@ineuron

JavaScript 2.0 boot camp

Assignment: 3

A complete guide to CSS flex box

Submitted by: Abhinandan Kumar

Email: abhinandankumar6204@gmail.com

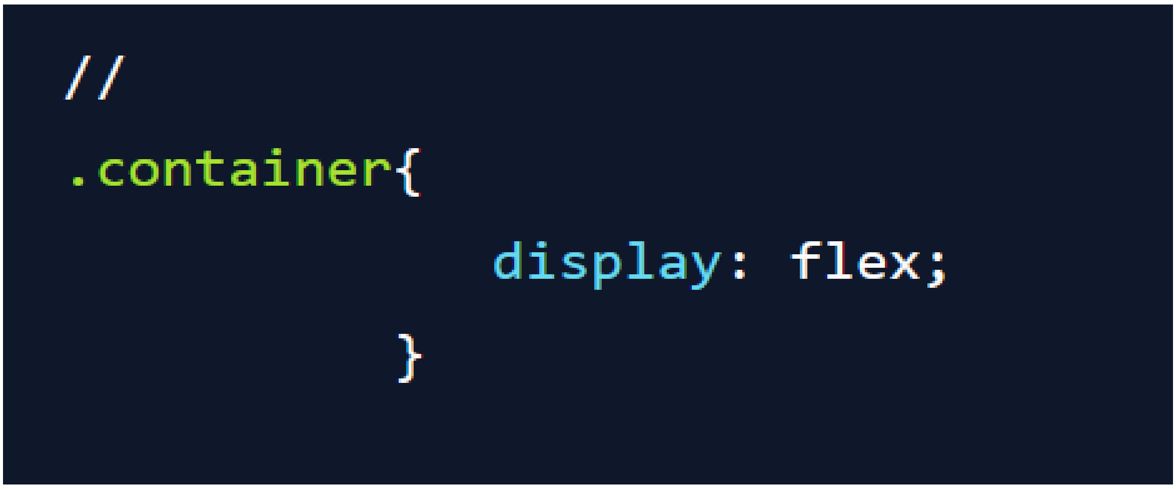
In this assignment we are going to learn the **CSS Flex box** now. In this part, we are going to learn what is flex box, what are flex-direction properties, properties on flex items and many things.

◈What is Flex box in CSS?

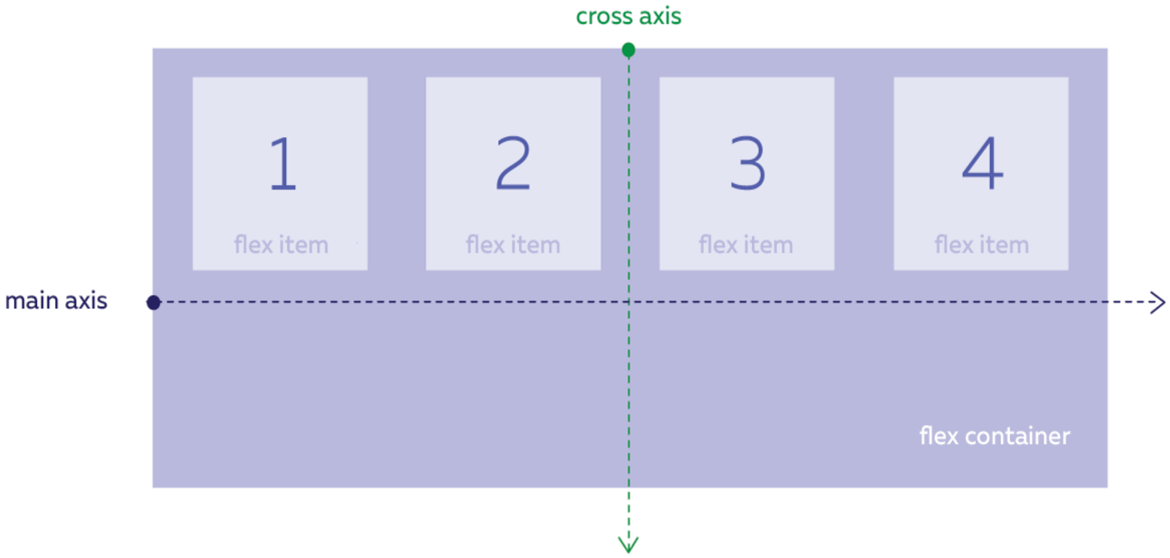
→The **flex box** is a property of CSS that provides a way to make the page layouts more efficient.

→In another word, we can say that the main aim of **Flex box** is providing a better way to layouts, align and distribute space among items in a container.

→Initialization of **Flex box** with the help of an image is shown below.



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



In this picture when we give the properties of display flex all the items became flex. The flex property enables all the flex box properties. The selected element becomes a flex container its child contents become flex items It will algorithmically layout its contents.

◈Flex-direction Properties:

→The flex-direction property defines the direction towards which items are laid can be row, row-reverse, column, or column-reverse.

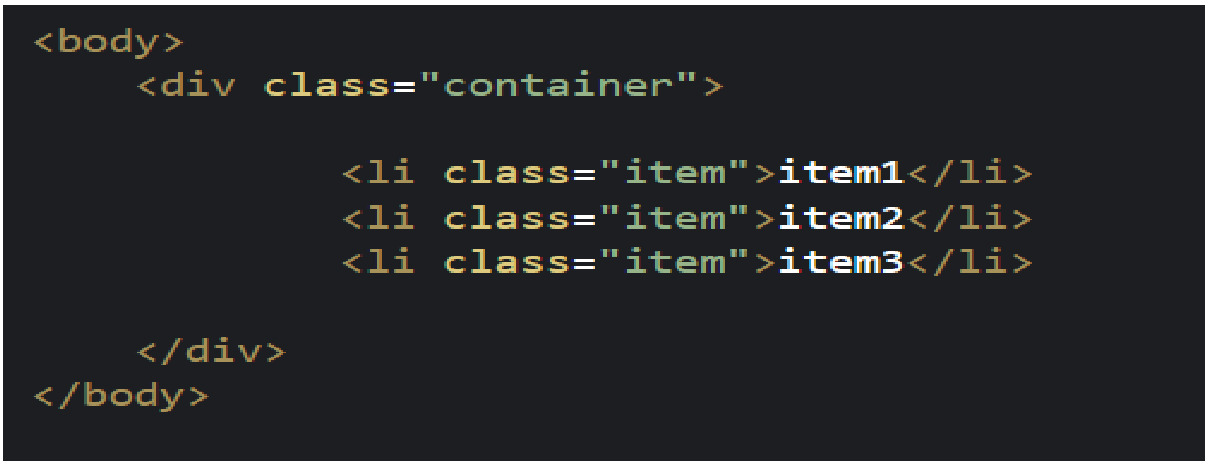
→By default, the value of flex-direction is row.

→Let's take an example with code that you can understand properly.

1| flex-direction: row; Flex-direction **is** row **by default** in Flex-box.

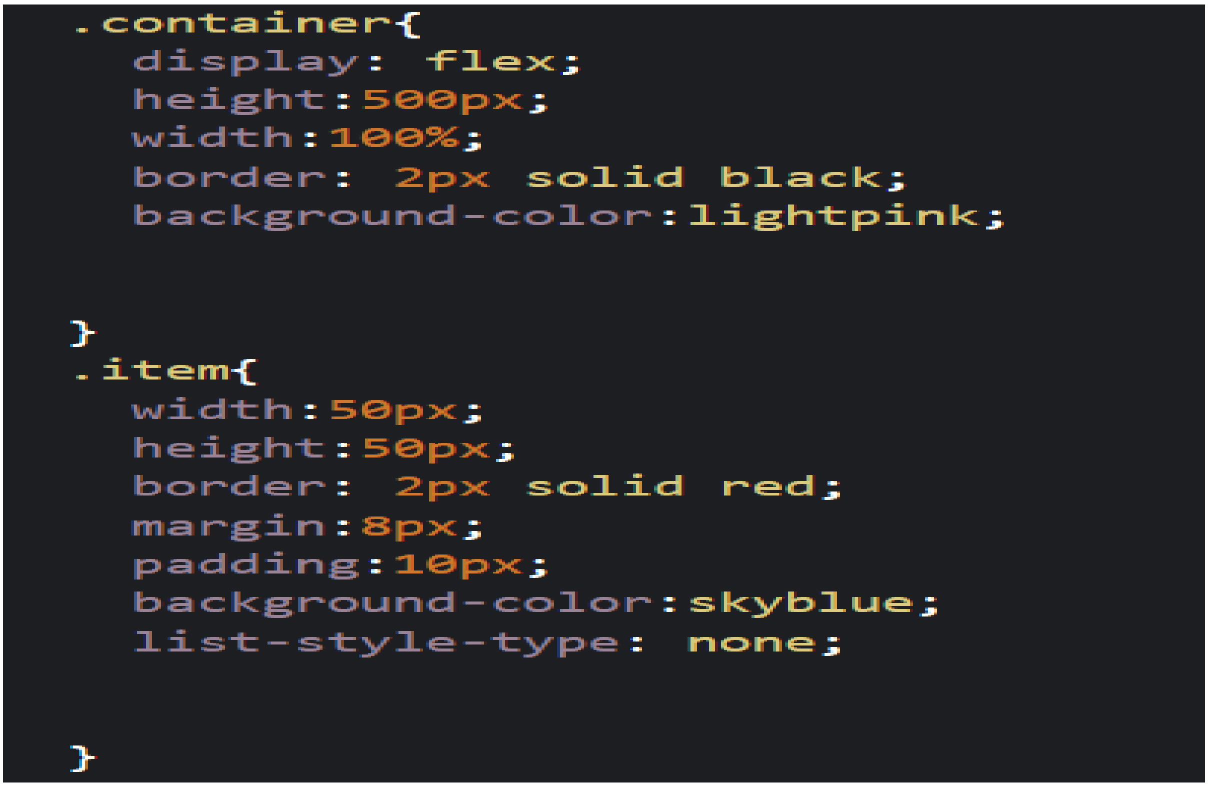
Let’s take an example:

HTML Code:



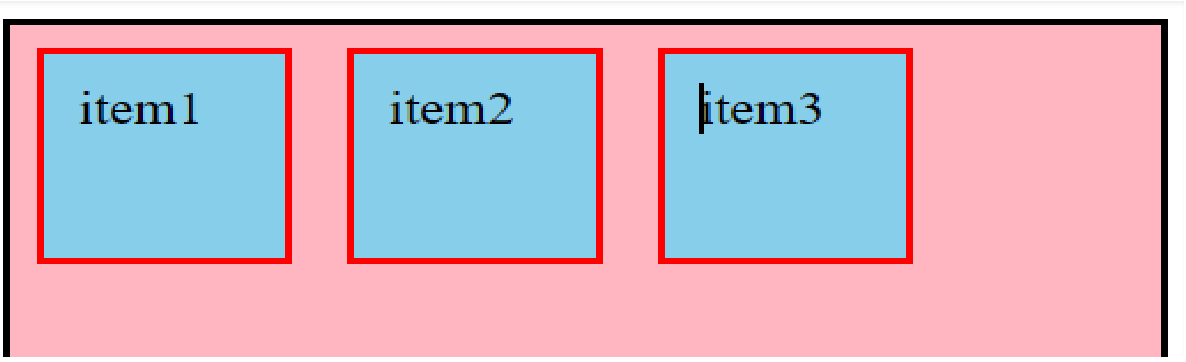
```
<body>
    <div class="container">

            <li class="item">item1</li>
            <li class="item">item2</li>
            <li class="item">item3</li>

    </div>
</body>
```

CSS Code



```
  .container{
    display: flex;
    height:500px;
    width:100%;
    border: 2px solid black;
    background-color:lightpink;


  }
  .item{
    width:50px;
    height:50px;
    border: 2px solid red;
    margin:8px;
    padding:10px;
    background-color:skyblue;
    list-style-type: none;


  }
```

Output

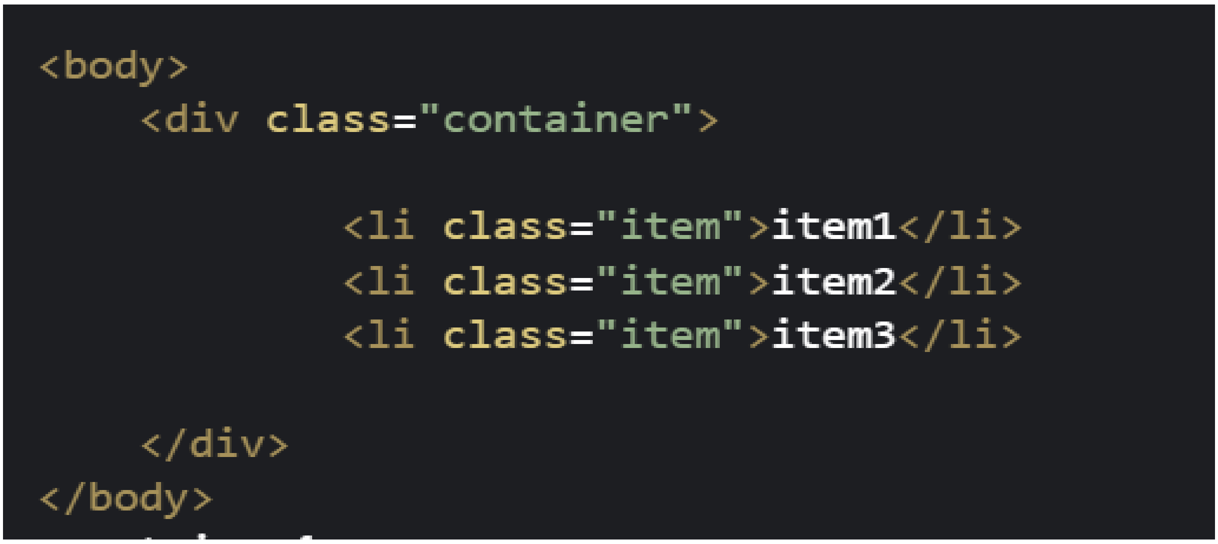


2| Flex-direction: column;

When we want it in the column we have to give the **flex-direction: column.**

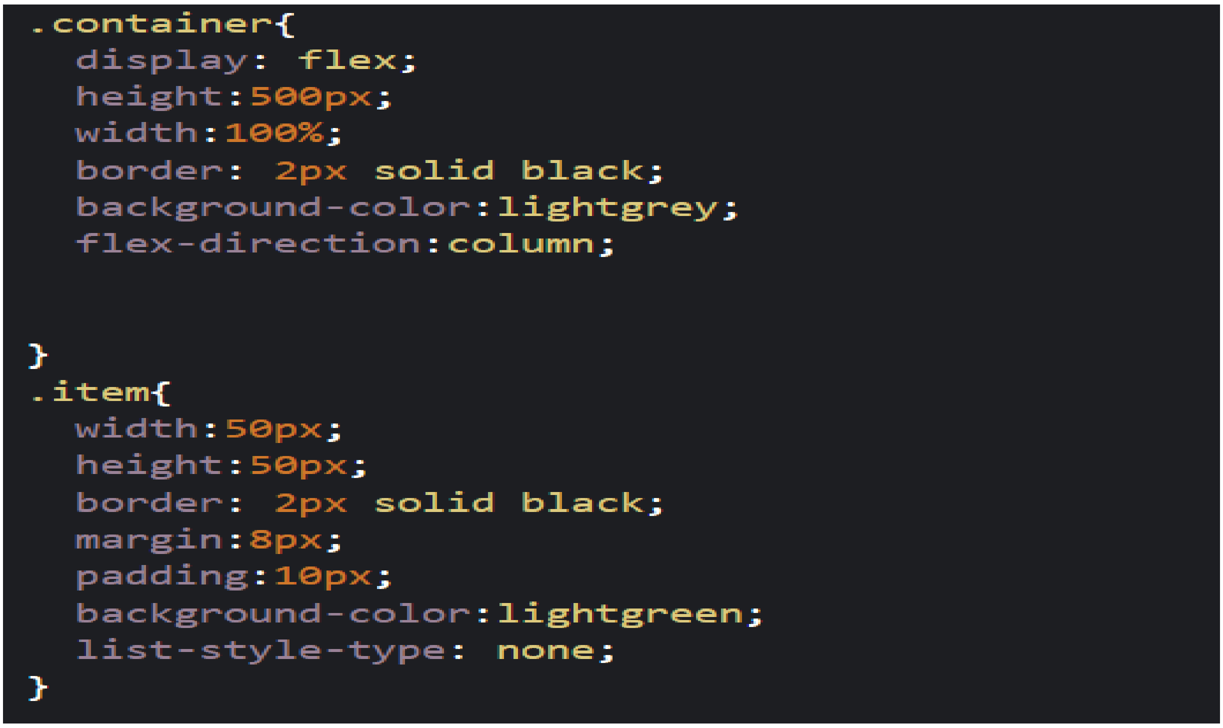
Let’s take an example:

HTML Code:



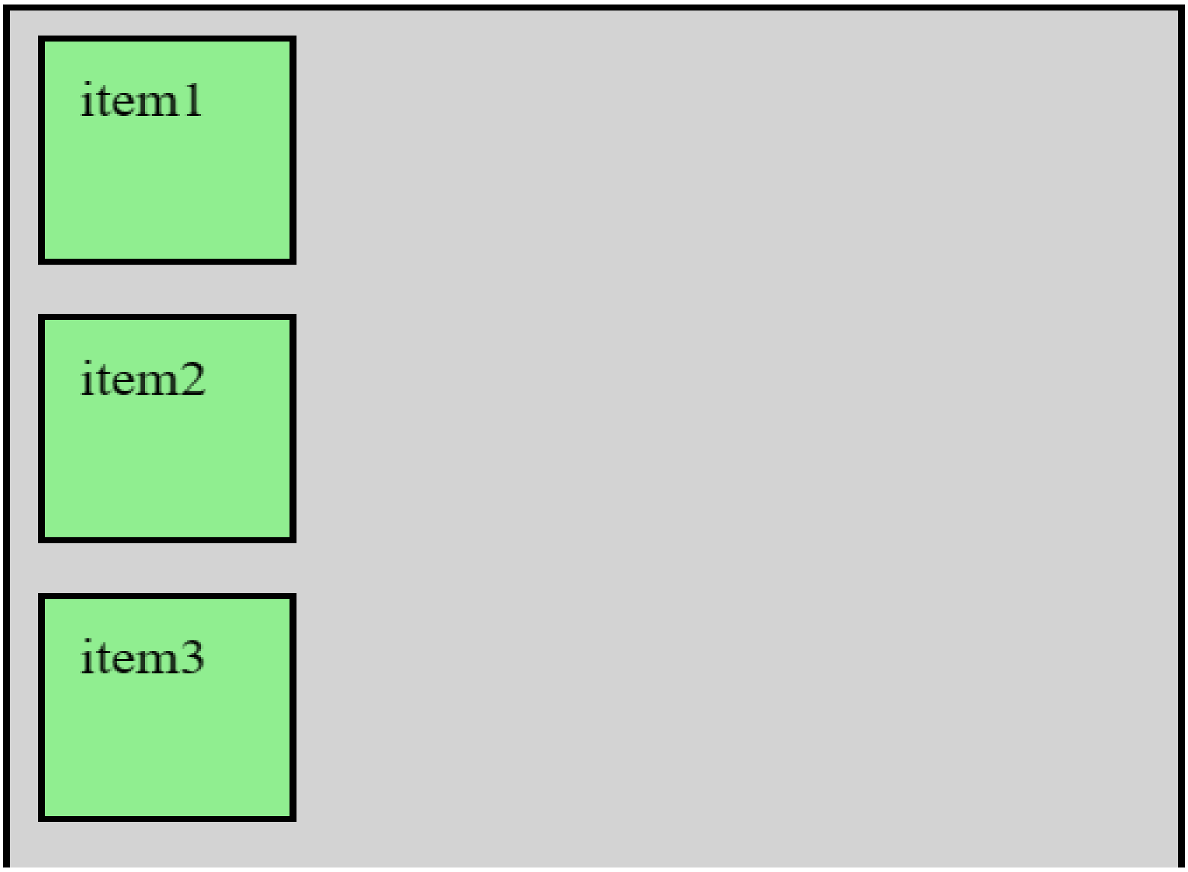
```
<body>
    <div class="container">

            <li class="item">item1</li>
            <li class="item">item2</li>
            <li class="item">item3</li>

    </div>
</body>
```

CSS Code:



```
.container{
  display: flex;
  height:500px;
  width:100%;
  border: 2px solid black;
  background-color:lightgrey;
  flex-direction:column;


}
.item{
  width:50px;
  height:50px;
  border: 2px solid black;
  margin:8px;
  padding:10px;
  background-color:lightgreen;
  list-style-type: none;
}
```

Output

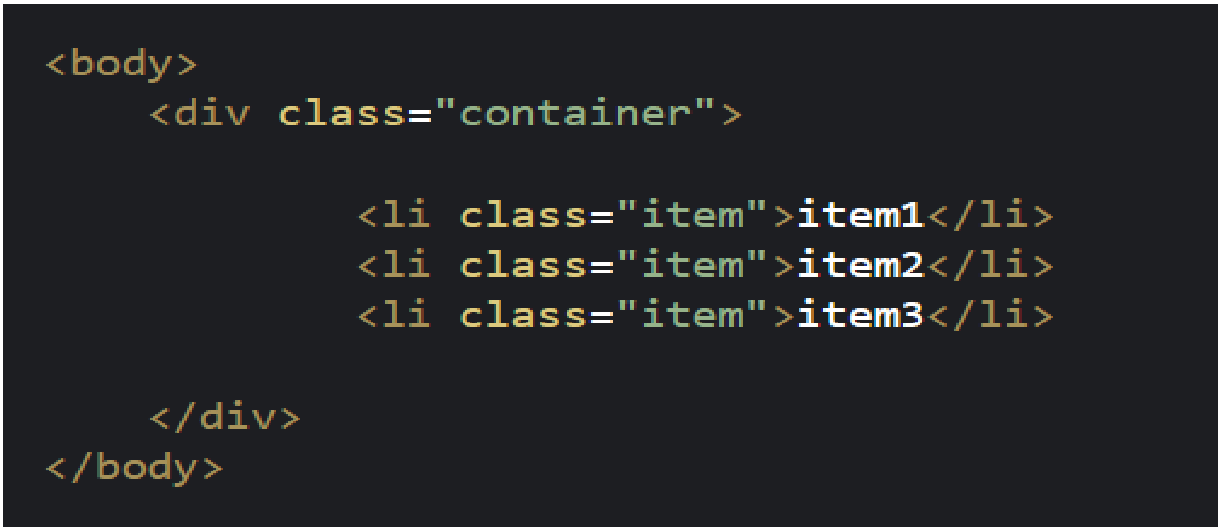


3| Flex-direction: row-reverse;

By using the **row-reverse** property the content will start from the **main end** in the **opposite direction.**

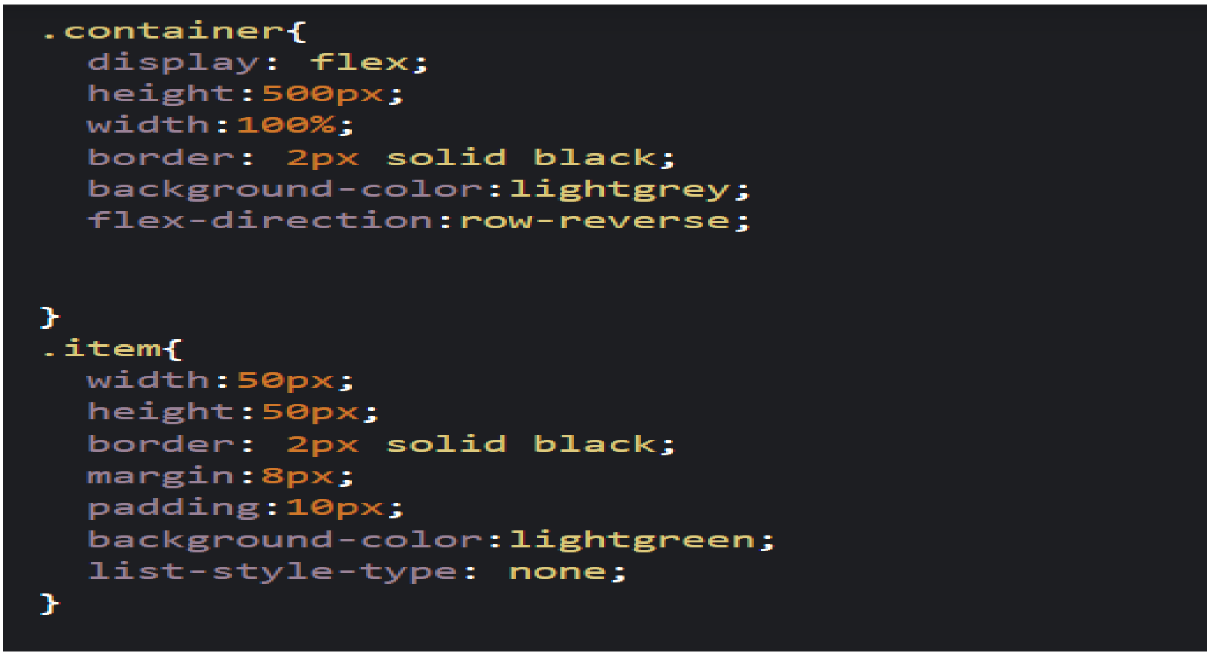
Let’s take an example:

HTML Code:



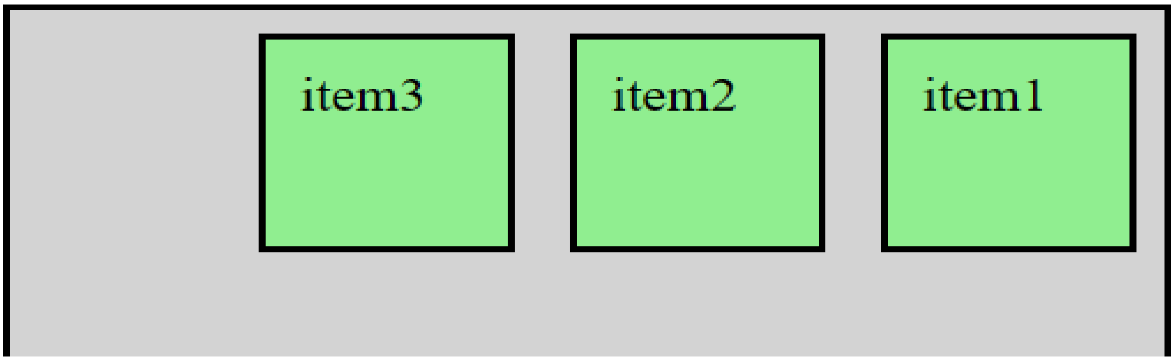
```html
<body>
    <div class="container">

            <li class="item">item1</li>
            <li class="item">item2</li>
            <li class="item">item3</li>

    </div>
</body>
```

CSS Code:



```css
.container{
  display: flex;
  height:500px;
  width:100%;
  border: 2px solid black;
  background-color:lightgrey;
  flex-direction:row-reverse;


}
.item{
  width:50px;
  height:50px;
  border: 2px solid black;
  margin:8px;
  padding:10px;
  background-color:lightgreen;
  list-style-type: none;
}
```

Output

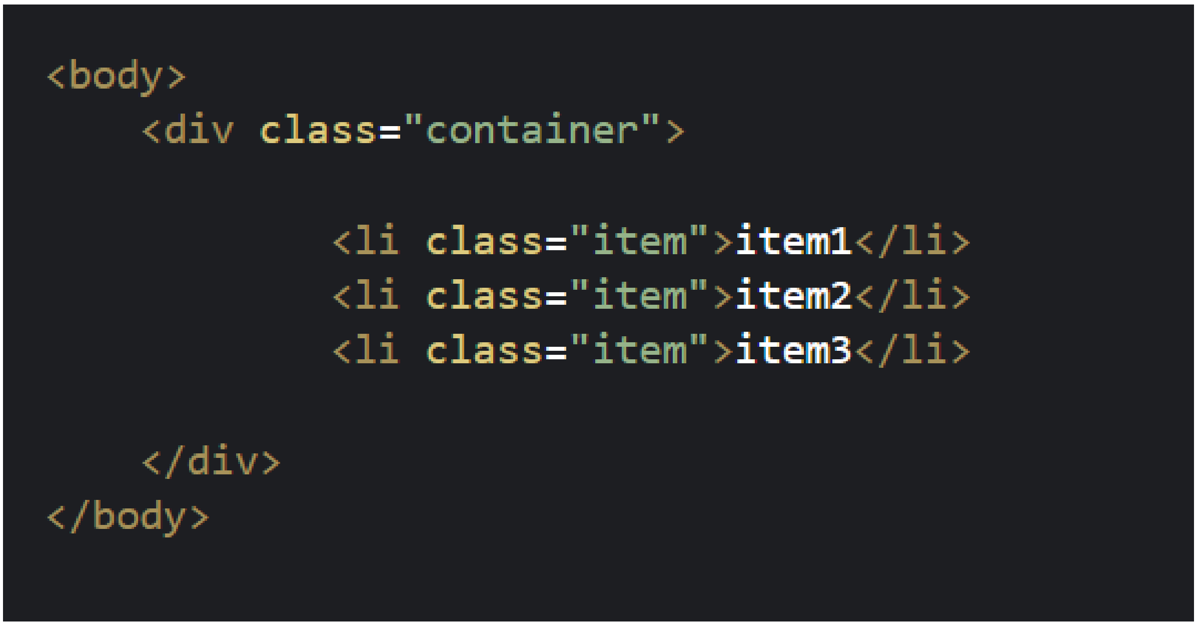


4| flex-direction: column-reverse;

By using the **column-reverse** property the content will start from the **bottom main end** in the **opposite direction.**

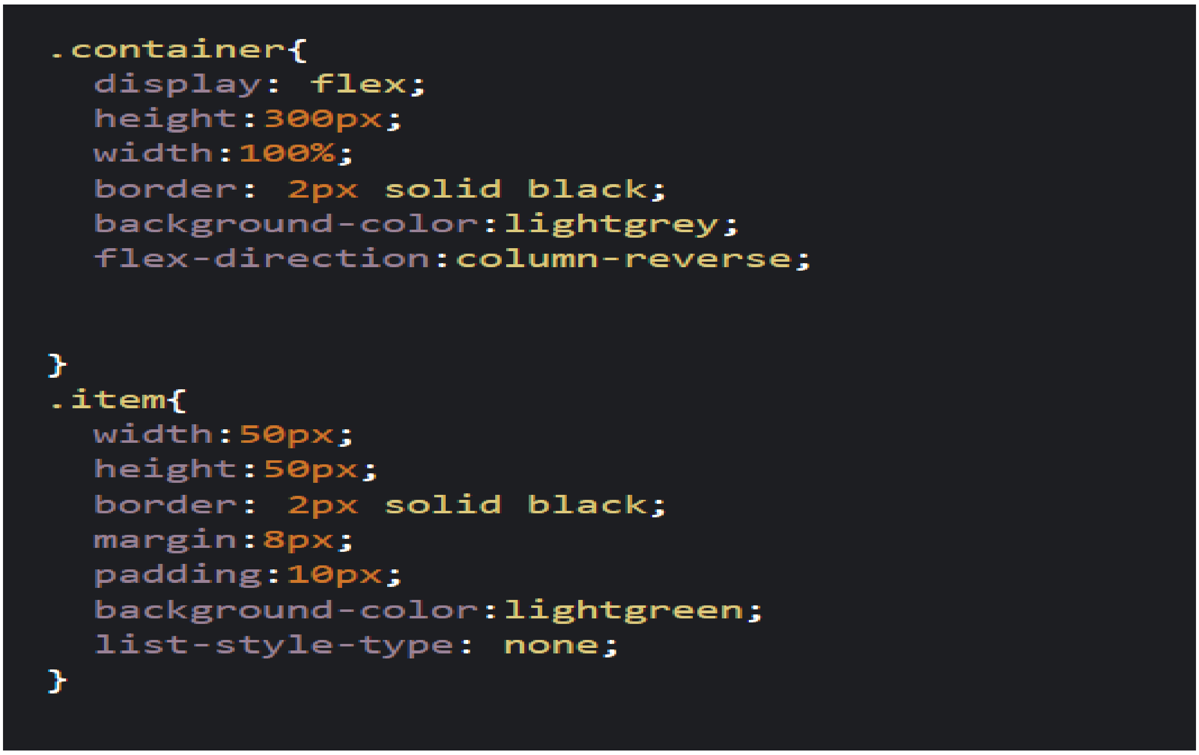
Let’s take an example:

HTML Code:



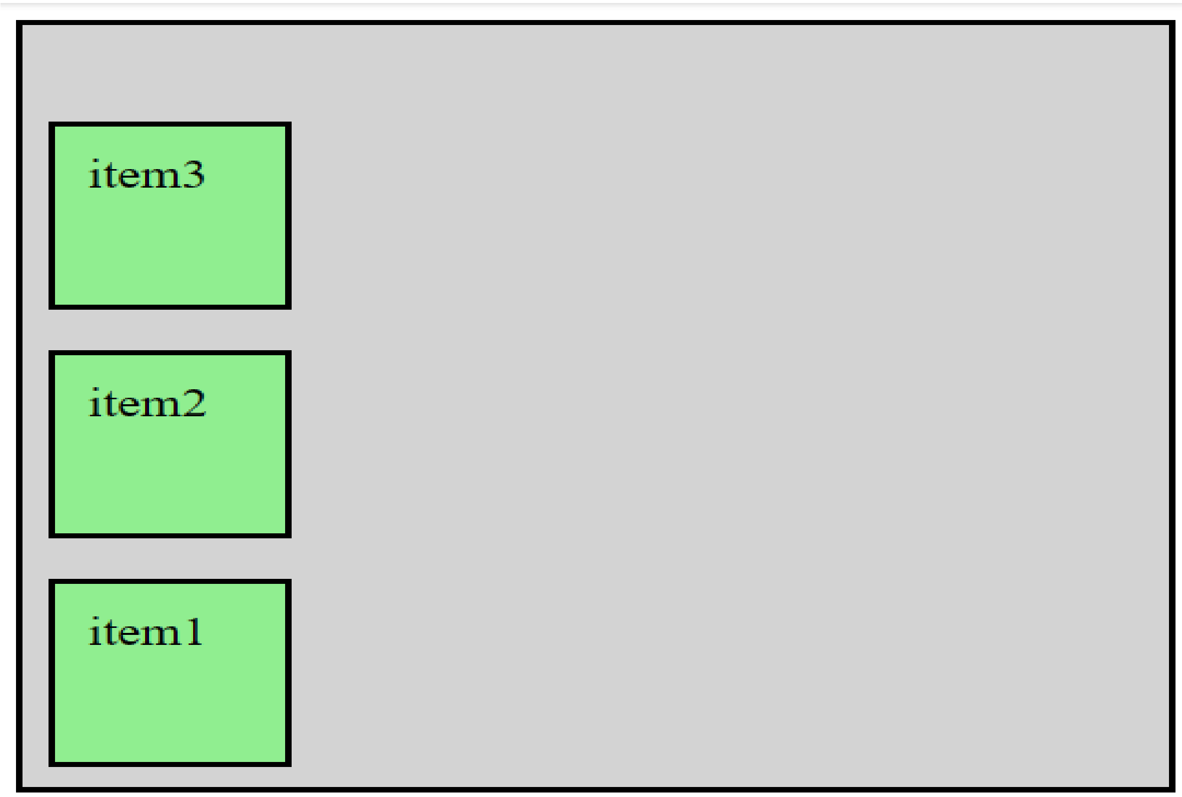
```
<body>
    <div class="container">

            <li class="item">item1</li>
            <li class="item">item2</li>
            <li class="item">item3</li>

    </div>
</body>
```

CSS Code:



```
.container{
  display: flex;
  height:300px;
  width:100%;
  border: 2px solid black;
  background-color:lightgrey;
  flex-direction:column-reverse;


}
.item{
  width:50px;
  height:50px;
  border: 2px solid black;
  margin:8px;
  padding:10px;
  background-color:lightgreen;
  list-style-type: none;
}
```

Output:



◆Flex properties for Parent (Flex Container).

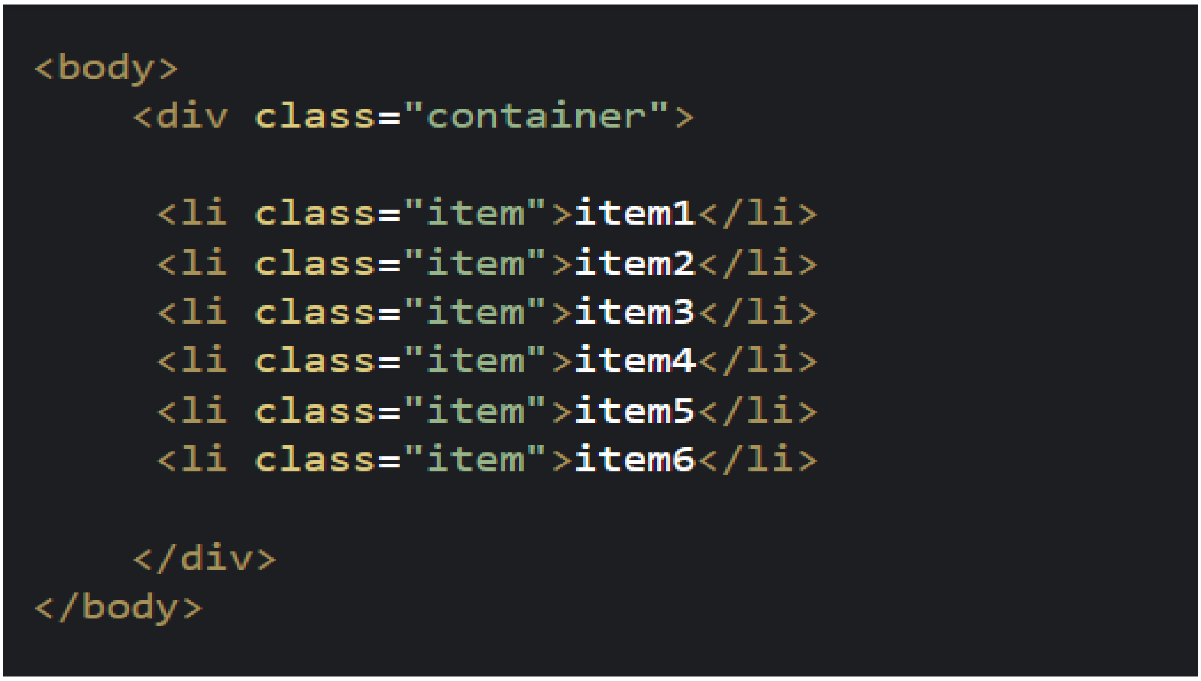
There are the following properties for flex parent.

1| Flex-wrap:

The CSS flex-wrap property is used to specify whether **flex items are forced into a single line or wrapped onto multiple lines.** It can be a wrap, no-wrap, wrap-reverse, or wrap-items as needed with these properties. By default it is no-wrap.

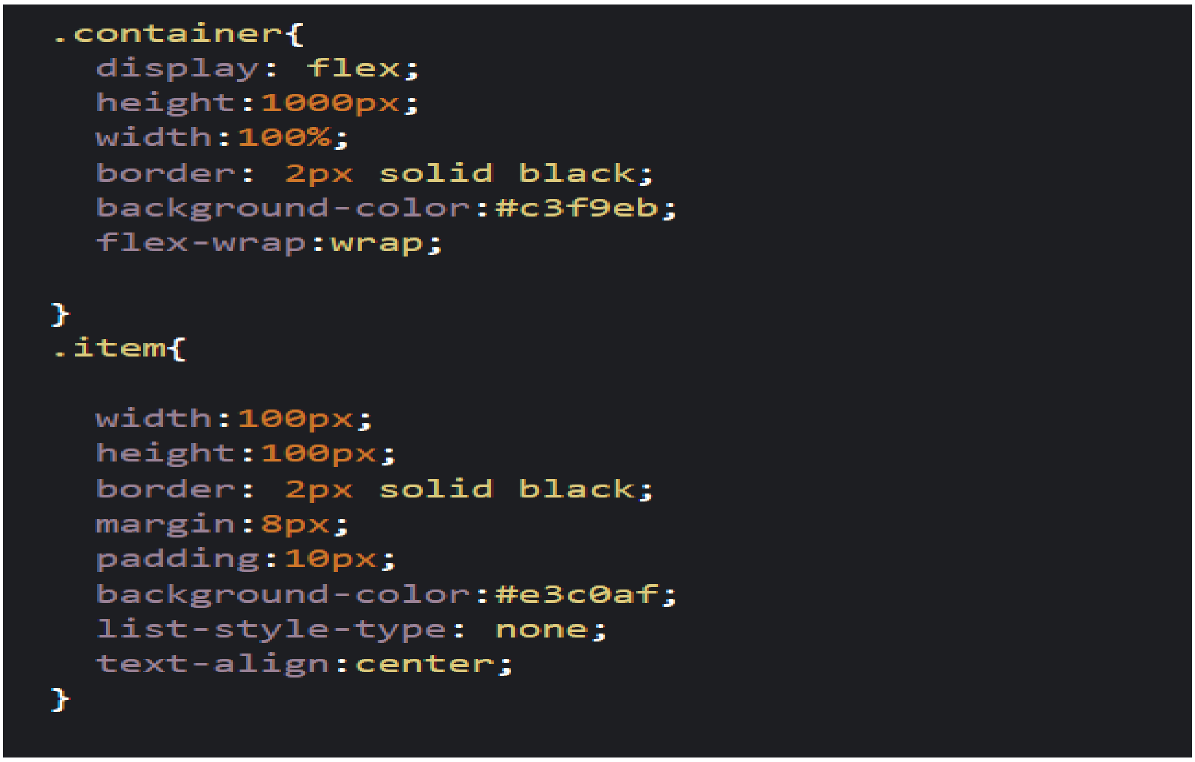
Let’s take an example:

HTML Code:



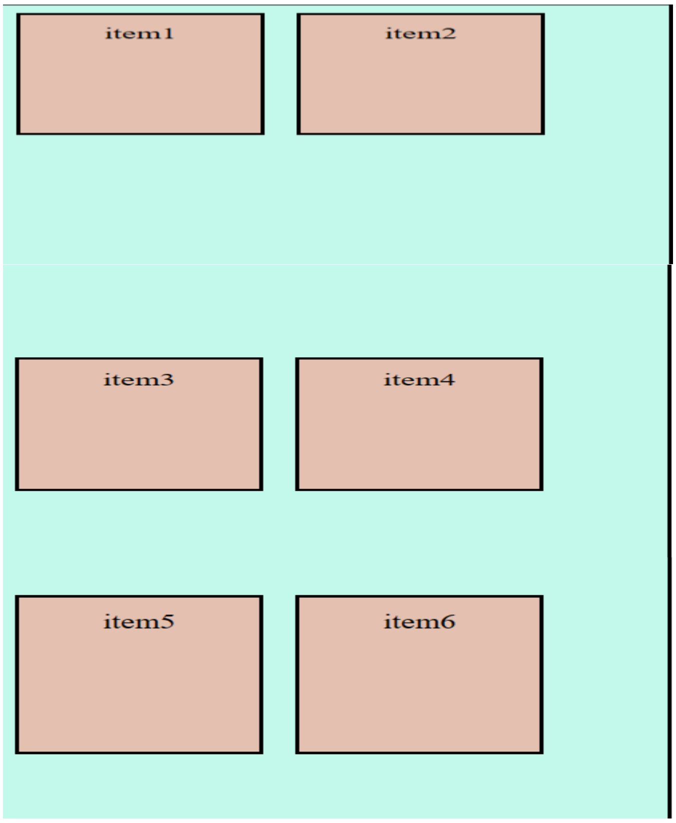
```
<body>
    <div class="container">

     <li class="item">item1</li>
     <li class="item">item2</li>
     <li class="item">item3</li>
     <li class="item">item4</li>
     <li class="item">item5</li>
     <li class="item">item6</li>

    </div>
</body>
```

CSS Code:



```
.container{
  display: flex;
  height:1000px;
  width:100%;
  border: 2px solid black;
  background-color:#c3f9eb;
  flex-wrap:wrap;

}
.item{

  width:100px;
  height:100px;
  border: 2px solid black;
  margin:8px;
  padding:10px;
  background-color:#e3c0af;
  list-style-type: none;
  text-align:center;
}
```

Output:



2| Justify-content:

It defines alignment along the main axis, or we can say that the **justify-content** property **in CSS** is used to describe the alignment of the flexible box container.

Justify content holds many properties

→**Justify-content: flex-start** (items aligned from the start of flex-direction.)

→**Justify-content: flex-end** (items aligned from the end of flex-direction.)

→**Justify-content: centre** (items are cantered in the horizontal axis.)

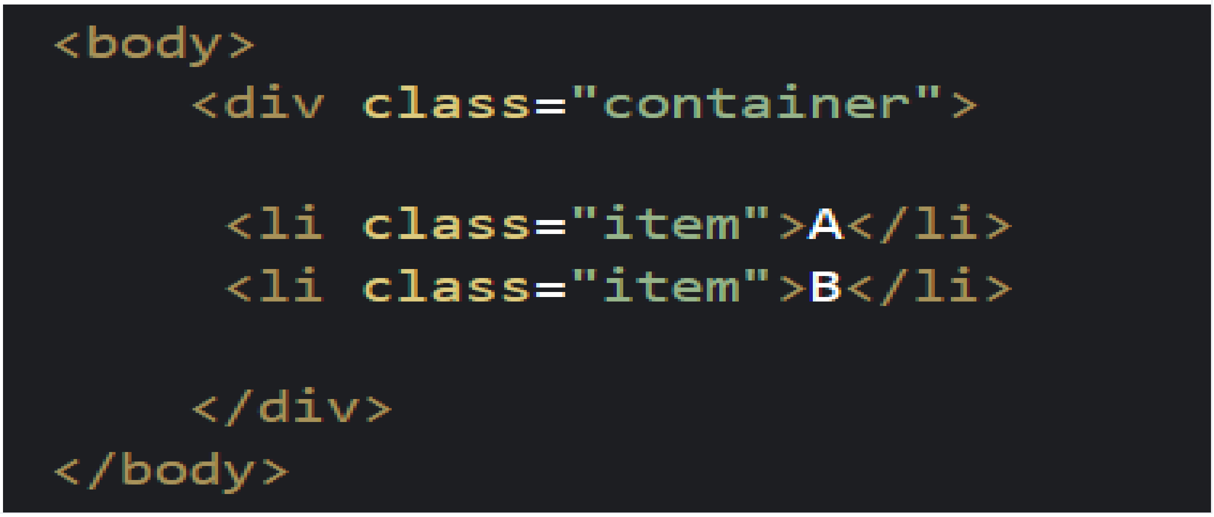
→**Justify-content: space-around** (items are evenly distributed in the line with an equal amount of space on both the sides of item.)

→**Justify-content: space-between** (items are evenly distributed in the line with space between each item.)

→**Justify-content: space-evenly** (items are evenly distributed so that space between two items and with the edge of the line is equal.)

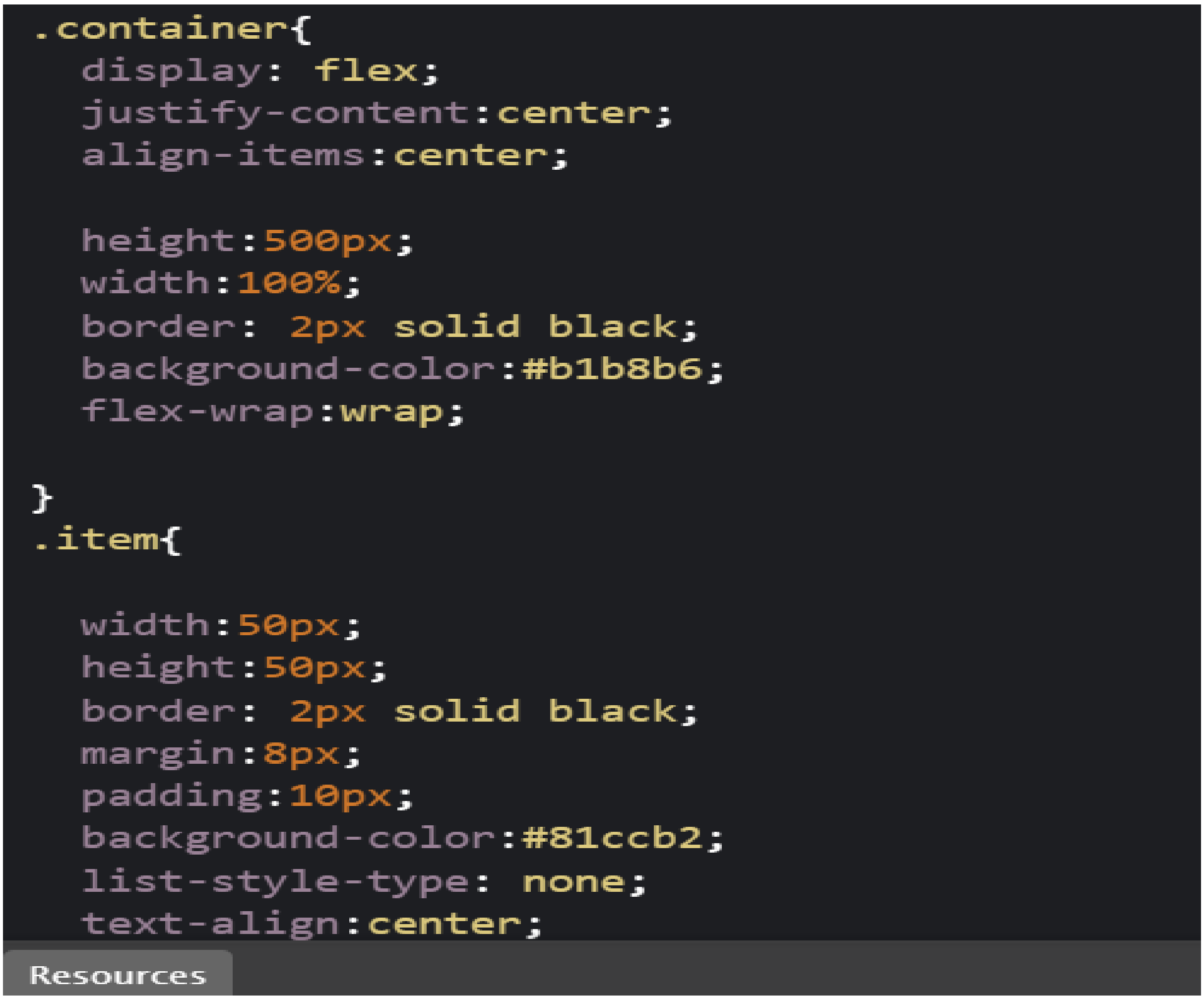
Let's take an example of **justify-content: centre**

HTML Code:



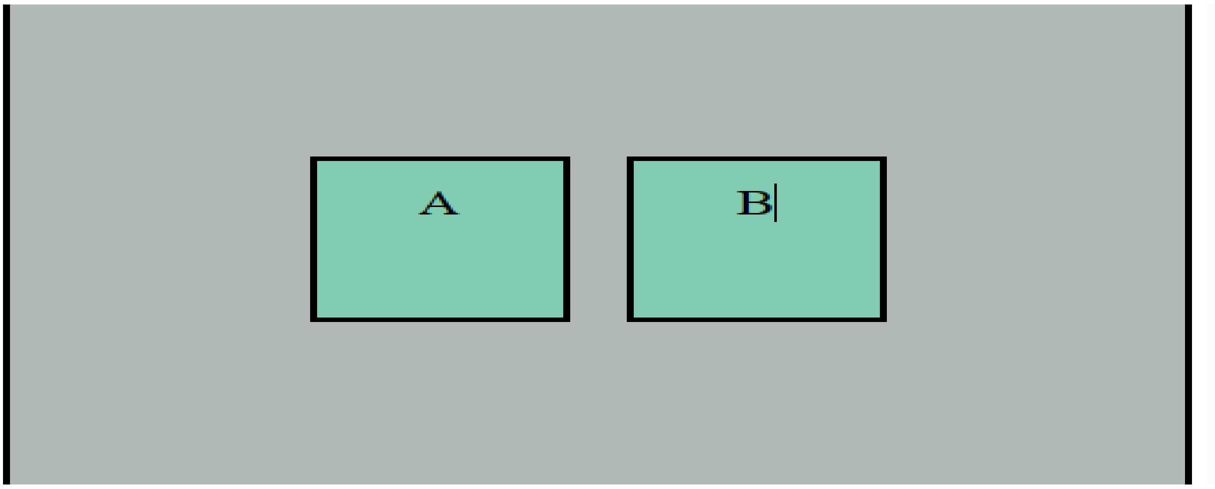
```
<body>
    <div class="container">

     <li class="item">A</li>
     <li class="item">B</li>

    </div>
</body>
```

CSS Code:



```
.container{
   display: flex;
   justify-content:center;
   align-items:center;

   height:500px;
   width:100%;
   border: 2px solid black;
   background-color:#b1b8b6;
   flex-wrap:wrap;

}
.item{

   width:50px;
   height:50px;
   border: 2px solid black;
   margin:8px;
   padding:10px;
   background-color:#81ccb2;
   list-style-type: none;
   text-align:center;
```

Output:

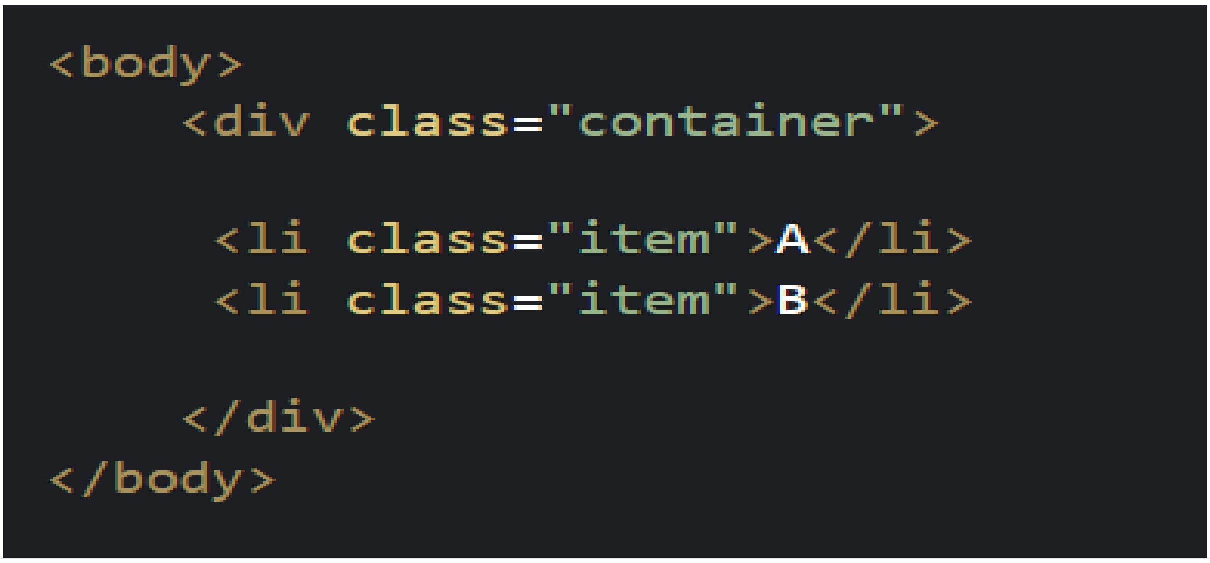


3| Align-items:

This defines the default behaviour for how flex items are laid out along the vertical axis of the current line.

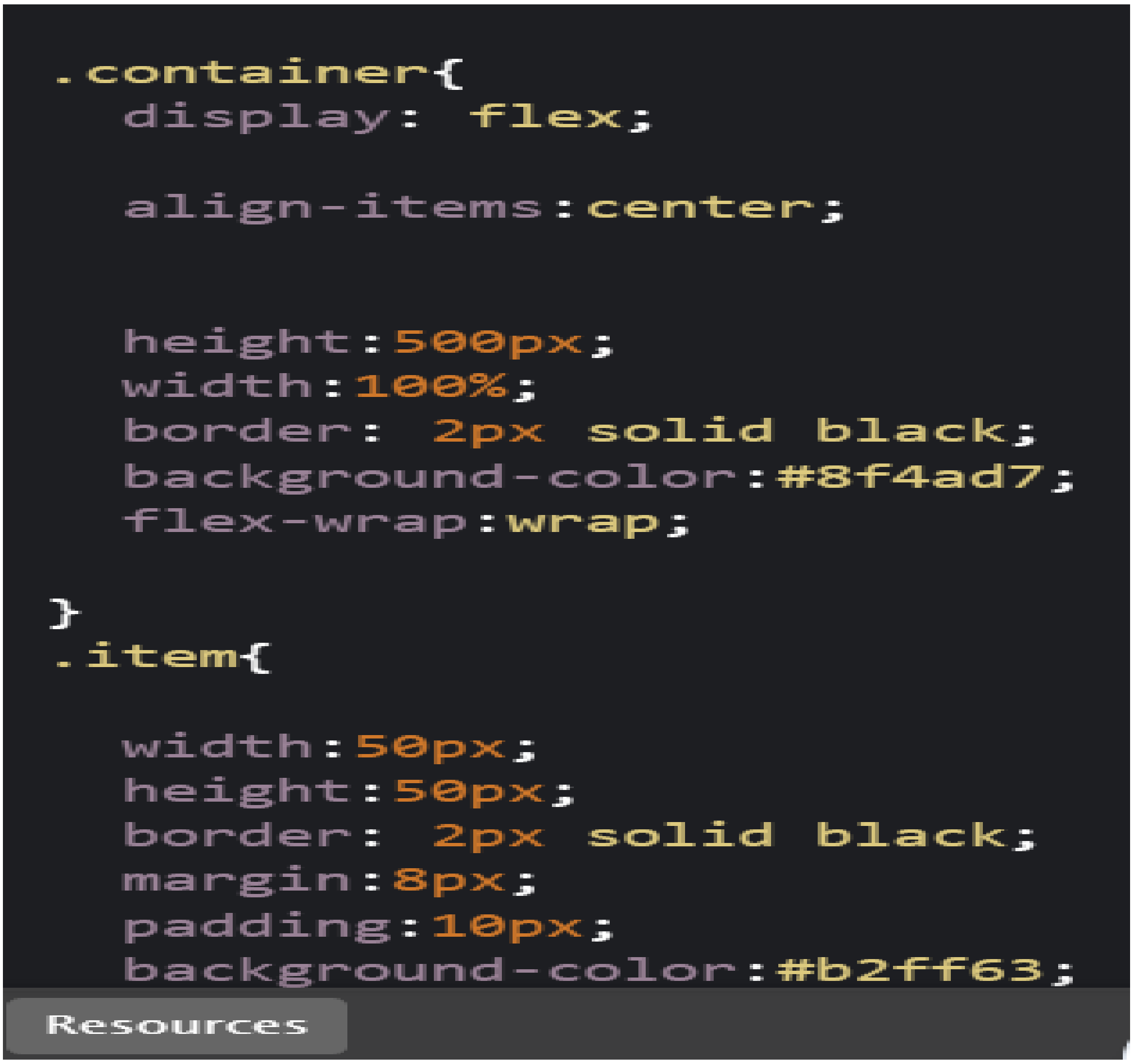
Let's take an example:

HTML Code:



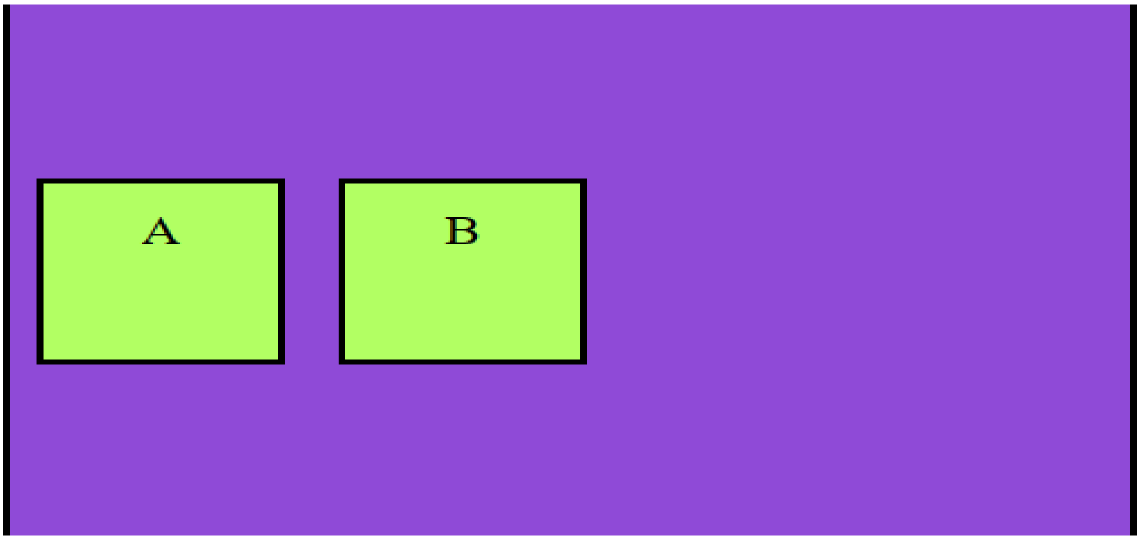
```
<body>
    <div class="container">

     <li class="item">A</li>
     <li class="item">B</li>

    </div>
</body>
```

CSS Code:



```
.container{
   display: flex;

   align-items:center;


   height:500px;
   width:100%;
   border: 2px solid black;
   background-color:#8f4ad7;
   flex-wrap:wrap;

}
.item{

   width:50px;
   height:50px;
   border: 2px solid black;
   margin:8px;
   padding:10px;
   background-color:#b2ff63;
```

Output:

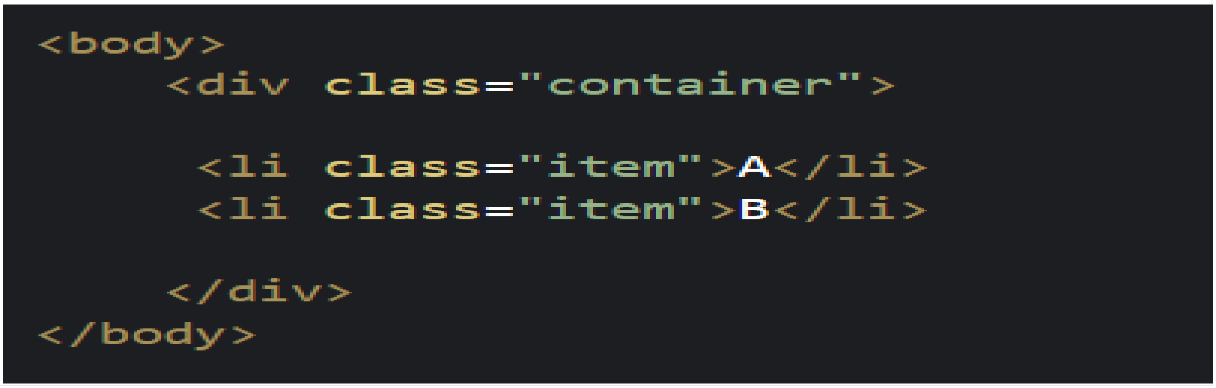


4| Align-content:

Aligns a flex container's lines when there is extra space in the cross-axis.

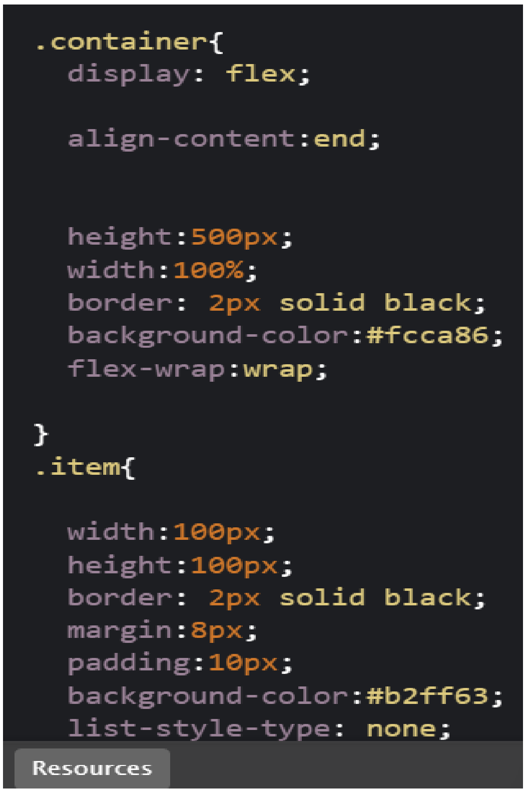
Let’s take an example:

HTML Code:



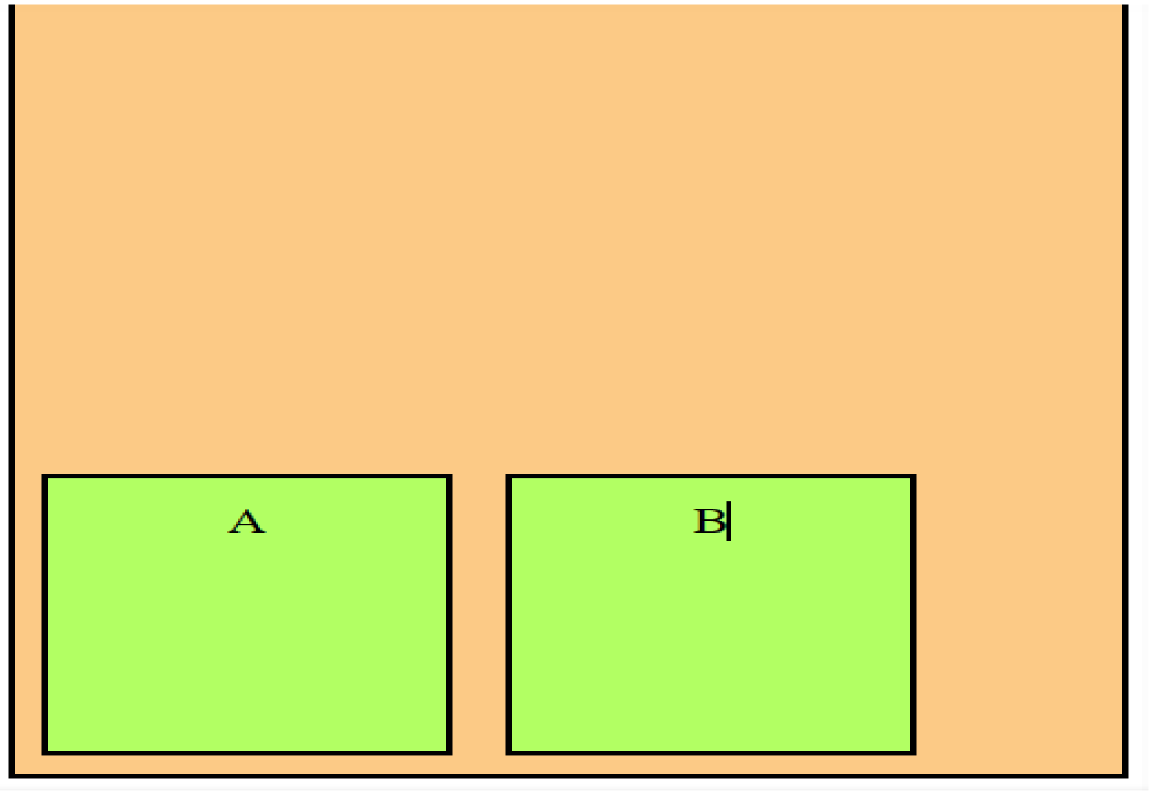
```
<body>
    <div class="container">

      <li class="item">A</li>
      <li class="item">B</li>

    </div>
</body>
```

CSS Code:



```
.container{
  display: flex;

  align-content:end;


  height:500px;
  width:100%;
  border: 2px solid black;
  background-color:#fcca86;
  flex-wrap:wrap;

}
.item{

  width:100px;
  height:100px;
  border: 2px solid black;
  margin:8px;
  padding:10px;
  background-color:#b2ff63;
  list-style-type: none;
```

Output:



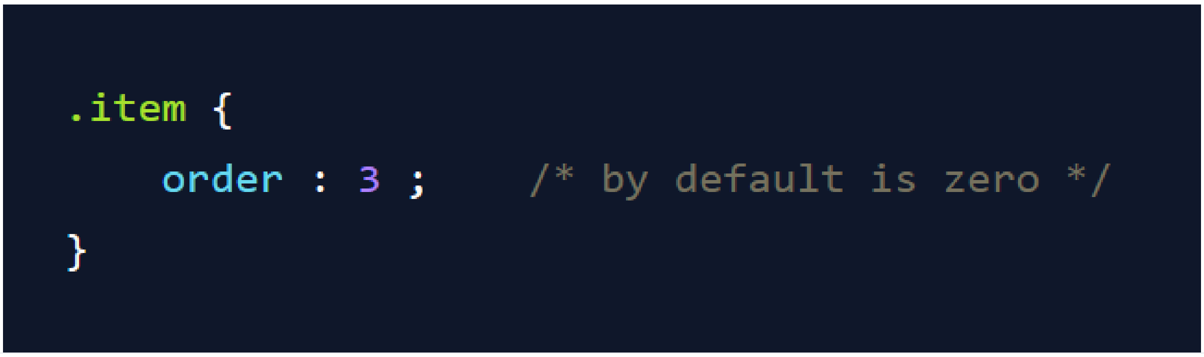
◈Flex properties for the children (Flex items).

Following are the properties for the flex children.

1| Order:

It controls the order in which the items appear in the flex container.

Syntax:

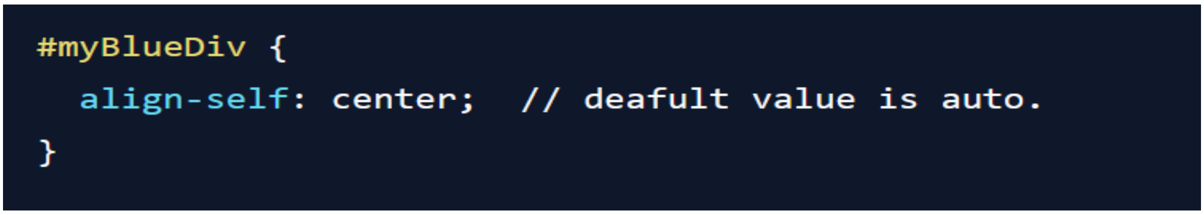


```
.item {
    order : 3 ;     /* by default is zero */
}
```

2| Align-self:

It allows default alignment to be overridden for the individual flex items.

Syntax:

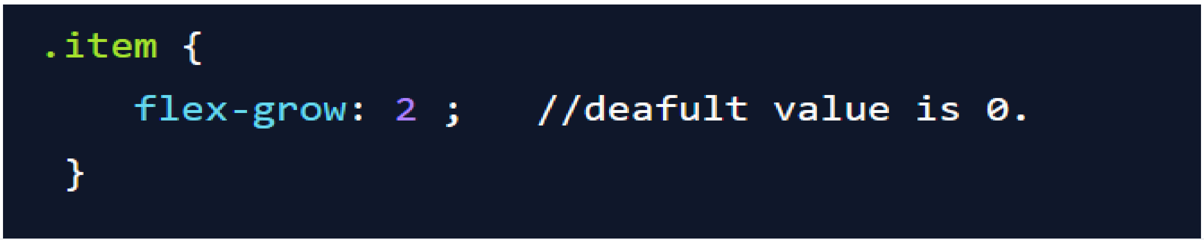


```
#myBlueDiv {
  align-self: center;  // deafult value is auto.
}
```

3| Flex-grow:

Defines the ability for a flex item to grow.

Syntax:

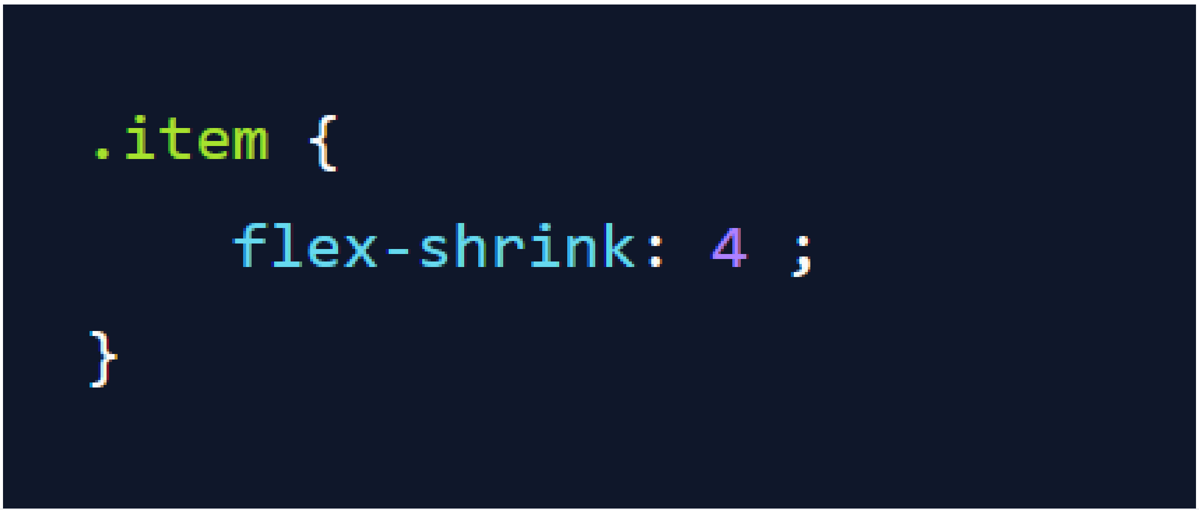


```
.item {
    flex-grow: 2 ;   //deafult value is 0.
 }
```

4| Flex-shrink:

It specifies how much flex items will shrink relative to the rest of the flex items.

Syntax:



```
.item {
    flex-shrink: 4 ;
}
```

The flex box holds many feature but these all features are the main features. Now we have some idea about Flex box in CSS.

Thank You ☺