

Flexbox: It's a model system for positioning or moving around elements for laying out a webpage. It makes things a lot easier and a lot less messy. Pre-flexbox days, people used things like floats. Flexbox is enabled by default that does not mean we have to know all about Flexbox to use Bootstrap 4, but it is now built on it and it wasn't in the past. This also means that if we do know Flexbox, then we can use some nice Flexbox wrappers, some nice classes those are included, these utility classes that we can use to move things or position things more precisely.

Flexbox is something that Bootstrap 4 is built on top of, and Bootstrap 3 wasn't. The next important point is that the main unit that was used in Bootstrap 3 was pixels (px) and now in Bootstrap 4 it is something called rems. It is basically a relative unit that allows us to scale the size of things on the webpage based off of global font size, and the global font size has changed from 14px to 16px, that does not mean everything is 16px on our entire page, it just means that the global font size is 16px, and when we use rems we are giving different size of things based off of that global size. So, if I say something is 2 times the global size or of 5 times the global size then I am referring to the global size of 16px.

There are dozen more utility classes those are useful, these allow us to set common properties like padding and margin, also specifically padding-top or margin-top etc. All these different combinations we can do easily with a single class that we add to elements.

We can set a border on our own without Bootstrap, we can also use the border utility that Bootstrap provides to make our job easier. We can change the display property on our own using CSS, or we can use the display utility using Bootstrap. It's the same thing with Flexbox, so Flexbox includes a bunch of CSS properties to move things around, to position elements inside of a container, inside of a page, and we can do it all without Bootstrap, Bootstrap 4 comes with a whole bunch of utilities that has to do with Flexbox.

So, when we go to the tab of Utilities and select Flex, we will see what we are talking about. **Flexbox is a model for positioning contents inside of a box.** Bootstrap makes it very easy perform the function of Flexbox.

```
<div class= "container">
  <h1 class= "text-center display-4">Flexbox</h1>

  <div class= "border border-dark">
    <button class= "btn btn-info btn-lg">LARGE</button>
    <button class= "btn btn-warning">SMALL</button>
  </div>
</div>
```

Output:



```
<div class= "container">  
  <h1 class= "text-center display-4">Flexbox</h1>  
  
  <div class= "border border-dark" style= "height: 200px">  
    <button class= "btn btn-info btn-lg">LARGE</button>  
    <button class= "btn btn-warning">SMALL</button>  
  </div>  
</div>
```

Output:



```

<div class= "container">

    <h1 class= "text-center display-4">Flexbox</h1>

    <div class= "border border-dark d-flex" style= "height: 200px">

        <button class= "btn btn-info btn-lg">LARGE</button>

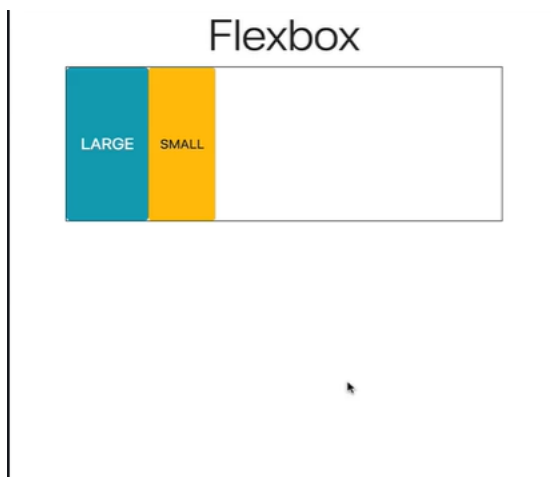
        <button class= "btn btn-warning">SMALL</button>

    </div>

</div>

```

Output:



What happened here is the height of the button now takes up all the border, it is stretched vertically. We will come back to it.

What we must know about Flexbox is that there are two directions, the first by default is the main axis of our content is from left to right. So, if I say the start of our flexbox its on the left side. The end is on the right side. Then there is also what is known as the cross axis, which is by default from top to bottom. The start of our cross axis is on the top and the end is at the bottom. There is a property on Bootstrap that lets us change how items are distributed on the main axis (left to right) and there is another property that lets us change the distribution of items vertically on the cross axis (top to bottom).

But the crazy thing is we can also change the direction of our flexbox, so it goes from right to left or from bottom to top. So, when we say the **flex direction** its from **left to right**, when we say **cross direction** its **top to bottom**.

If we are working on the left to right axis, then there is a property called ***justify-content***. Justify-content allows us to move the contents from left to right on the flexbox, and the default value is justify-content start. We don't have to write *justify-content-start* as only writing justify-content will work as justify-content-start. But if we add the class ***justify-content-end*** that will push the buttons at the end (right) of the flexbox.

```
<div class= "container">
  <h1 class= "text-center display-4">Flexbox</h1>

  <div class= "border border-dark d-flex justify-content-end" style= "height: 200px">
    <button class= "btn btn-info btn-lg">LARGE</button>
    <button class= "btn btn-warning">SMALL</button>
  </div>
</div>
```

Output:



So, the class ***justify-content-end*** moves everything to the right.

We also have ***justify-content-center*** that will bring the buttons to the center of the flexbox.

```

<div class= "container">

    <h1 class= "text-center display-4">Flexbox</h1>

    <div class= "border border-dark d-flex justify-content-center" style= "height: 200px">

        <button class= "btn btn-info btn-lg">LARGE</button>

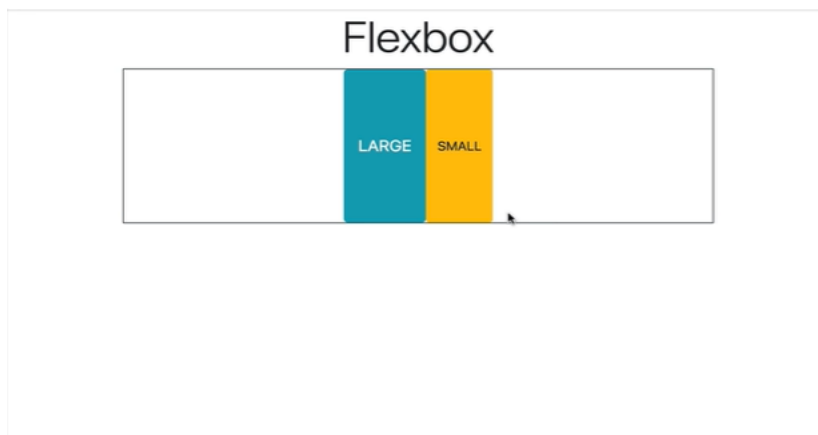
        <button class= "btn btn-warning">SMALL</button>

    </div>

</div>

```

Output:



We also have another flexbox class call **justify-content-between**, this will take all the available space and put it between the two buttons or however many buttons are in there.

```

<div class= "container">

    <h1 class= "text-center display-4">Flexbox</h1>

    <div class= "border border-dark d-flex justify-content-between" style= "height: 200px">

        <button class= "btn btn-info btn-lg">LARGE</button>

        <button class= "btn btn-warning">SMALL</button>

    </div>

</div>

```

Output:



```
<div class= "container">
```

```
  <h1 class= "text-center display-4">Flexbox</h1>
```

```
  <div class= "border border-dark d-flex justify-content-between" style= "height: 200px">
```

```
    <button class= "btn btn-info btn-lg">LARGE</button>
```

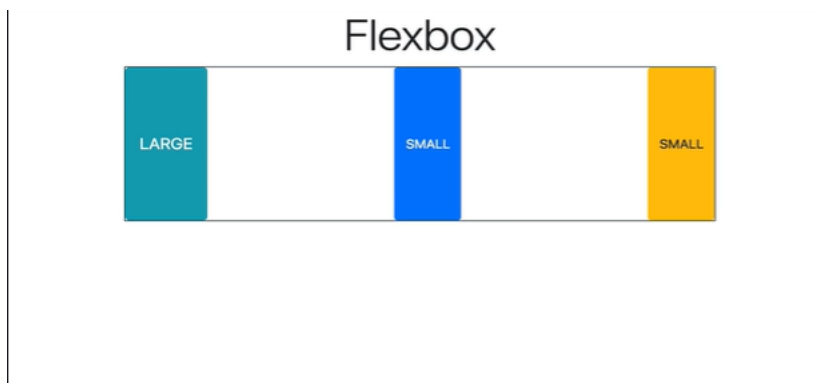
```
    <button class= "btn btn-primary">SMALL</button>
```

```
    <button class= "btn btn-warning">SMALL</button>
```

```
  </div>
```

```
</div>
```

Output:



Now all the spaces are distributed between the items.

Finally, we have another class which is called ***justify-content-around***. The property of this class takes all the space and distribute evenly around the items or the buttons. So, there is space between the ends of the buttons and between the buttons.

```
<div class= "container">
```

```
  <h1 class= "text-center display-4">Flexbox</h1>
```

```
  <div class= "border border-dark d-flex justify-content-around" style= "height: 200px">
```

```
    <button class= "btn btn-info btn-lg">LARGE</button>
```

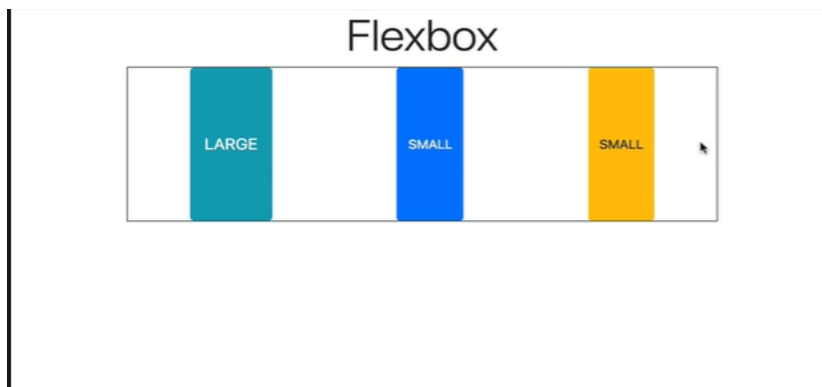
```
    <button class= "btn btn-primary">SMALL</button>
```

```
    <button class= "btn btn-warning">SMALL</button>
```

```
  </div>
```

```
</div>
```

Output:



It does not need to be only three items, there could be more items and we can move them horizontally on our main axis using the properties of these classes.

justify-content-start

justify-content-end

justify-content-center

justify-content-between

justify-content-around

Now what if we want to move our items from top to bottom on our cross axis. Then we follow the similar rules but now it is called ***align-items***.

So, if we use the class ***align-items-start*** the buttons will move to the top of the flexbox.

If we use the class ***align-items-end*** the buttons will move to the bottom of the flexbox.

If we use the class ***align-items-center*** the buttons will stay at the center of the cross axis.

If we use the class ***align-items-baseline*** the buttons will move to the top of the cross axis.

If we use the class ***align-items-stretch*** the buttons will cover the all of the cross axis from top to bottom.

Use **align-items** utilities on flexbox containers to change the alignment of flex items on the cross axis (the y-axis to start, x-axis if **flex-direction: column**). Choose from **start**, **end**, **center**, **baseline**, or **stretch** (browser default).



```
<div class="d-flex align-items-start">...</div>
<div class="d-flex align-items-end">...</div>
<div class="d-flex align-items-center">...</div>
<div class="d-flex align-items-baseline">...</div>
<div class="d-flex align-items-stretch">...</div>
```

Copy

```
<div class= "container">
```

```
  <h1 class= "text-center display-4">Flexbox</h1>
```

```
  <div class= "border border-dark d-flex justify-content-around align-items-start" style=
    "height: 200px">
```

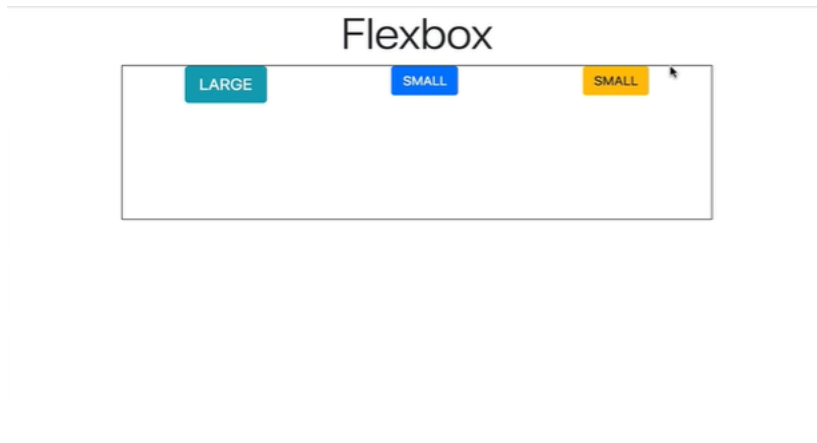
```
    <button class= "btn btn-info btn-lg">LARGE</button>
```

```
    <button class= "btn btn-primary">SMALL</button>
```

```
    <button class= "btn btn-warning">SMALL</button>
```


</div>

Output:



<div class= "container">

<h1 class= "text-center display-4">Flexbox</h1>

<div class= "border border-dark **d-flex justify-content-around align-items-end**" style= "height: 200px">

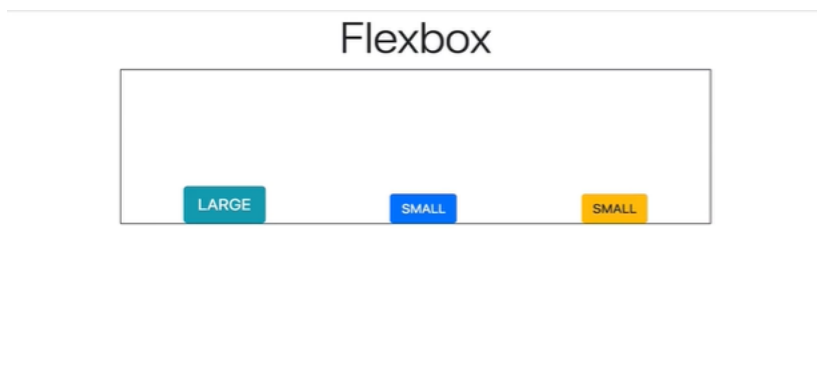
<button class= "btn btn-info btn-lg">LARGE</button>

<button class= "btn btn-primary">SMALL</button>

<button class= "btn btn-warning">SMALL</button>

</div>

Output:



```
<div class= "container">
```

```
  <h1 class= "text-center display-4">Flexbox</h1>
```

```
  <div class= "border border-dark d-flex justify-content-around align-items-center" style=
    "height: 200px">
```

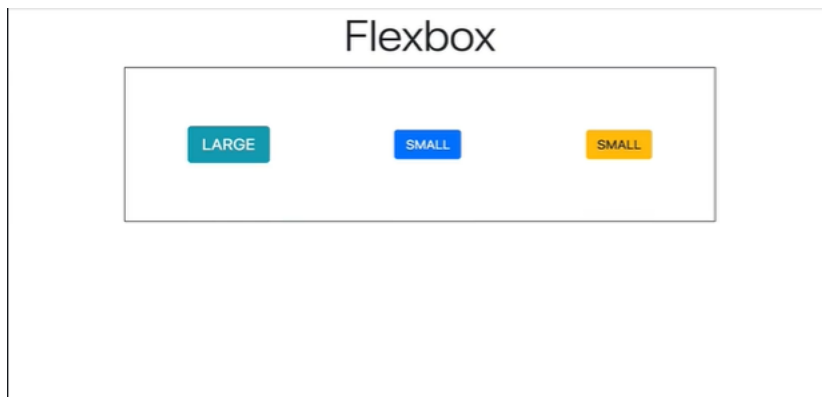
```
    <button class= "btn btn-info btn-lg">LARGE</button>
```

```
    <button class= "btn btn-primary">SMALL</button>
```

```
    <button class= "btn btn-warning">SMALL</button>
```

```
  </div>
```

Output:



```
<div class= "container">
```

```
  <h1 class= "text-center display-4">Flexbox</h1>
```

```
  <div class= "border border-dark d-flex justify-content-center align-items-center" style=
    "height: 200px">
```

```
    <button class= "btn btn-info btn-lg">LARGE</button>
```

```
    <button class= "btn btn-primary">SMALL</button>
```

```
    <button class= "btn btn-warning">SMALL</button>
```

```
  </div>
```

Output:

