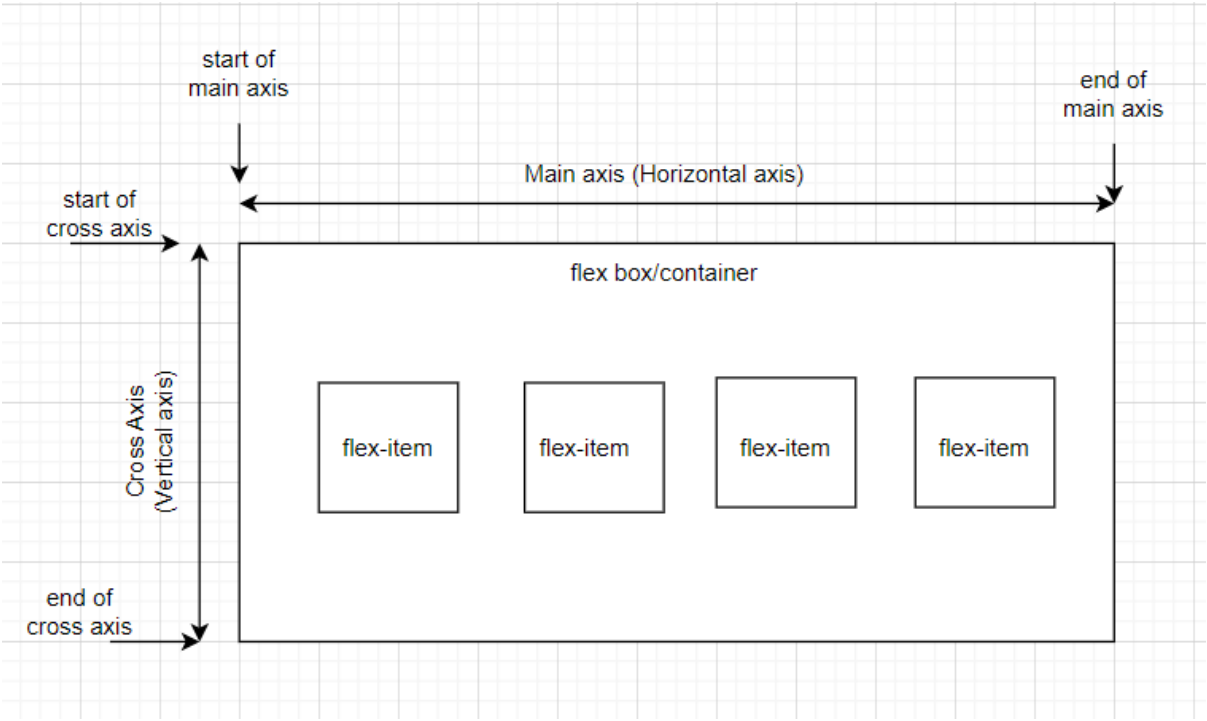
CSS FlexBox

- It is 1D layout model use for **space distribution** and **alignment capabilities.**
- It is used to get more flexibility in your layout and simplify responsive layout design
- It makes it easy to align elements in 2d plane
- Two main components of flexbox are -
    - **Flex Container** – div/container which we want to convert in flexbox using display : flex;
    - **Flex Items** - direct children/immediate elements in container



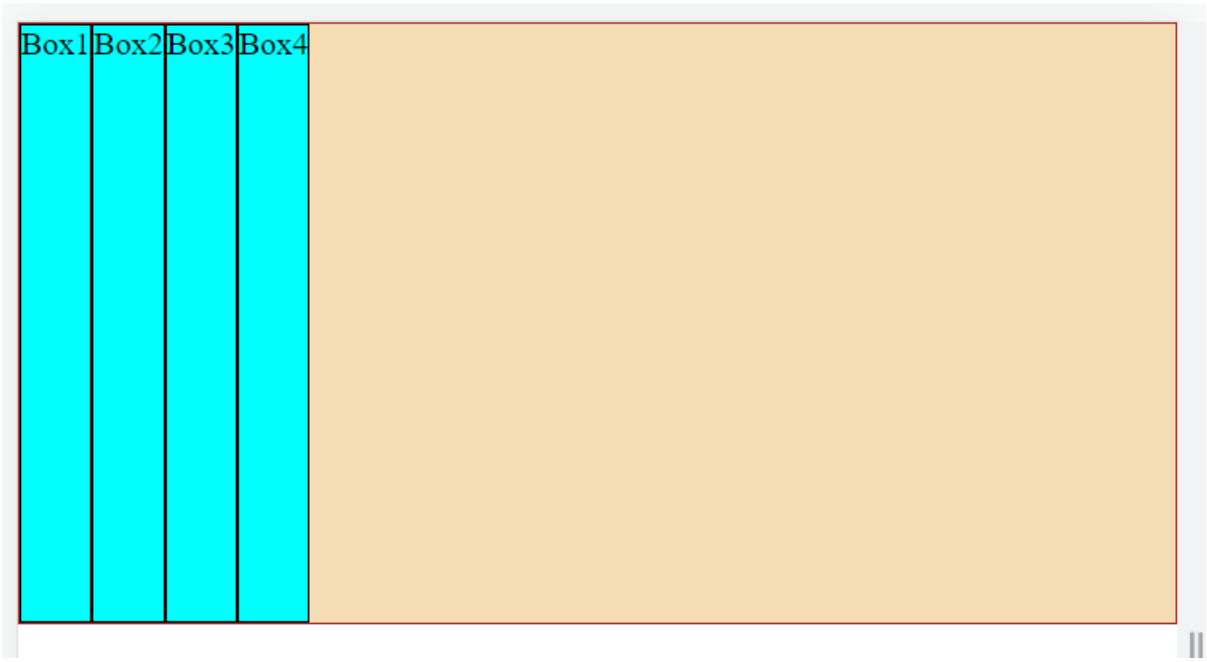
Flex-Container Properties:

Flex container becomes flexible by setting -

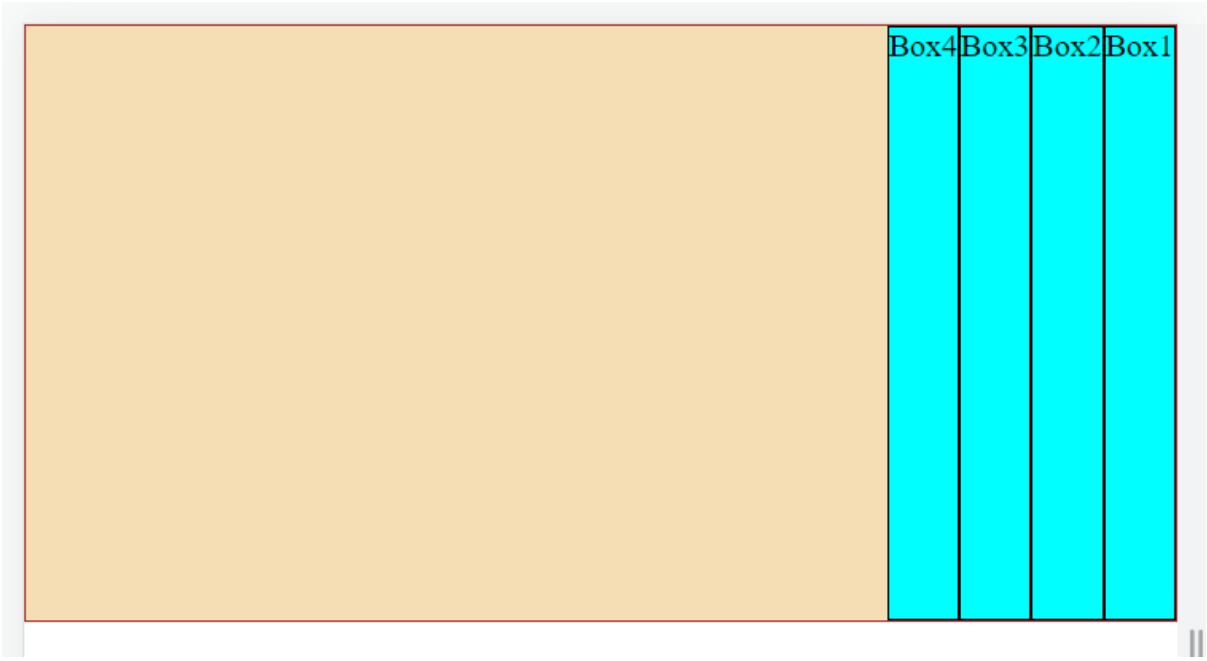
```
.container{
    display: flex;
}
```

flex-direction - change the direction of main axis

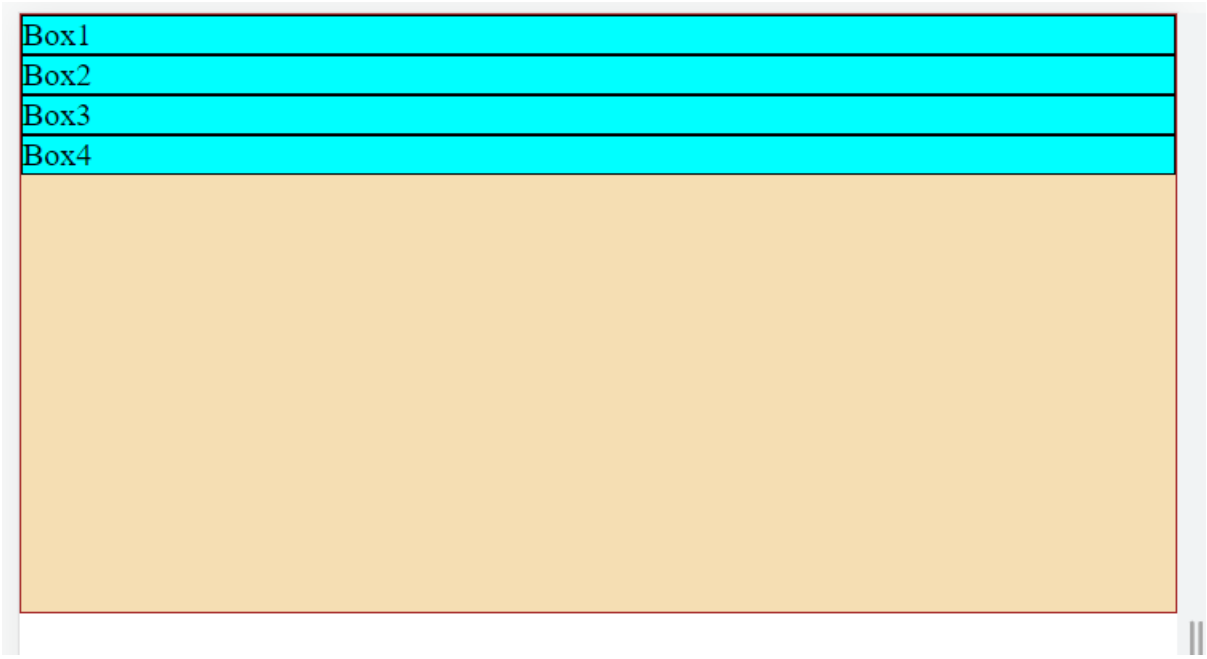
```
flex-direction: row;  // This is default value
```



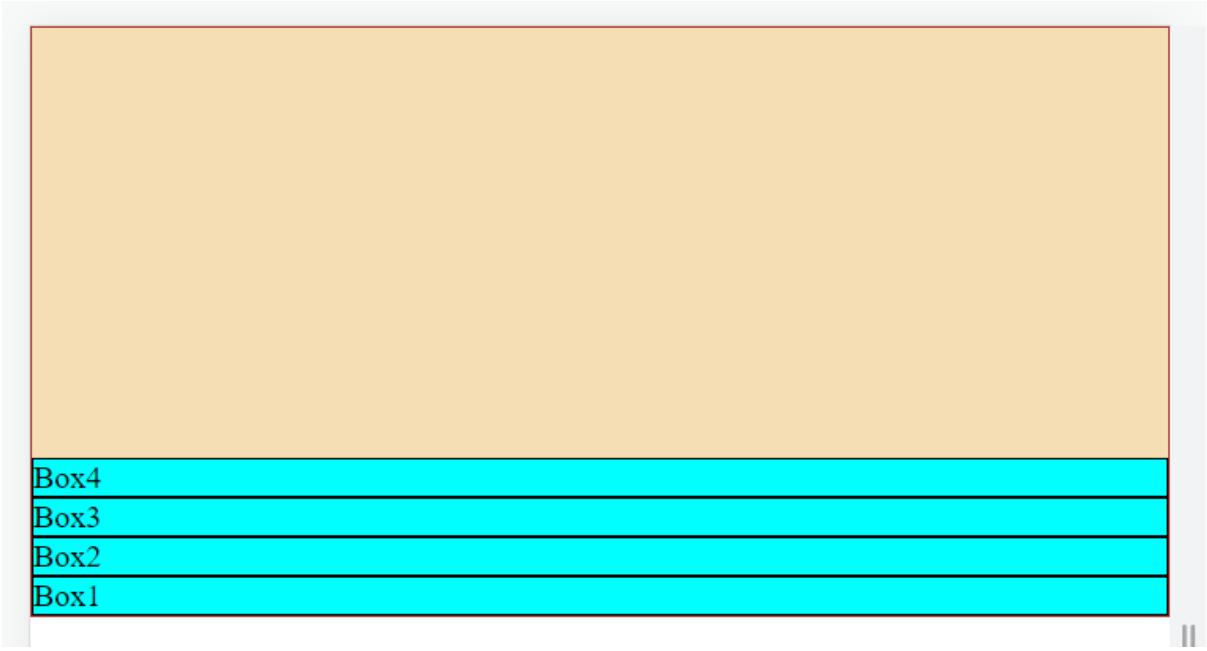

```
  flex-direction: row-reverse;
```



```
flex-direction: column;
```




```
flex-direction: column-reverse;
```



flex-direction: row is default, when we reduce the width of viewport then width of flex items get changes and becomes less.

What if we want to have fixed width?

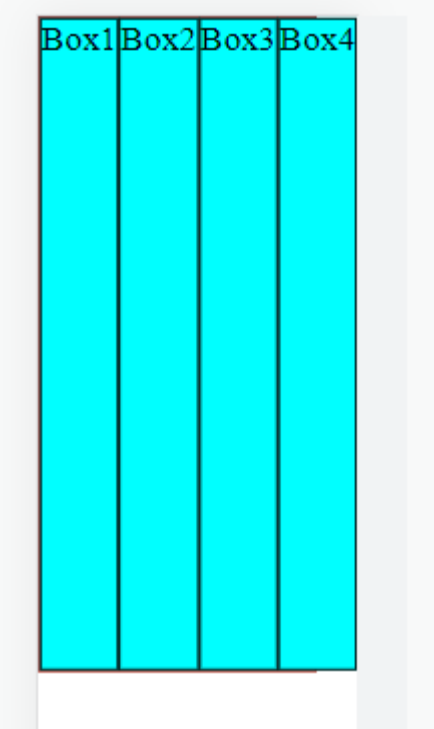
We can set the fix width for flex items

```
.box{
    border: 1px solid black;
```

```
    background-color: aqua;
    width: 50px;
}
```

Oh, but still it squeezes when we reduce the size of viewport, why?

```
  flex-wrap: nowrap; // this is default value for flexwrap
```



What can we do to keep width intact?

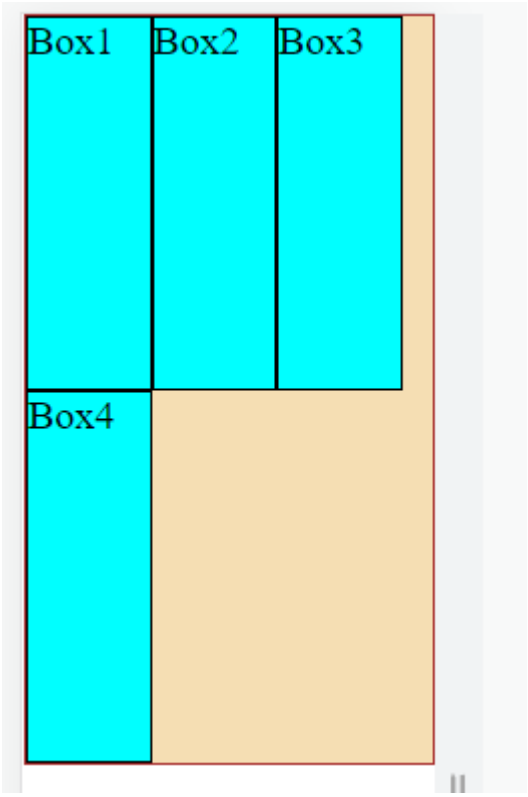
We can use

```
 flex-wrap:wrap;
```

By setting this property, flex items moves on next line when we reduce the size of viewport.

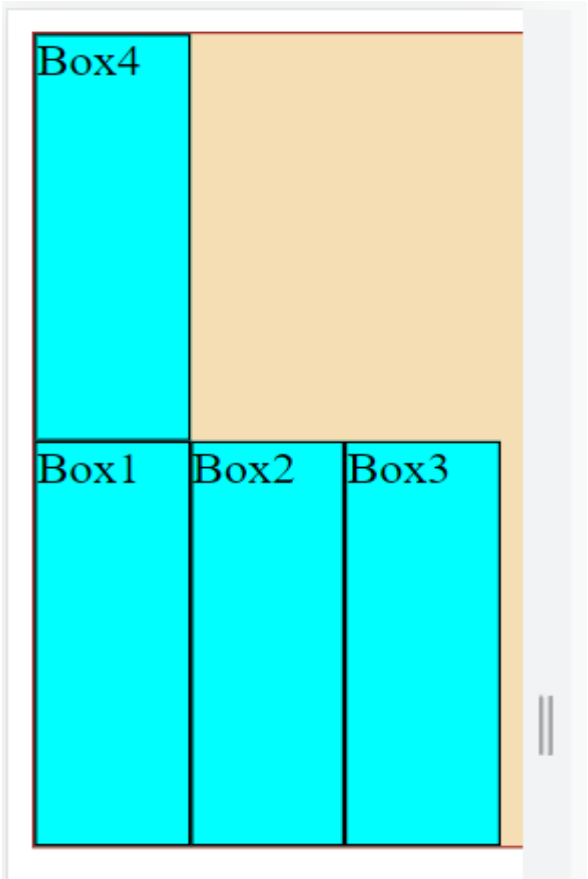
It tries to keeps the same width (which we had defined for flex-items) as much as it can

When all elements places on one line and then also we reduce the size of viewport then width will reduce!



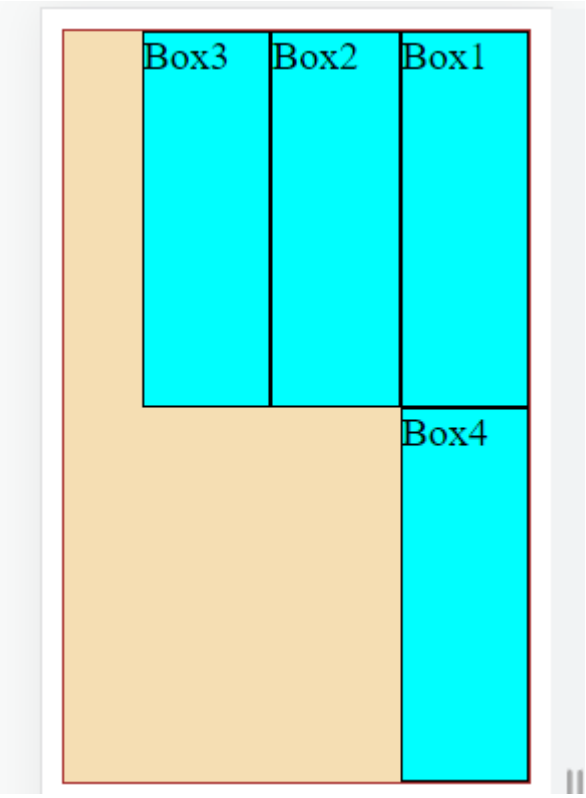

```
flex-wrap: wrap-reverse;
```

Instead of pushing the element on next line, it pushes the fourth item on previous (above) line



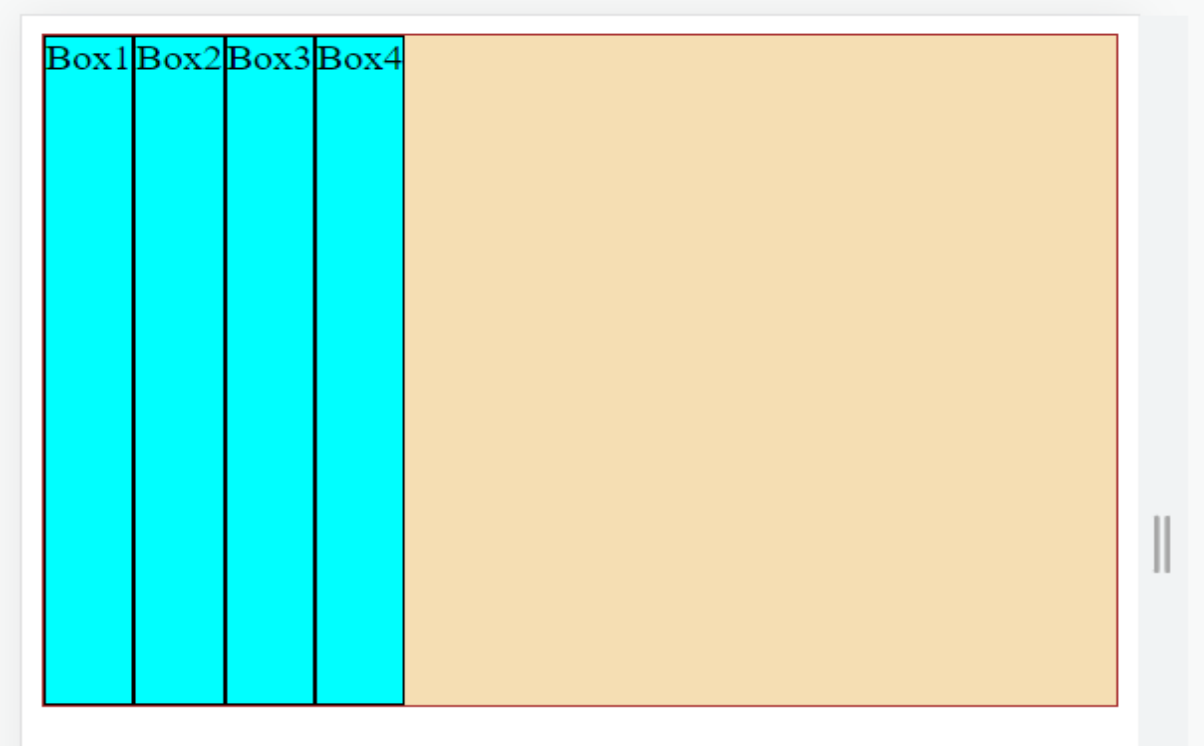
flex-flow - shorthand notation for flex-direction and flex wrap

flex-flow: row wrap

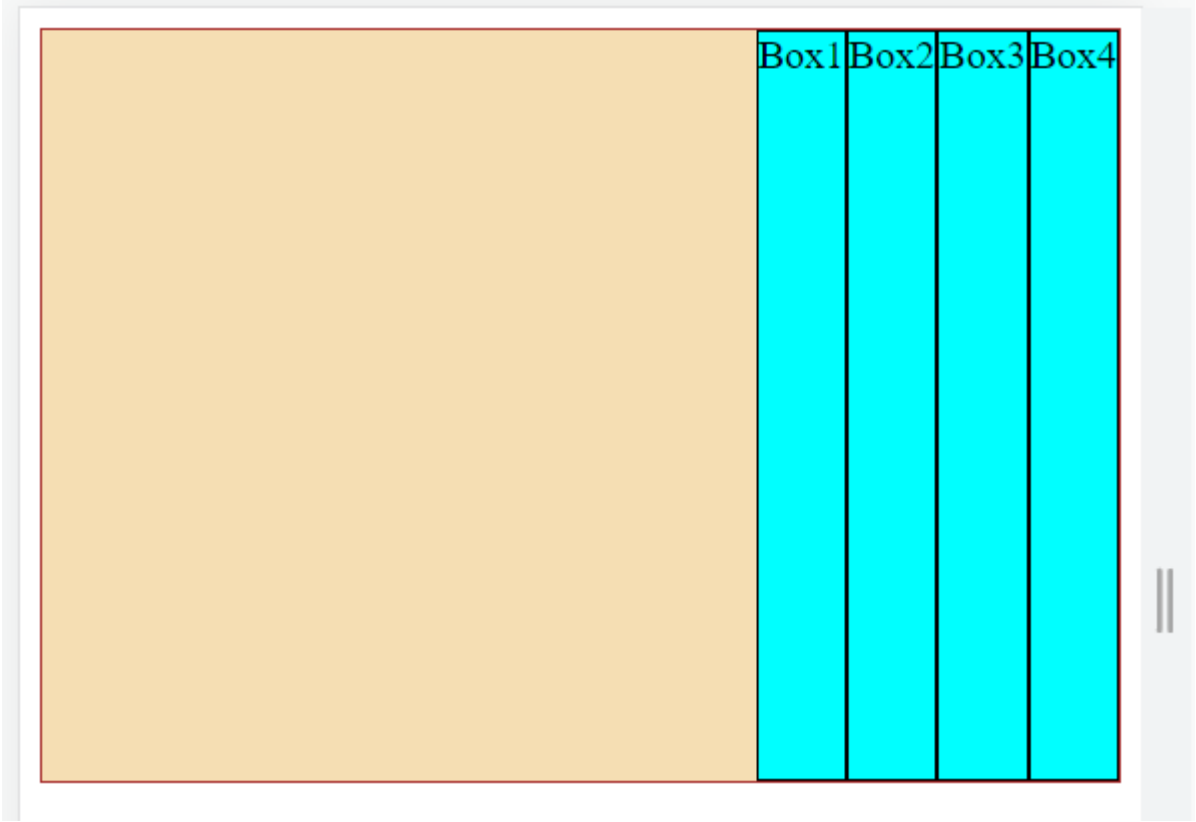


Justify-content - align the elements/items according to main axis (horizontal axis)

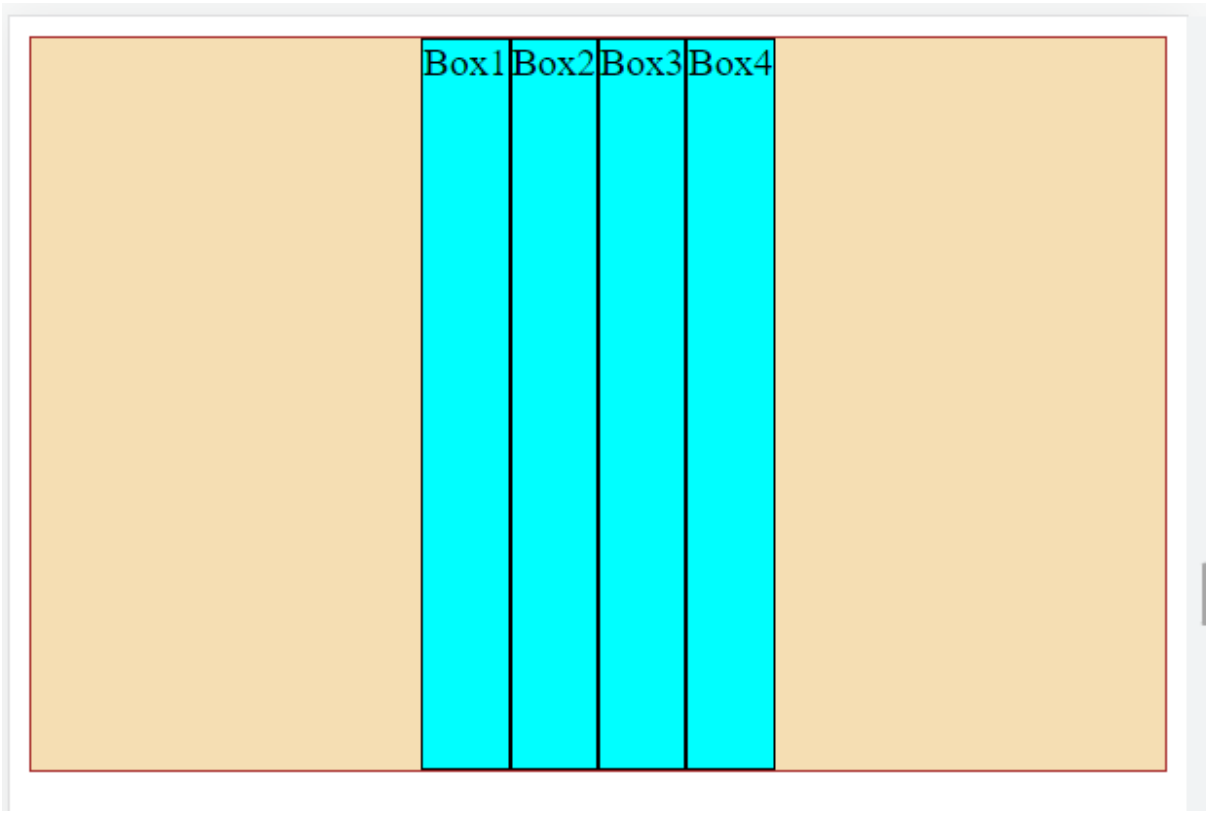
```
justify-content: flex-start; //all items will be placed at start of main-axis
```



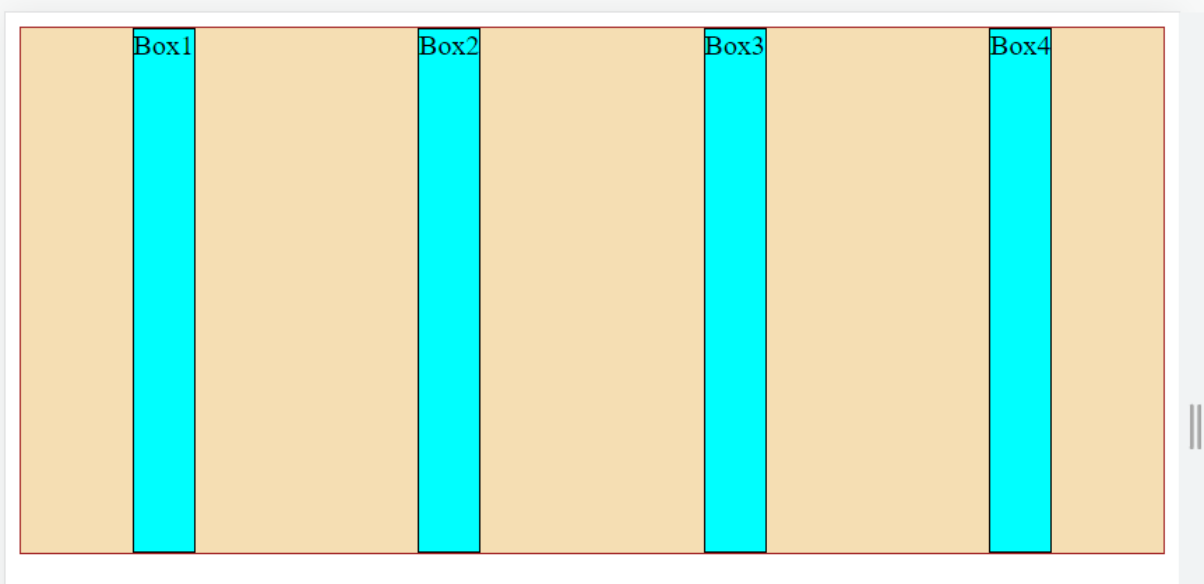
```
justify-content: flex-end; //all items will be placed at the end of main axis
```



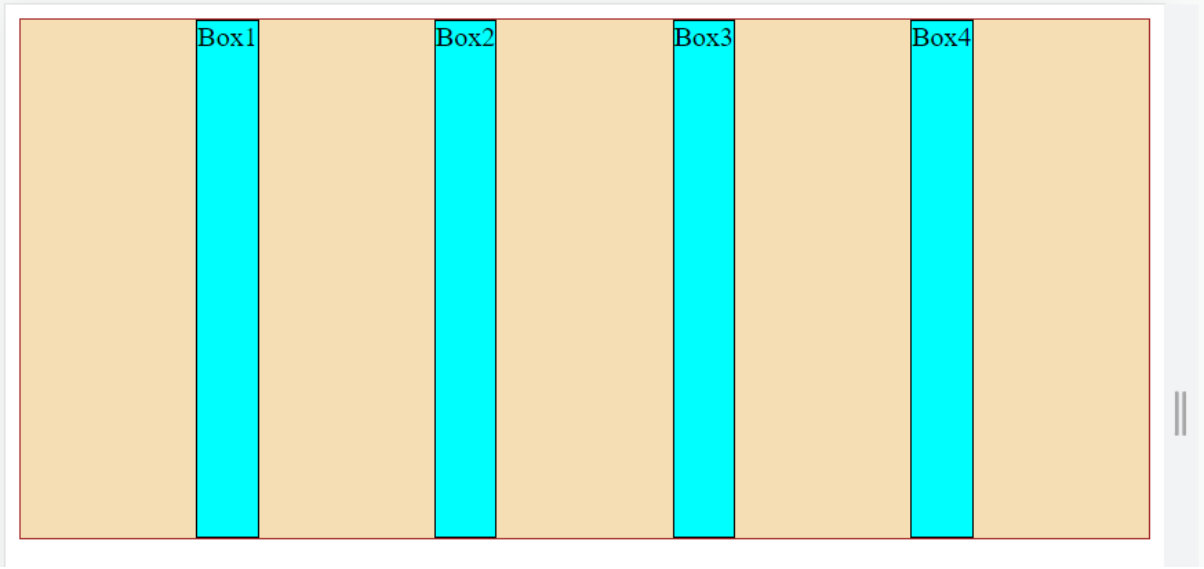

```
justify-content: center; //all items will be placed at center of main axis
```



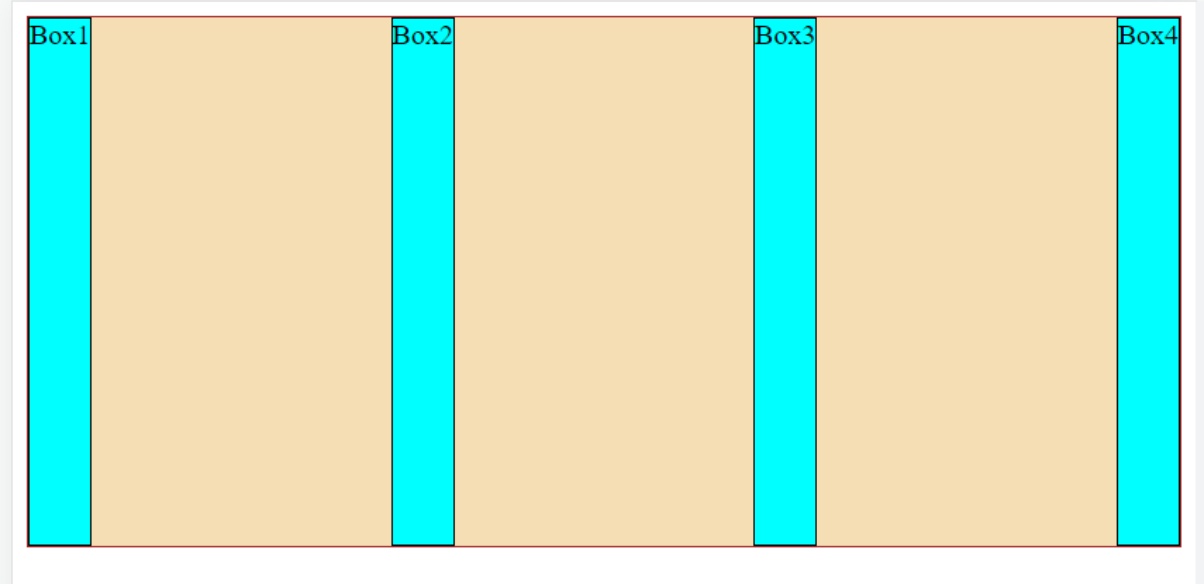
```
justify-content: space-around;
```
It applies the same spacing **in between** all the items (left side of first item and right side of last item don't have same spacing)




```
justify-content: space-evenly;
```
It applies same spacing in between all items and before first item and after last items also.




```
justify-content: space-between;
```
It applies equal same **in between items only** (first and last items are kept touched to left and right sides) as shown below;



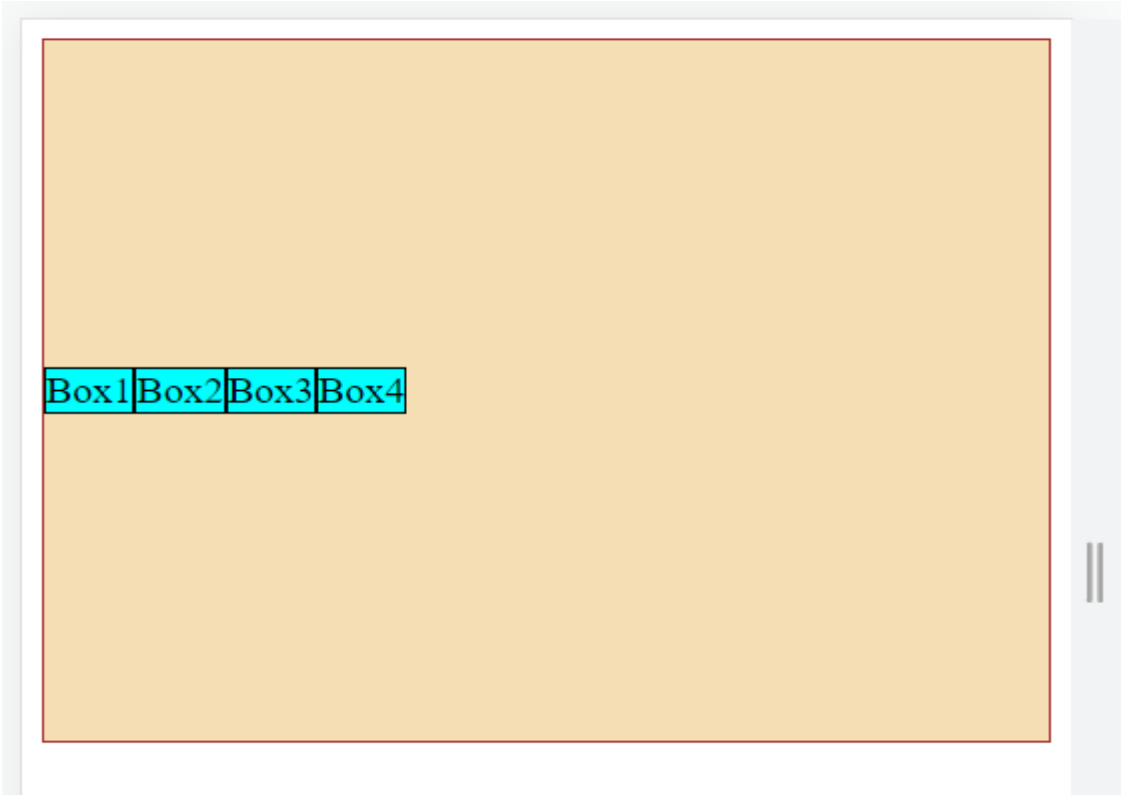
align-items - align the items on cross axis (vertical alignment)

```
align-items: flex-start;
```

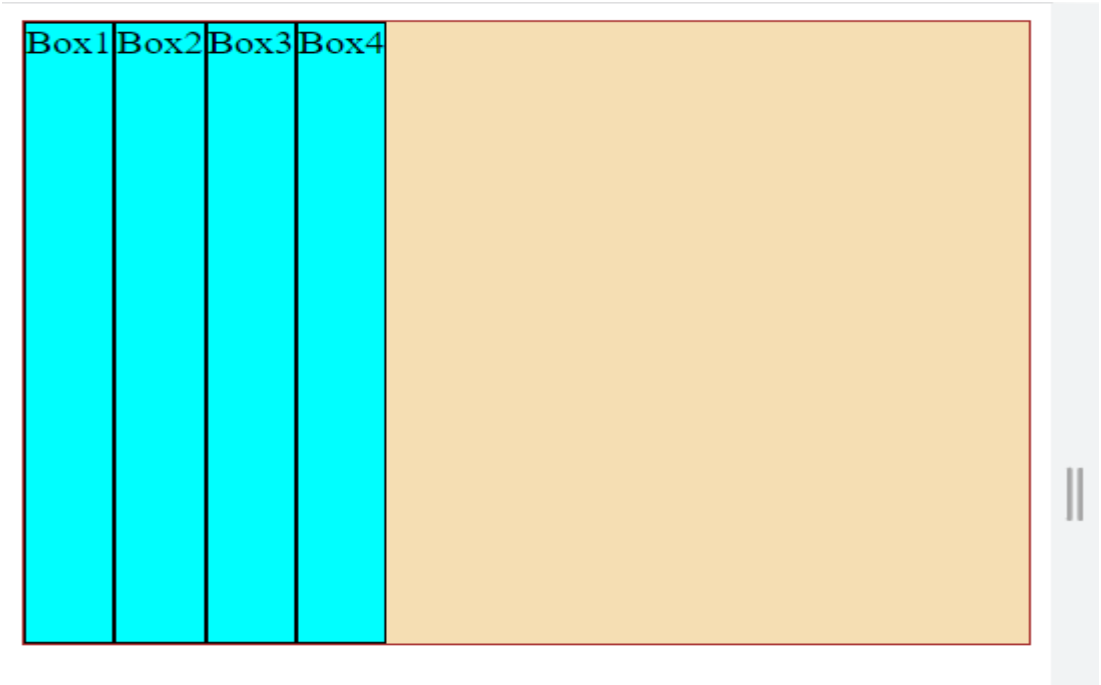
All items will align at the start of cross axis (i.e. top-left corner)

```
align-items: flex-end;
```
All items will align at the end of cross axis (i.e. bottom-left corner)

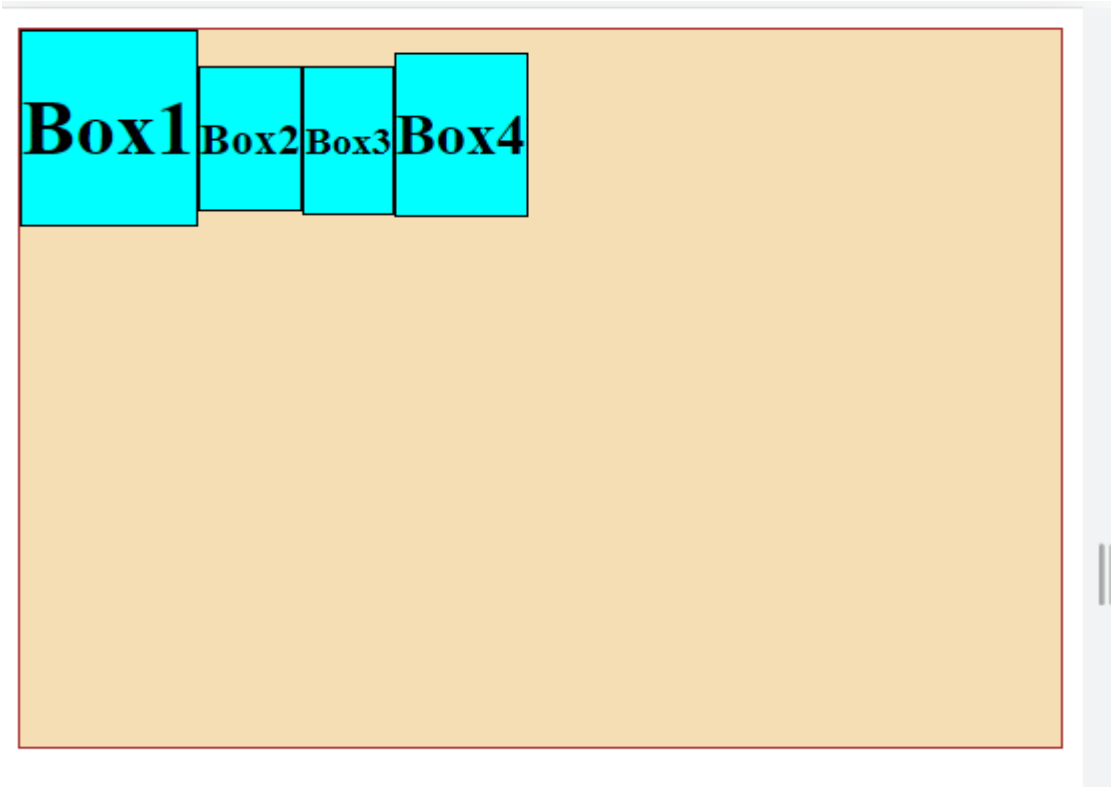

```
align-items: center;
```
All items will align at the center of cross axis



```
align-items: stretch;
```
Stretch the items from top to bottom of cross axis (i.e. vertically) - this is default value for flex container.




```
align-items: baseline; //align all the items according to the baseline of content
```



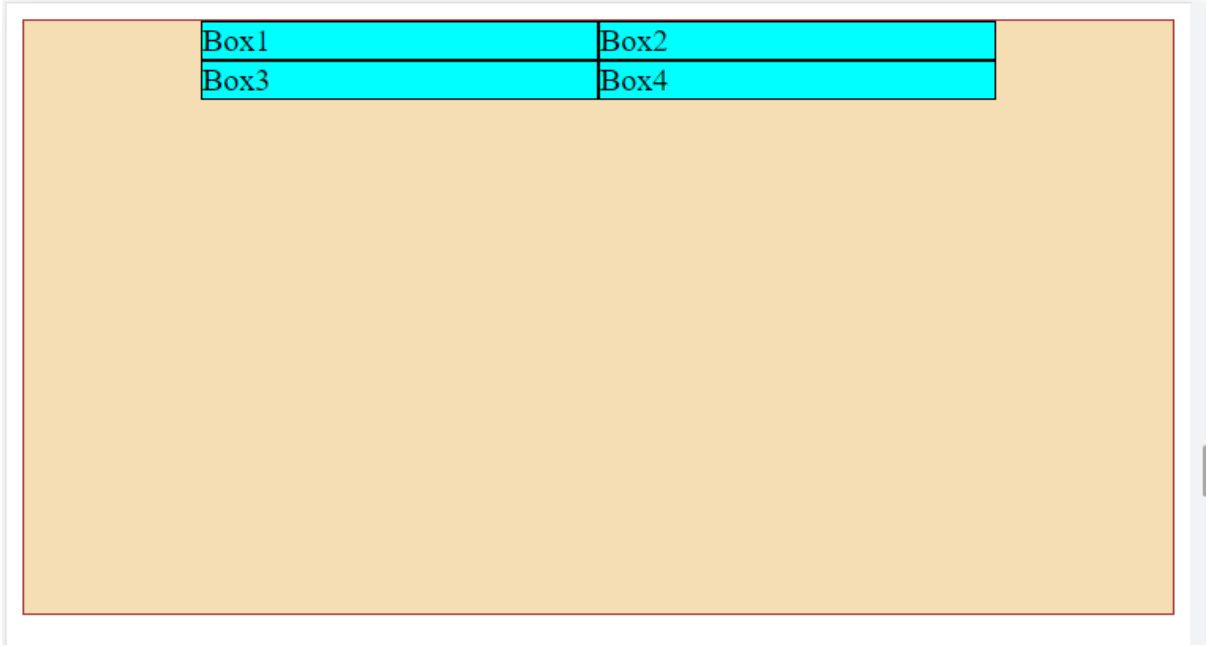
align-content - handle spacing between multiple rows.

To create rows:

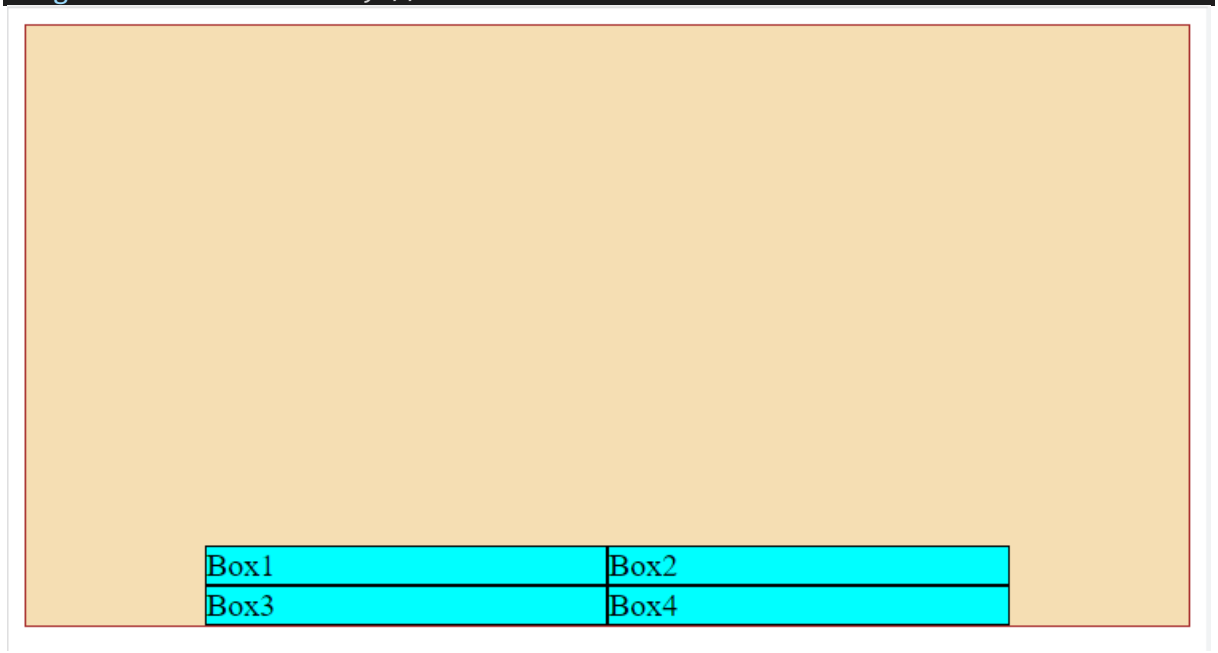
```
.container{
    height: 300px;
    border: 1px solid brown;
    background-color: wheat;
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    align-items: center;
}
```

```
.box{
    border: 1px solid black;
    background-color: aqua;
    width: 200px; //setting width for items
}
```

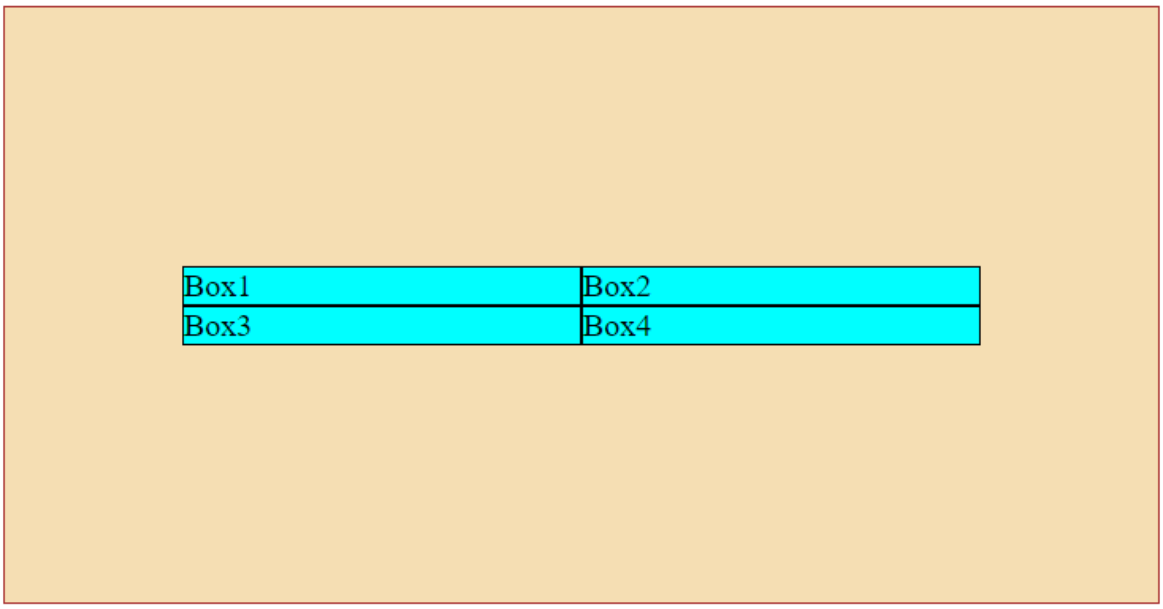
```
align-content: flex-start; //all rows will shift at the top  (of cross axis)
```



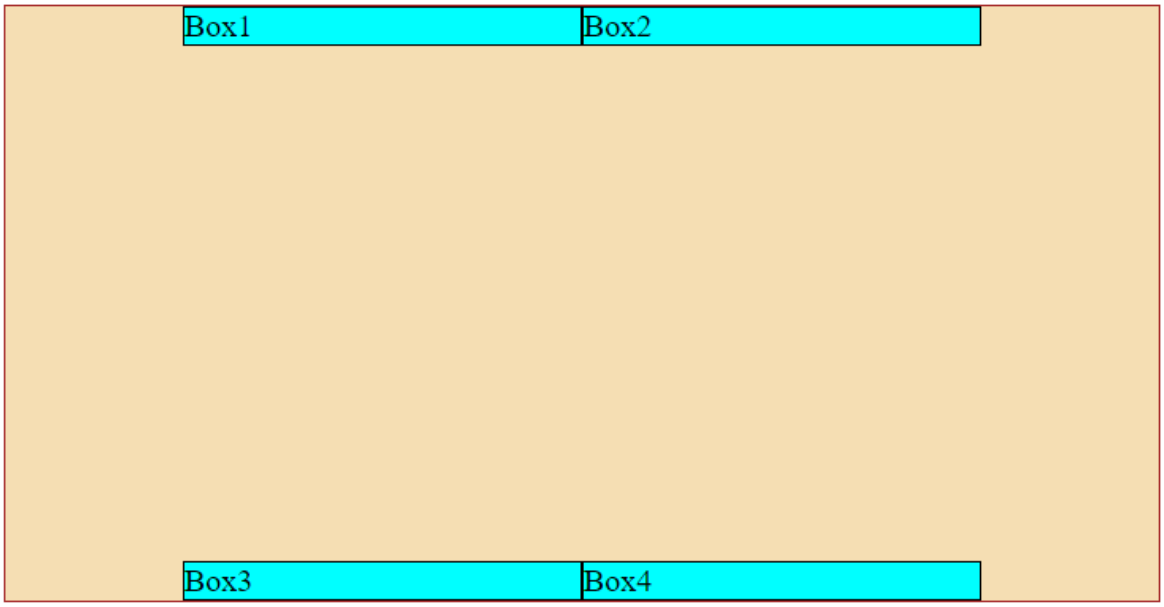
```
align-content: flex-end; //all rows will shift at bottom
```



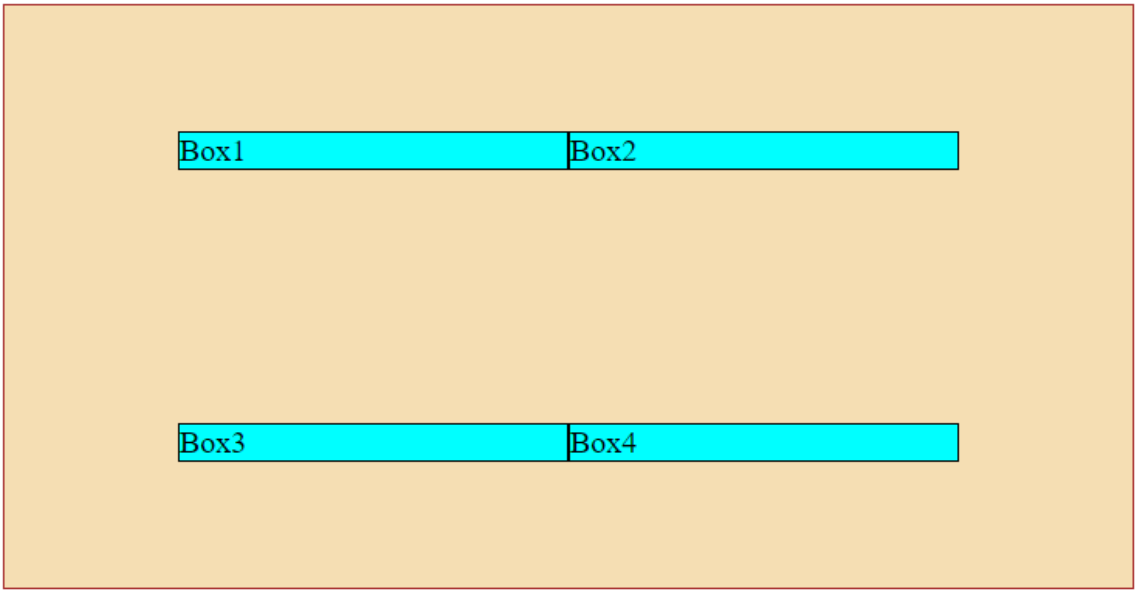

```
align-content: center; //all rows will be center
```



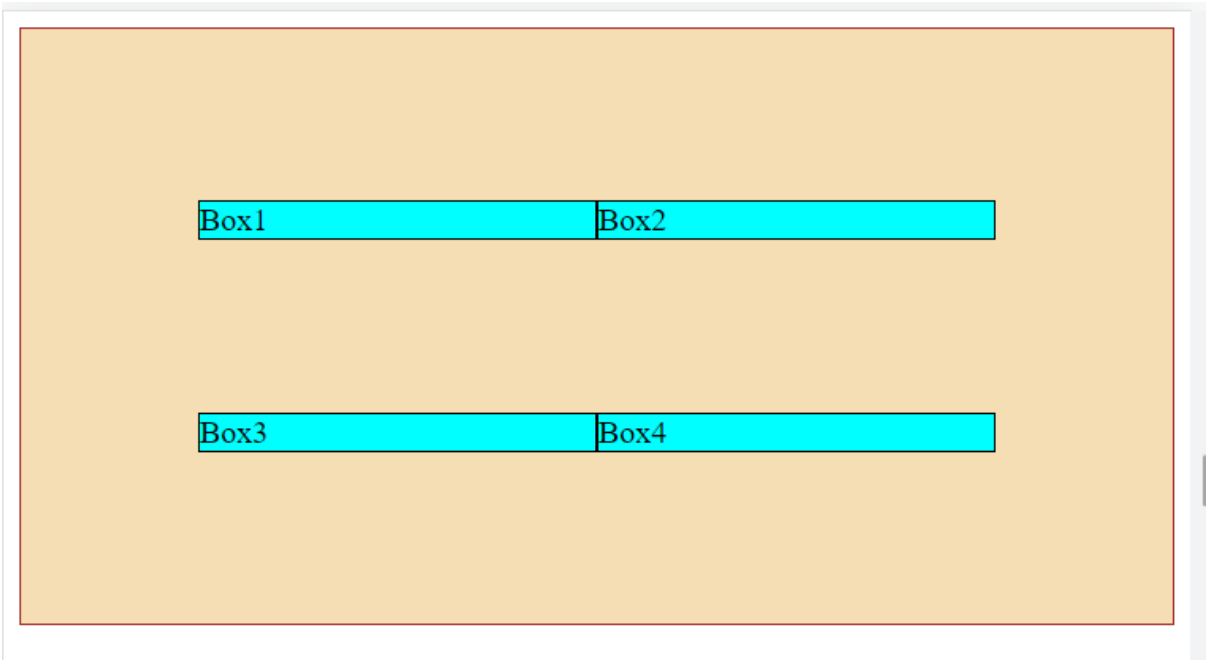
`align-content: space-between;` //one row top, other at bottom and complete space in between them



`align-content: space-around;` //add space between two rows, but not equal at top and bottom

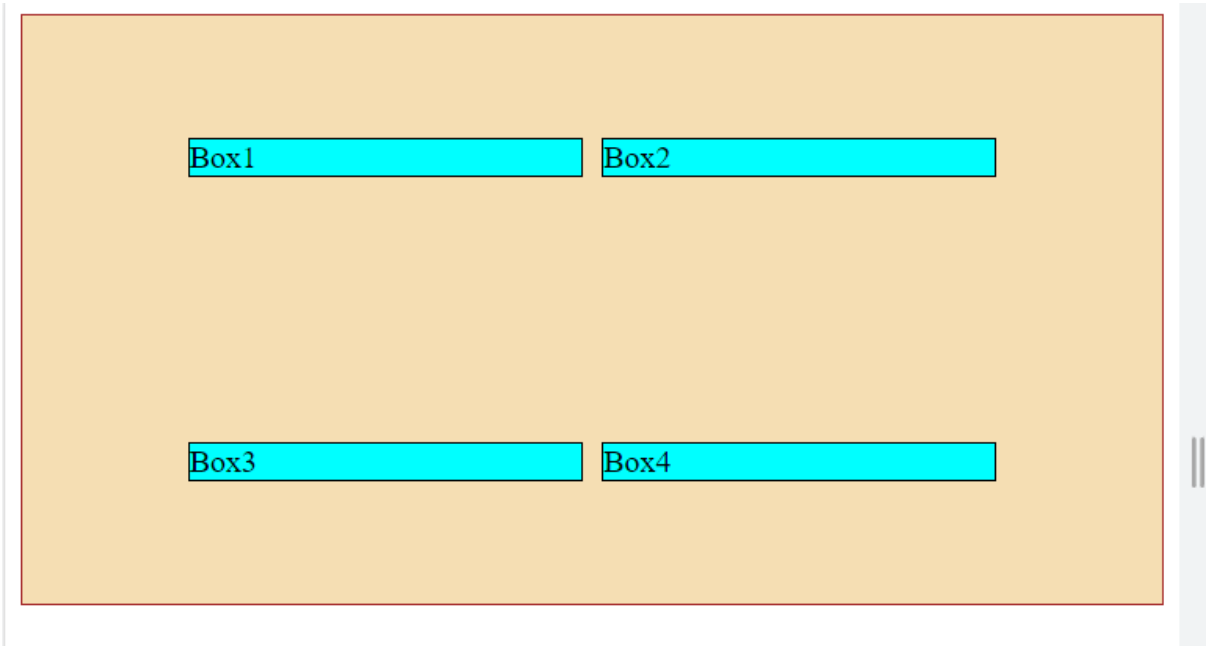


```
align-content: space-evenly; // same space from top, in between and bottom
```



gap - to add gap between items

```
gap: 10px;
```



It is shorthand notation as gap: row-gap, column-gap;

Flex-items Properties

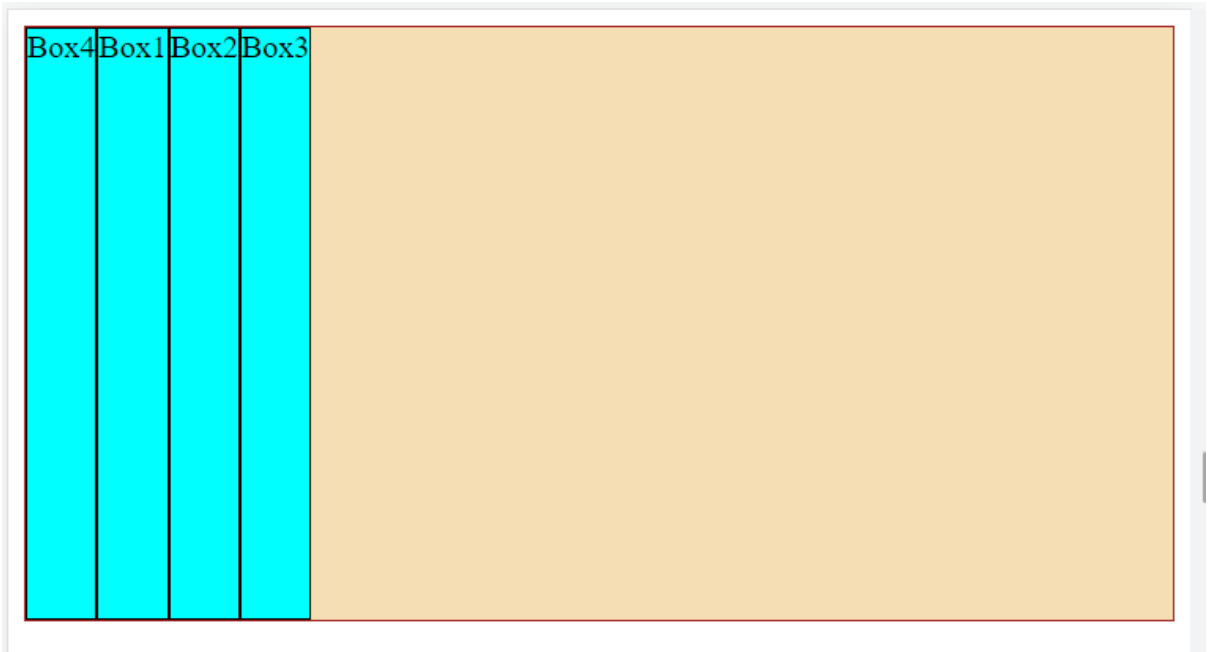
The direct child elements of flex container automatically becomes flexible (flex) items when we make container as flexbox by using display:flex;

order

- To changes the order of items like 2431
- By default the order of all elements is 0
- If we change order of any one element then it gets placed at last

```
#box1{
    order: 1;
}
#box2{
    order: 5;
}
#box3{
    order: 8;
}
#box4{

}
```



flex-grow I

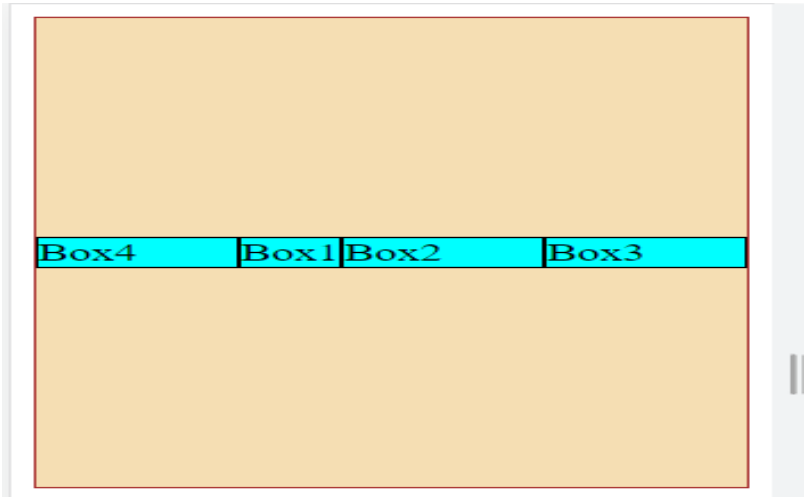
- Increase the size of items/particular item
- The default value is 0 for all items

```
#box4{
    flex-grow: 1;
}
```

flex-shrink

- Sets the shrinking speed of box
- The default value is 1 initially

```
#box1{
    order: 1;
    flex-shrink: 3; //then box1 will shrink more faster than other boxes
}
```



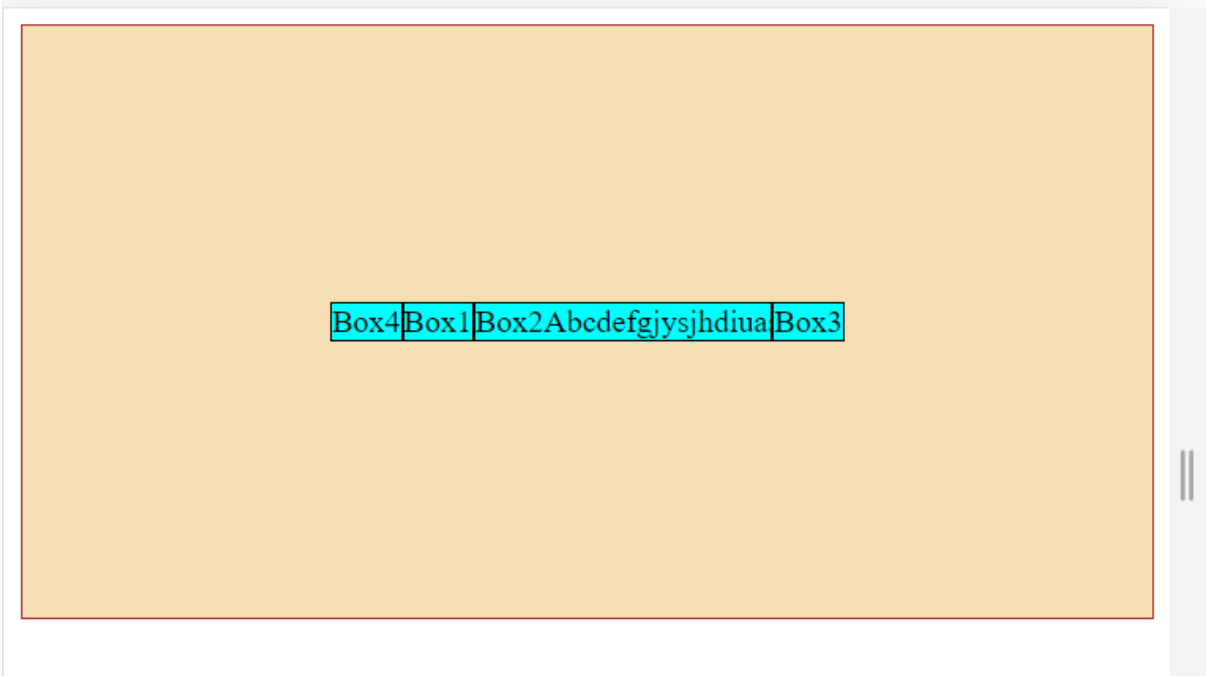
flex-basis - used to set the width of items

What is difference between flex-basis and width?

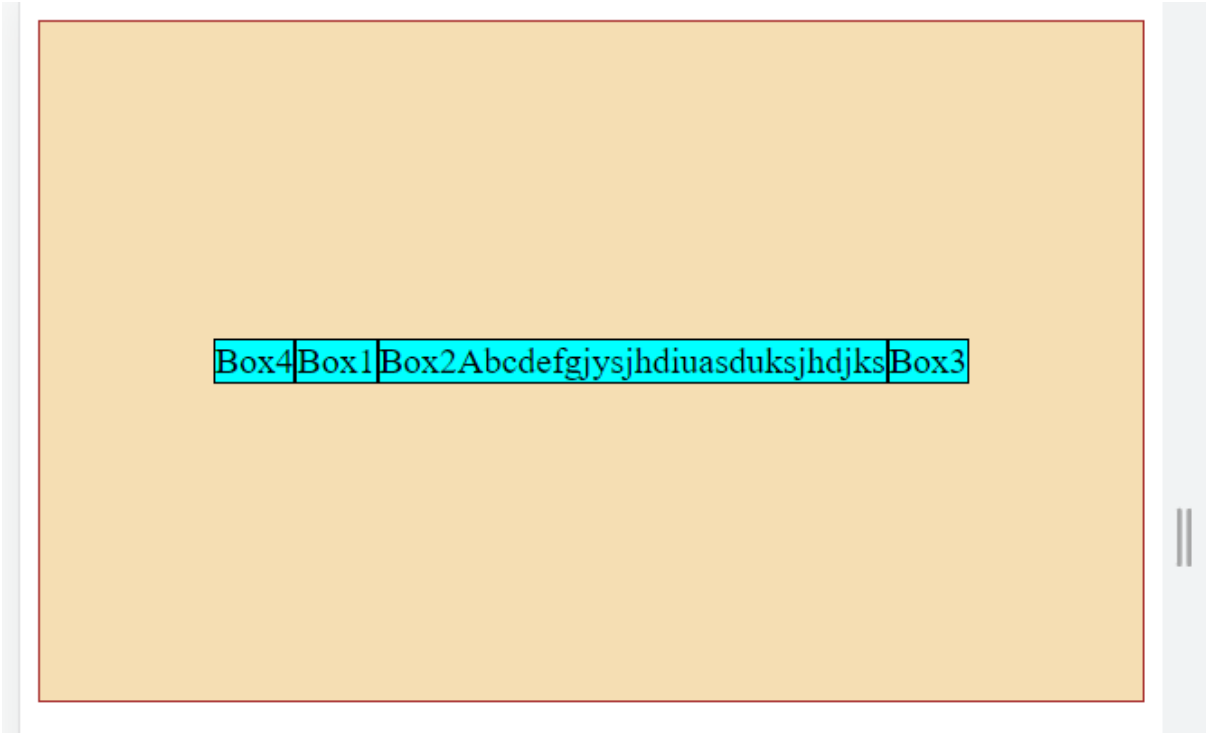
If we have set width for item and content is more than width then content gets hidden

If we have used flex basis to set the width and if content is more than width then it will increase the size of item to fit the content in. with flex-basis, website becomes responsive!

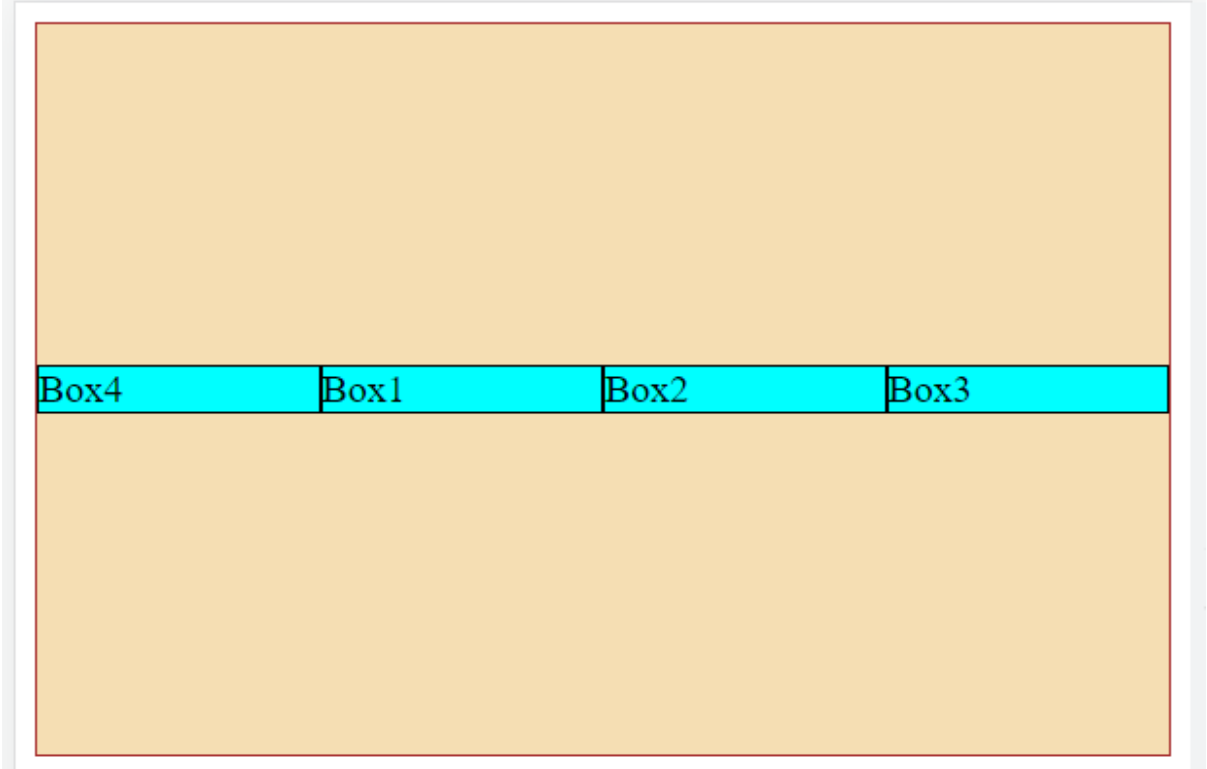
```
#box2{
    order: 5;
    /* flex-basis: 150px; */
    width: 150px;
}
```




```
#box2{
    order: 5;
    flex-basis: 150px;
}
```



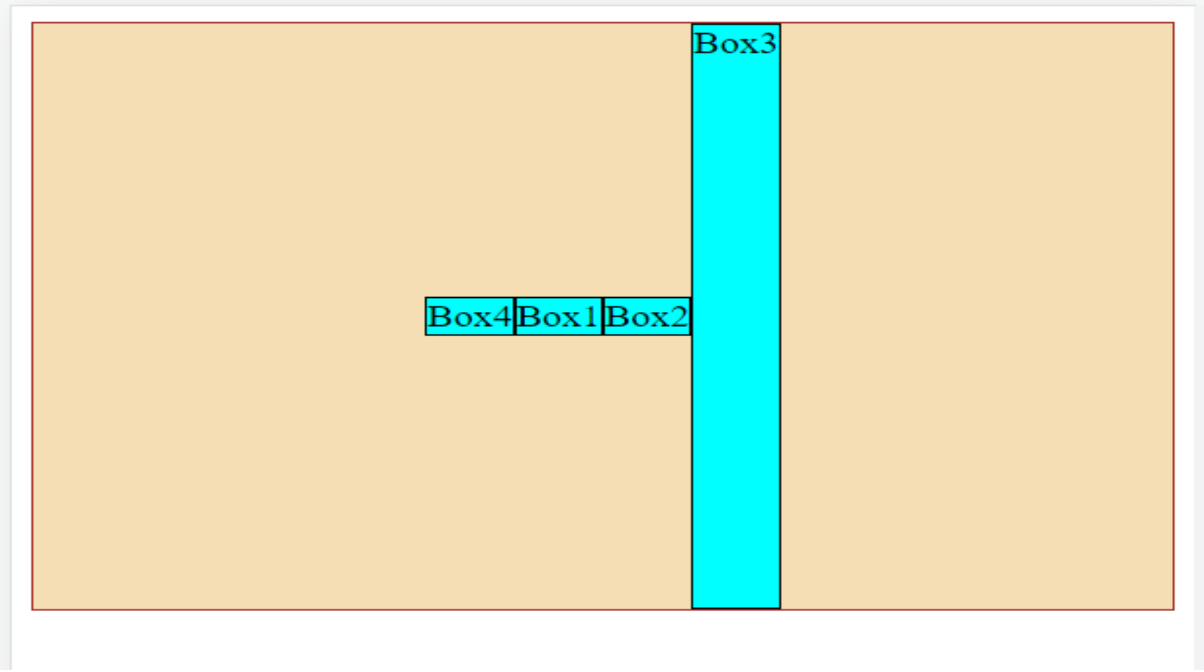

```
flex-basis: 50%; //one item will take 50% of width, here are 4 items so it takes 25% each
```



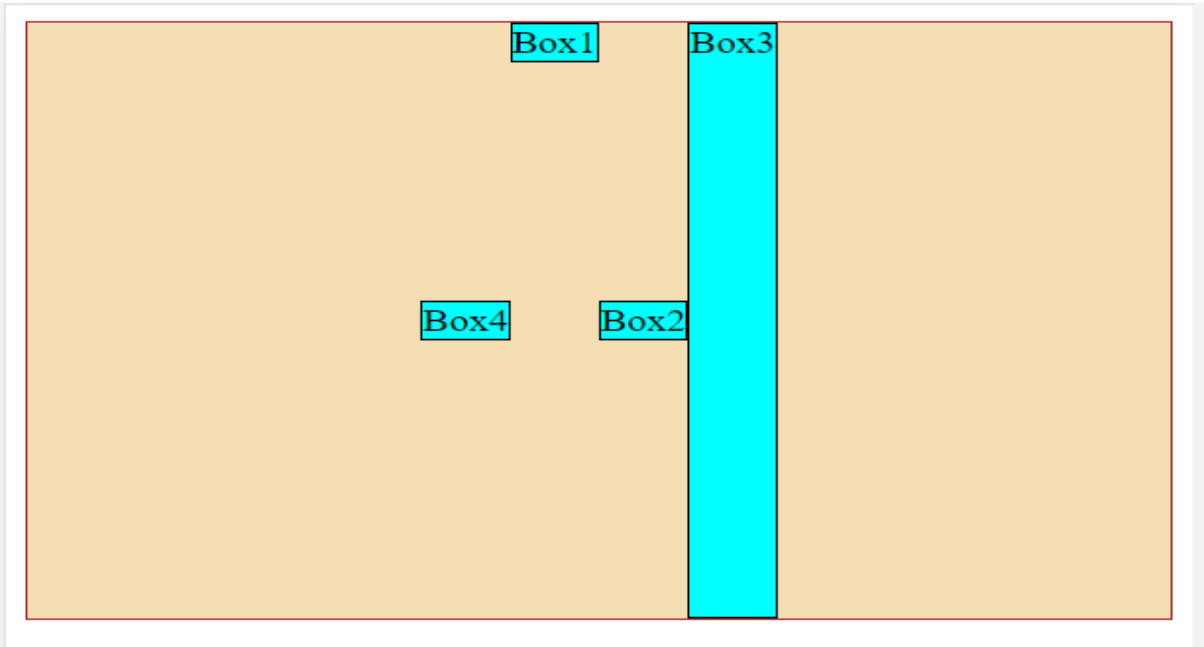
flex - shorthand notation for flex-grow, flex-shrink, flex-basis

align-self - to align particular element as stretch, start, end, center

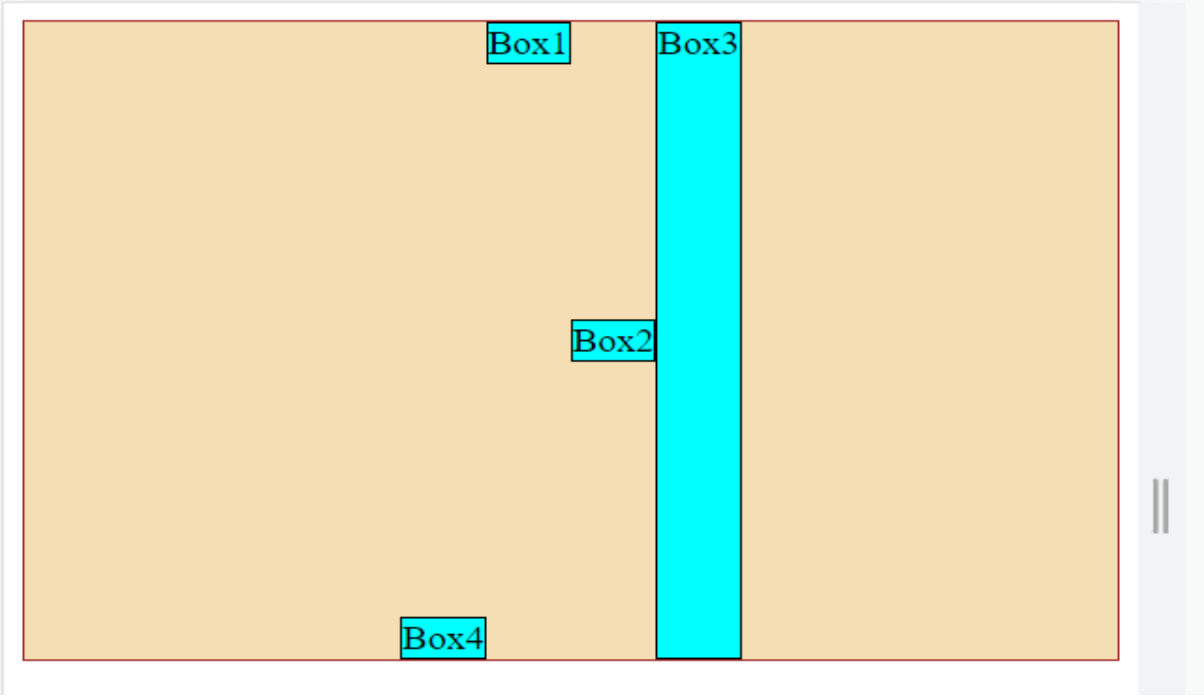
```
#box3{
    order: 8;
    align-self: stretch;// stretch the particular element from top to bottom
}
```




```
#box1{
    order: 1;
    align-self: flex-start; // place the element at start line
}
```



```
#box4{
    align-self: flex-end; //place the element at bottom line
}
```




```
#box2{
    order: 5;
    align-self: center;
}
```