

# CSS FlexBox

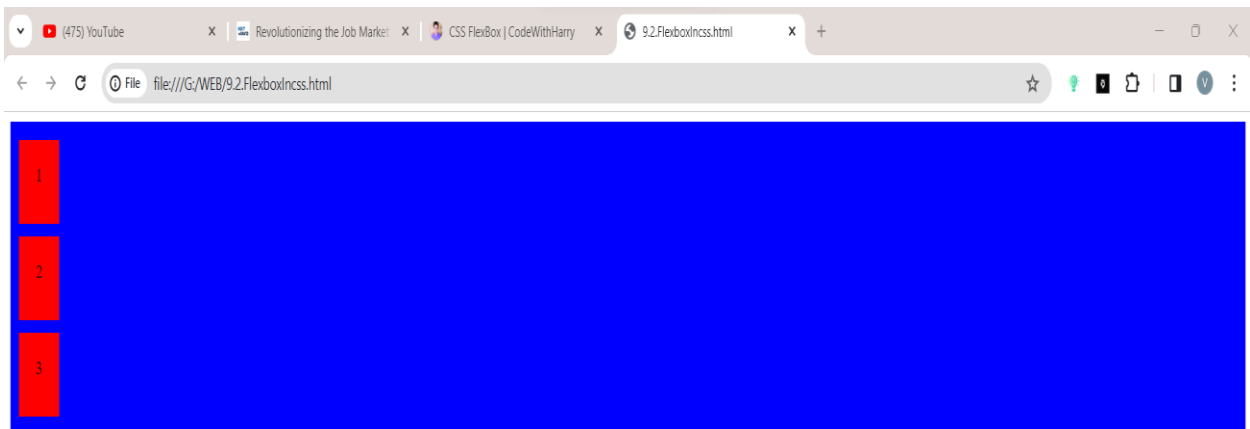
FlexBox aka Flexible Box Layout makes it easier to layout, align and style items in the container while maintaining the responsiveness of the website.

To create a flexbox you need to set the display of the container as flex

Eg: {display: flex;}

This element is called the flex container, and stores the sub-elements which are known as flex items

```
<head>
  <style>
    .background {
      background-color: blue;
      padding-top: 5px;
      padding-bottom: 5px;
    }
    .cubes {
      text-align: center;
      padding-top: 20px;
      height: 50px;
      width: 50px;
      background-color: red;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div class="background">
    <div class="cubes">1</div>
    <div class="cubes">2</div>
    <div class="cubes">3</div>
  </div>
</body>
html>
```



## Flex Container Properties

The flex container properties are:

### 1. Flex Direction

It defines in which direction the flex elements would be displayed. It takes values like **row**, **column** or "reverse" too.

1.COLUMN:

```
4      <style>
5      .background {
6          background-color: blue;
7          padding-top: 5px;
8          padding-bottom: 5px;
9          display: flex;
10         flex-direction: column;
11     }
12     .cubes {
13         text-align: center;
14         padding-top: 20px;
15         height: 50px;
16         width: 50px;
17         background-color: red;
18         margin: 10px;
19     }
20     </style>
21 </head>
22 <body>
23     <div class="background">
24         <div class="cubes">1</div>
25         <div class="cubes">2</div>
26         <div class="cubes">3</div>
27     </div>
28 </body>
29 </html>
```



## 2.ROW

```
3 <head>
4 <style>
5   .background {
6     background-color: blue;
7     padding-top: 5px;
8     padding-bottom: 5px;
9     display: flex;
10    flex-direction: row;
11  }
12  .cubes {
13    text-align: center;
14    padding-top: 20px;
15    height: 50px;
16    width: 50px;
17    background-color: red;
18    margin: 10px;
19  }
20 </style>
21 </head>
22 <body>
23   <div class="background">
24     <div class="cubes">1</div>
25     <div class="cubes">2</div>
26     <div class="cubes">3</div>
27   </div>
28 </body>
29 </html>
```

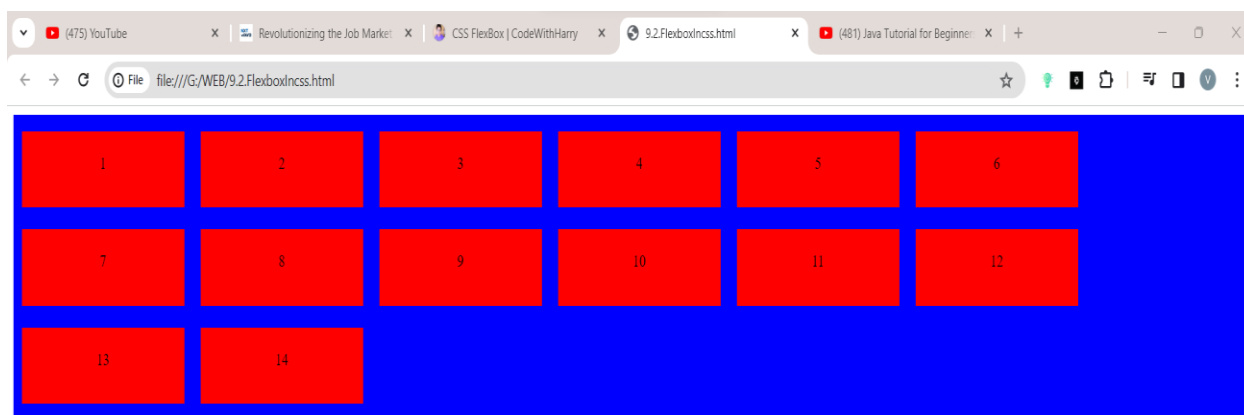


## 2. Flex Wrap

By using this property we can make our elements responsive for different screen sizes.

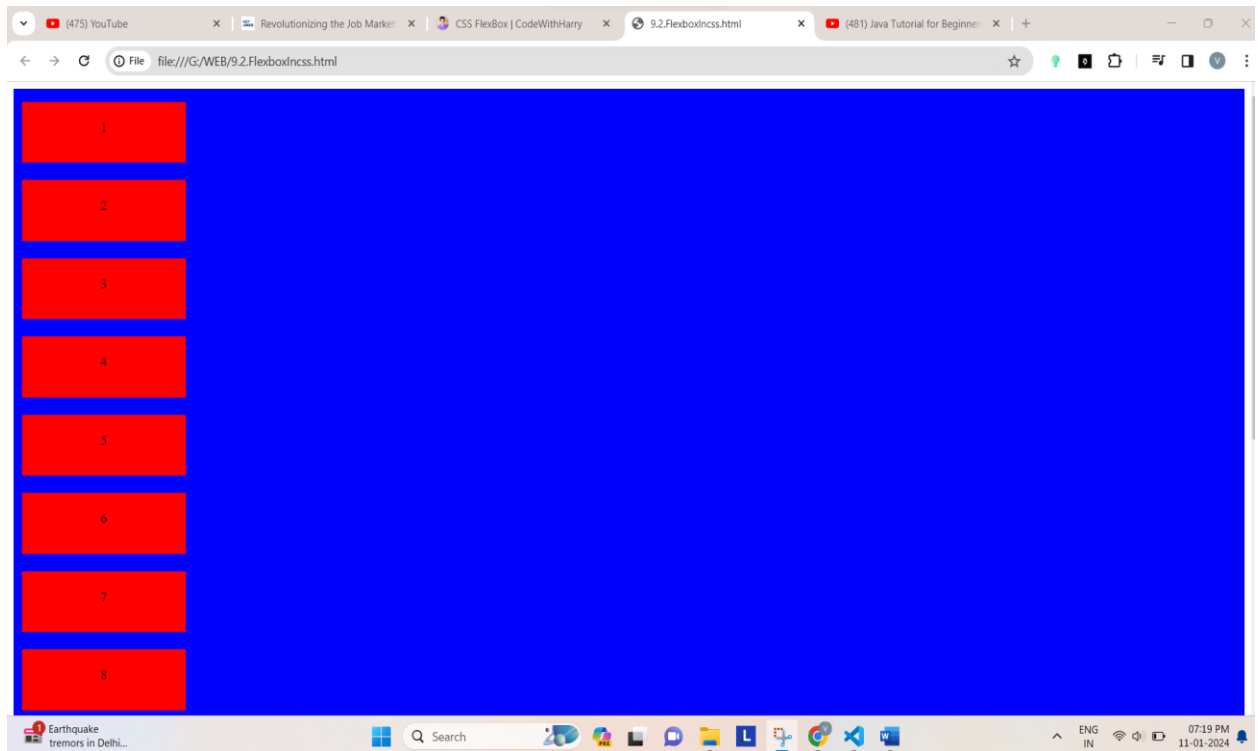
1.ROW

```
<style>
  .background {
    background-color: blue;
    padding-top: 5px;
    padding-bottom: 5px;
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
  }
  .cubes {
    text-align: center;
    padding-top: 20px;
    height: 50px;
    width: 200px;
    background-color: red;
    margin: 10px;
  }
</style>
</head>
<body>
  <div class="background">
    <div class="cubes">1</div>
    <div class="cubes">2</div>
    <div class="cubes">3</div>
    <div class="cubes">4</div>
    <div class="cubes">5</div>
    <div class="cubes">6</div>
    <div class="cubes">7</div>
    <div class="cubes">8</div>
    <div class="cubes">9</div>
    <div class="cubes">10</div>
    <div class="cubes">11</div>
    <div class="cubes">12</div>
    <div class="cubes">13</div>
    <div class="cubes">14</div>
  </div>
```



## 2.COLUMN

```
<style>
  .background {
    background-color: blue;
    padding-top: 5px;
    padding-bottom: 5px;
    display: flex;
    flex-direction: column;
    flex-wrap: wrap;
  }
  .cubes {
    text-align: center;
    padding-top: 20px;
    height: 50px;
    width: 200px;
    background-color: red;
    margin: 10px;
  }
</style>
</head>
<body>
  <div class="background">
    <div class="cubes">1</div>
    <div class="cubes">2</div>
    <div class="cubes">3</div>
    <div class="cubes">4</div>
    <div class="cubes">5</div>
    <div class="cubes">6</div>
    <div class="cubes">7</div>
    <div class="cubes">8</div>
    <div class="cubes">9</div>
    <div class="cubes">10</div>
    <div class="cubes">11</div>
    <div class="cubes">12</div>
    <div class="cubes">13</div>
    <div class="cubes">14</div>
  </div>
```



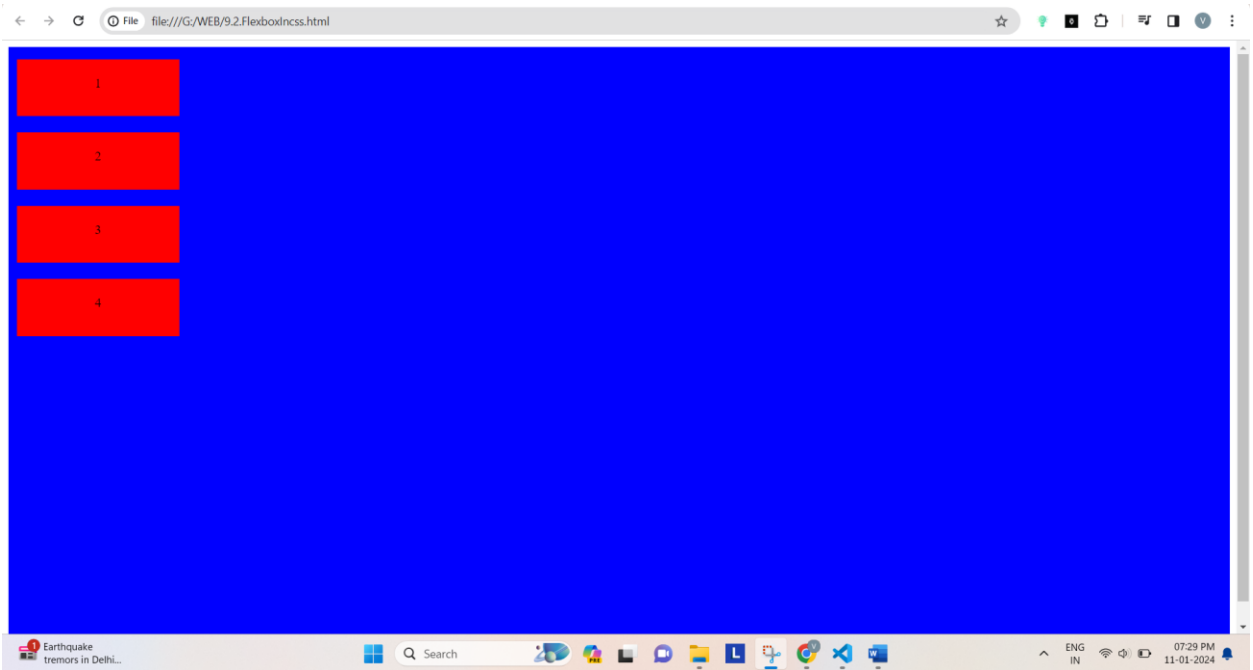
### 3. Justify Content

This property is used to set the position of content or rather align content along the main axis.

#### 1.COLUMN START

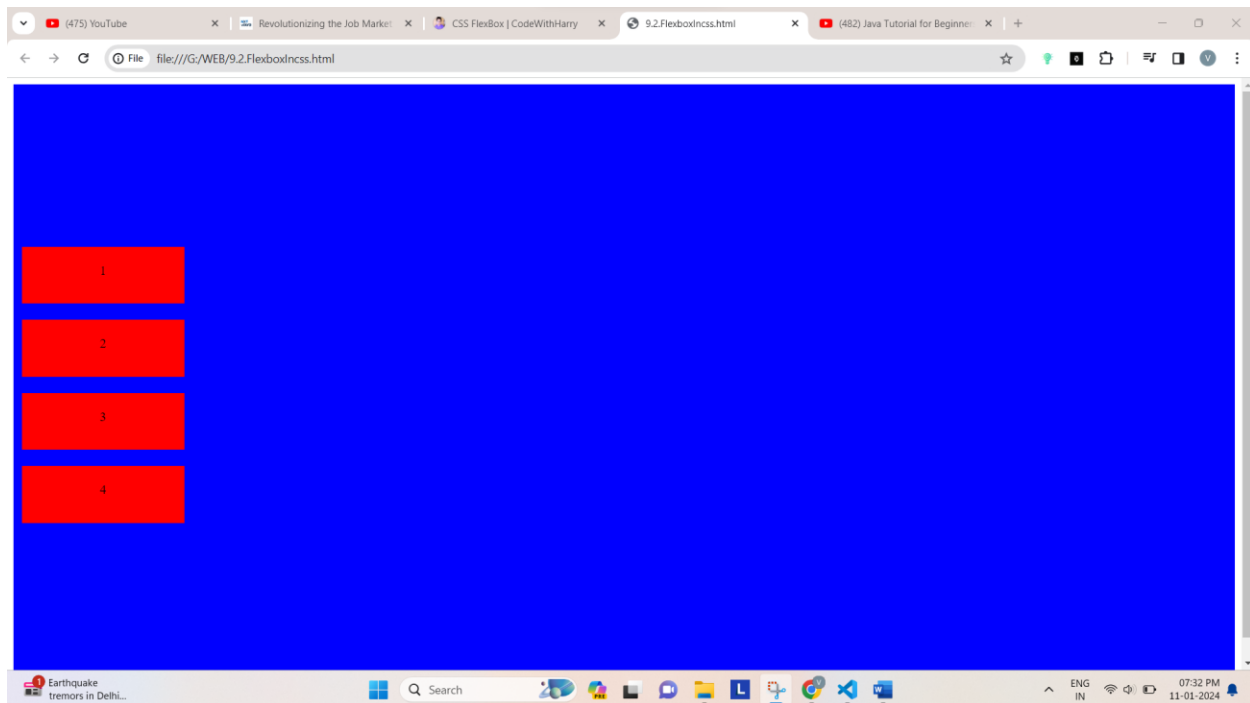
```
<head>
  <style>
    .background {
      background-color: blue;
      height: 100vh;
      padding-top: 5px;
      padding-bottom: 5px;
      display: flex;
      flex-direction: column;
      justify-content: start;
    }
    .cubes {
      text-align: center;
      padding-top: 20px;
      height: 50px;
      width: 200px;
      background-color: red;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div class="background">
    <div class="cubes">1</div>
    <div class="cubes">2</div>
    <div class="cubes">3</div>
    <div class="cubes">4</div>
  </div>
</body>
</html>
```





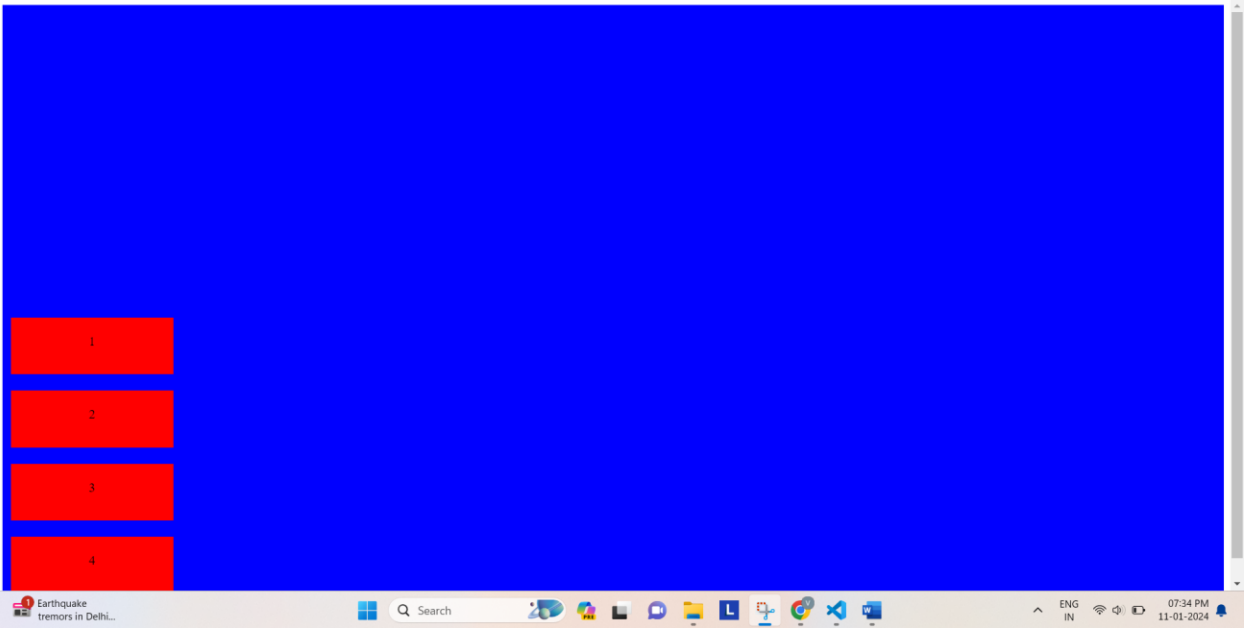
## 2.COLUMN-CENTER

```
<head>
  <style>
    .background {
      background-color: blue;
      height: 100vh;
      padding-top: 5px;
      padding-bottom: 5px;
      display: flex;
      flex-direction: column;
      justify-content: center;
    }
    .cubes {
      text-align: center;
      padding-top: 20px;
      height: 50px;
      width: 200px;
      background-color: red;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div class="background">
    <div class="cubes">1</div>
    <div class="cubes">2</div>
    <div class="cubes">3</div>
    <div class="cubes">4</div>
  </div>
</body>
</html>
```



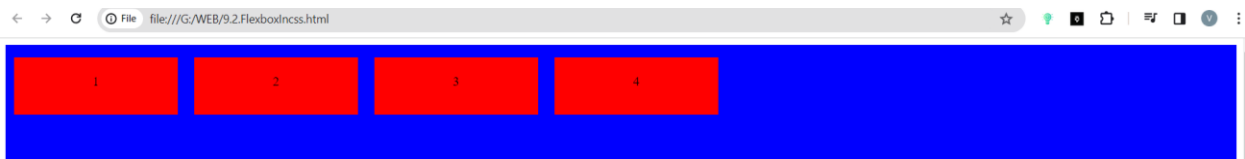
### 3.COLUMN-END

```
<head>
  <style>
    .background {
      background-color: blue;
      height: 100vh;
      padding-top: 5px;
      padding-bottom: 5px;
      display: flex;
      flex-direction: column;
      justify-content: end;
    }
    .cubes {
      text-align: center;
      padding-top: 20px;
      height: 50px;
      width: 200px;
      background-color: red;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div class="background">
    <div class="cubes">1</div>
    <div class="cubes">2</div>
    <div class="cubes">3</div>
    <div class="cubes">4</div>
  </div>
</body>
</html>
```



## 1.ROW-START

```
<head>
  <style>
    .background {
      background-color: blue;
      height: 100vh;
      padding-top: 5px;
      padding-bottom: 5px;
      display: flex;
      flex-direction: row;
      justify-content: start;
    }
    .cubes {
      text-align: center;
      padding-top: 20px;
      height: 50px;
      width: 200px;
      background-color: red;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div class="background">
    <div class="cubes">1</div>
    <div class="cubes">2</div>
    <div class="cubes">3</div>
    <div class="cubes">4</div>
  </div>
</body>
</html>
```



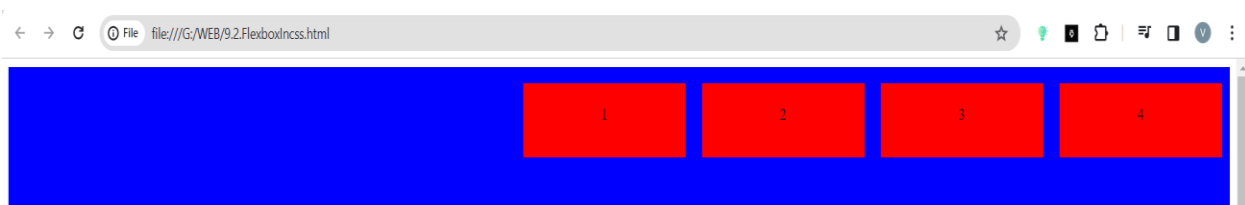
## 2.ROW-CENTER

```
<style>
  .background {
    background-color: blue;
    height: 100vh;
    padding-top: 5px;
    padding-bottom: 5px;
    display: flex;
    flex-direction: row;
    justify-content: center;
  }
  .cubes {
    text-align: center;
    padding-top: 20px;
    height: 50px;
    width: 200px;
    background-color: red;
    margin: 10px;
  }
</style>
</head>
<body>
  <div class="background">
    <div class="cubes">1</div>
    <div class="cubes">2</div>
    <div class="cubes">3</div>
    <div class="cubes">4</div>
  </div>
</body>
</html>
```



### 3.ROW-END

```
<style>
  .background {
    background-color: blue;
    height: 100vh;
    padding-top: 5px;
    padding-bottom: 5px;
    display: flex;
    flex-direction: row;
    justify-content: end;
  }
  .cubes {
    text-align: center;
    padding-top: 20px;
    height: 50px;
    width: 200px;
    background-color: red;
    margin: 10px;
  }
</style>
</head>
<body>
  <div class="background">
    <div class="cubes">1</div>
    <div class="cubes">2</div>
    <div class="cubes">3</div>
    <div class="cubes">4</div>
  </div>
</body>
</html>
```






space-between:

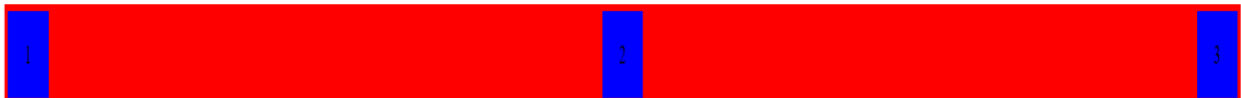
- Distributes items evenly, with the first item at the start and the last item at the end.
- Example:

CSS

 Copy code

```
justify-content: space-between;
```


```
<style>
  .main-div {
    background-color: red;
    padding: 2px;
    display: flex;
    flex-direction: row;
    justify-content: space-between;
  }
  .sub-div {
    height: 50px;
    width: 50px;
    background-color: blue;
    margin: 2px;
    display: flex;
    flex-direction: row;
    justify-content: center;
    align-items: center;
  }
</style>
</head>
<body>
  <div class="main-div">
    <div class="sub-div">1</div>
    <div class="sub-div">2</div>
    <div class="sub-div">3</div>
  </div>
</body>
```



space-around :

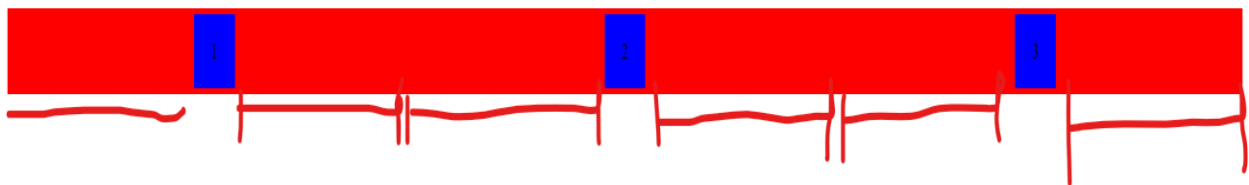
- Distributes items with equal space around them. Items have half-sized space at the ends compared to the space between items.
- Example:

CSS

 Copy code

```
justify-content: space-around;
```


```
<style>
  .main-div {
    background-color: red;
    padding: 2px;
    display: flex;
    flex-direction: row;
    justify-content: space-around;
  }
  .sub-div {
    height: 50px;
    width: 50px;
    background-color: blue;
    margin: 2px;
    display: flex;
    flex-direction: row;
    justify-content: center;
    align-items: center;
  }
</style>
</head>
<body>
  <div class="main-div">
    <div class="sub-div">1</div>
    <div class="sub-div">2</div>
    <div class="sub-div">3</div>
  </div>
```



space-evenly:

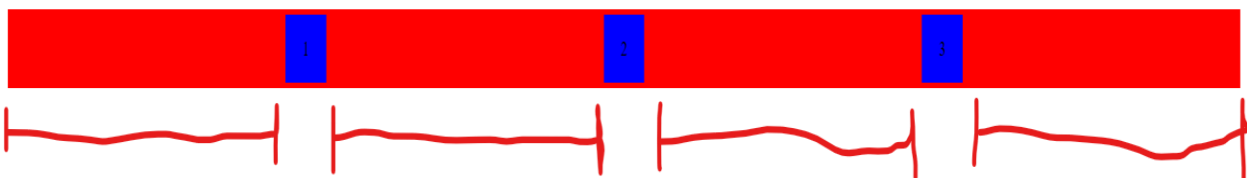
- Distributes items with equal space between and around them.
- Example:

CSS

 Copy code

justify-content: space-evenly;

```
<style>
  .main-div {
    background-color: red;
    padding: 2px;
    display: flex;
    flex-direction: row;
    justify-content: space-evenly;
  }
  .sub-div {
    height: 50px;
    width: 50px;
    background-color: blue;
    margin: 2px;
    display: flex;
    flex-direction: row;
    justify-content: center;
    align-items: center;
  }
</style>
</head>
<body>
  <div class="main-div">
    <div class="sub-div">1</div>
    <div class="sub-div">2</div>
    <div class="sub-div">3</div>
  </div>
</body>
```



## 4. Align Items

Just like the justify-content property, align-items define the alignment of the flex container but along the **cross-axis**.

### Difference Between Main Axis and Cross Axis:

In flexbox, the main axis and cross axis depend on the flex-direction property:

#### 1. Main Axis:

- It's the **direction in which the flex items are arranged**.
- By default, it's **horizontal (left to right)** because `flex-direction` is set to `row` by default.

#### 2. Cross Axis:

- It's **perpendicular to the main axis**.
- For example, if the main axis is horizontal (`row`), the cross axis will be **vertical (top to bottom)**.

### Flex Directions:

`flex-direction: row` (default):

- Main Axis: Horizontal (left to right).
- Cross Axis: Vertical (top to bottom).

`flex-direction: column`:

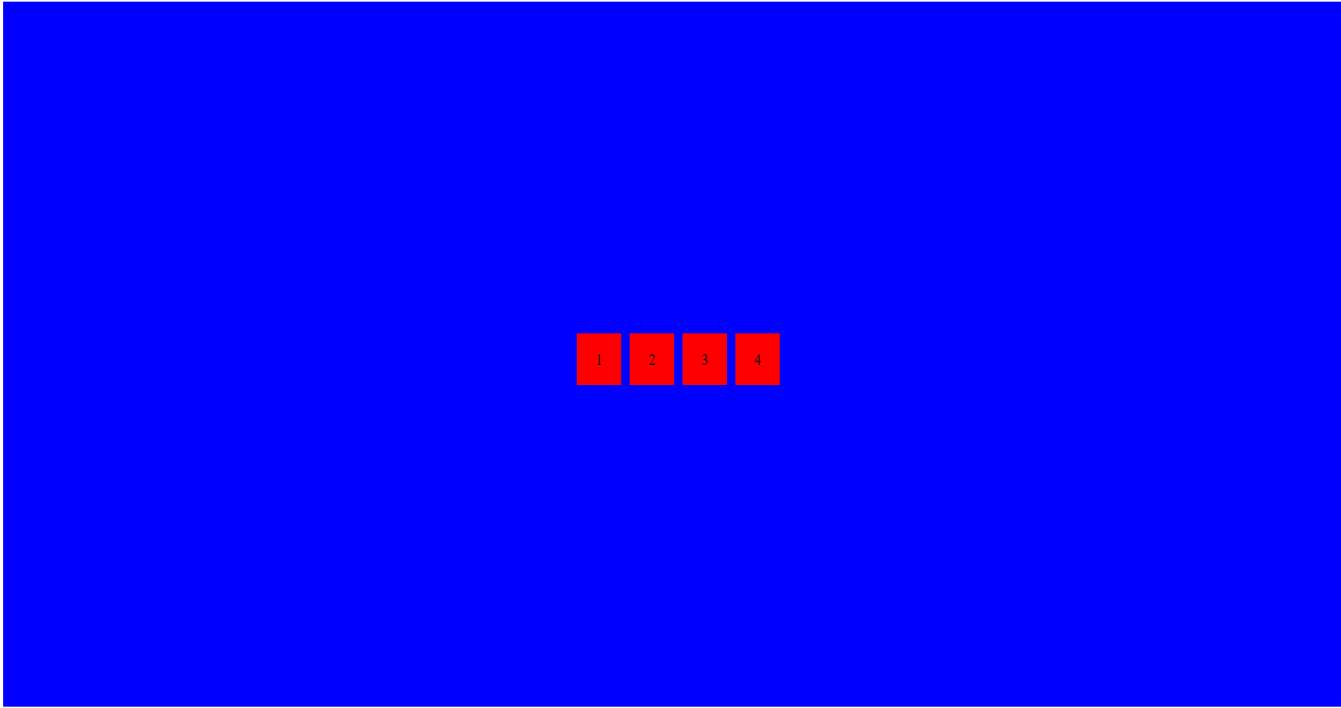
- Main Axis: Vertical (top to bottom).
- Cross Axis: Horizontal (left to right).

## Key Points:

- Main axis is controlled by `flex-direction`.
- Cross axis is always perpendicular to the main axis.
- Use `justify-content` to align items along the main axis.
- Use `align-items` to align items along the cross axis.

TO use `align-items` We have  
mention `justify-content` bcz  
they both are perpendicular to  
each other.

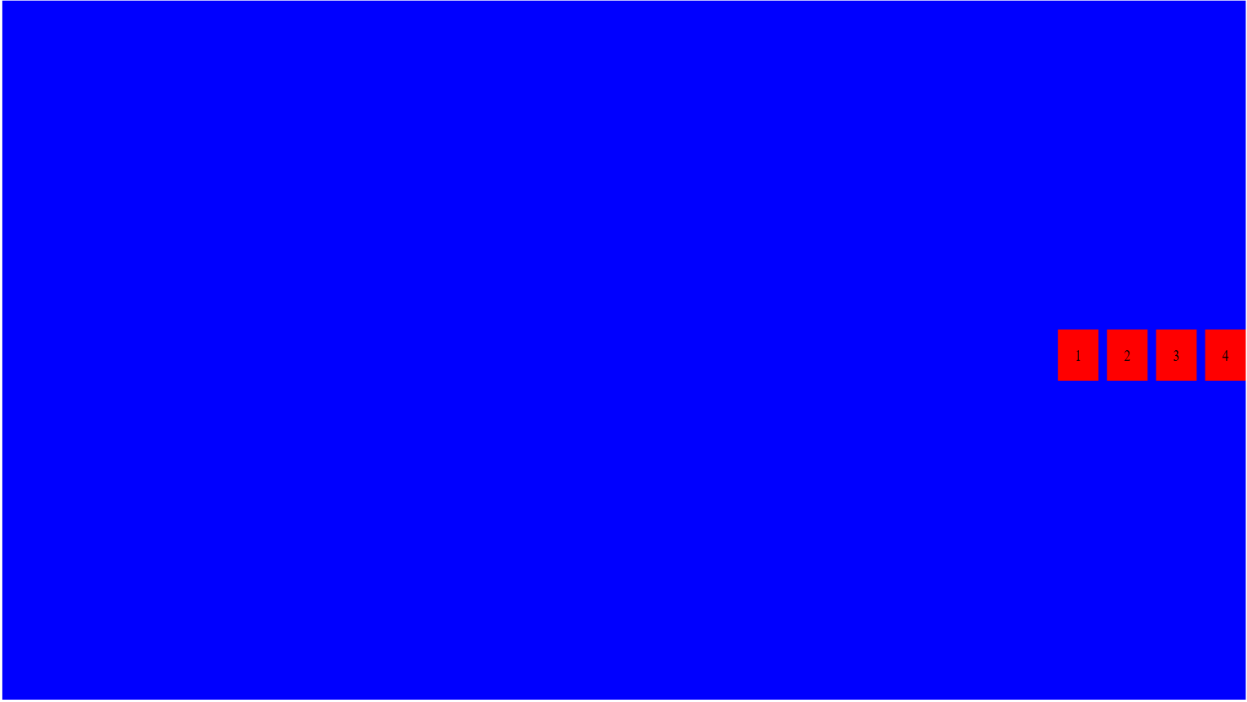
```
<style>
  .mainbox {
    background-color: blue;
    height: 90vh;
    padding-top: 10px;
    padding-bottom: 10px;
    padding-left: 3px;
    display: flex;
    flex-direction: row;
    justify-content: center;
    align-items: center;
  }
  .box {
    background-color: red;
    height: 50px;
    width: 50px;
    margin-top: 10px;
    margin-left: 10px;
    display: flex;
    justify-content: center;
    align-items: center;
  }
</style>
</head>
<body>
  <div class="mainbox">
    <div class="box">1</div>
    <div class="box">2</div>
    <div class="box">3</div>
    <div class="box">4</div>
  </div>
</body>
</html>
```



1 2 3 4

```
<style>
  .mainbox {
    background-color: blue;
    height: 90vh;
    padding-top: 10px;
    padding-bottom: 10px;
    padding-left: 3px;
    display: flex;
    flex-direction: row;
    justify-content: end;
    align-items: center;
  }
  .box {
    background-color: red;
    height: 50px;
    width: 50px;
    margin-top: 10px;
    margin-left: 10px;
    display: flex;
    justify-content: center;
    align-items: center;
  }
</style>
</head>
<body>
  <div class="mainbox">
    <div class="box">1</div>
    <div class="box">2</div>
    <div class="box">3</div>
    <div class="box">4</div>
  </div>
</body>
```





```
.mainbox {
  background-color: blue;
  height: 90vh;
  padding-top: 10px;
  padding-bottom: 10px;
  padding-left: 3px;
  display: flex;
  flex-direction: row;
  justify-content: end;
  align-items: end;
}

.box {
  background-color: red;
  height: 50px;
  width: 50px;
  margin-top: 10px;
  margin-left: 10px;
  display: flex;
  justify-content: center;
  align-items: center;
}

</style>
<head>
<body>
<div class="mainbox">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
</div>
</body>
</html>
```



