grid-column-end;**
It is shorthand property for grid-column start and grid-column-end.
Grid-column-start specifies on which column to start displaying an item.
Grid-column-end specifies on which column line to stop displaying an item or how many columns to span.

```
#box2{
    /*box2 will start from column1 and will span across 2 columns */
    grid-column: 1 / span 2;
}
```