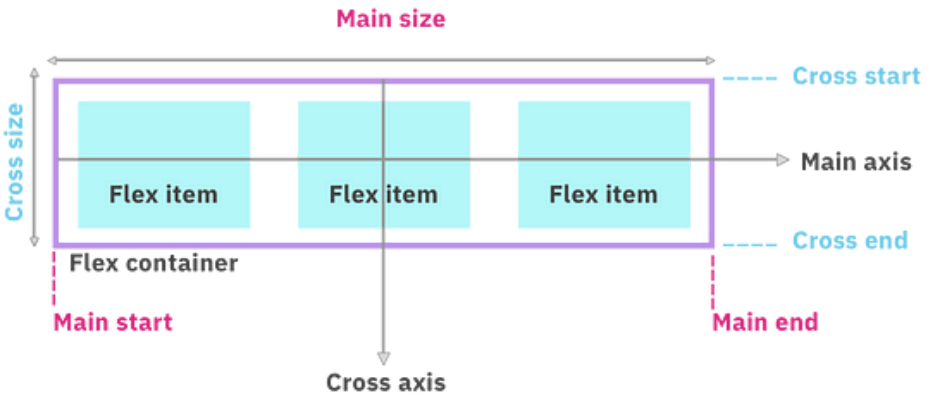
Flexbox, or CSS3 Flexible Box, is a super lightweight way to lay out UI components on a web page. It makes designing for responsive screen sizes painless .

The flexible box layout module, usually referred to as flexbox, was designed as a one-dimensional layout model, and as a method that could offer space distribution between items in an interface and powerful alignment capabilities.

Flexbox Terminology



Main axis: This is the primary axis along which flex items are distributed. This will change depending upon the value of flex-direction.

Cross axis: This is the axis that is perpendicular to the main axis. The direction depends on the main axis direction.

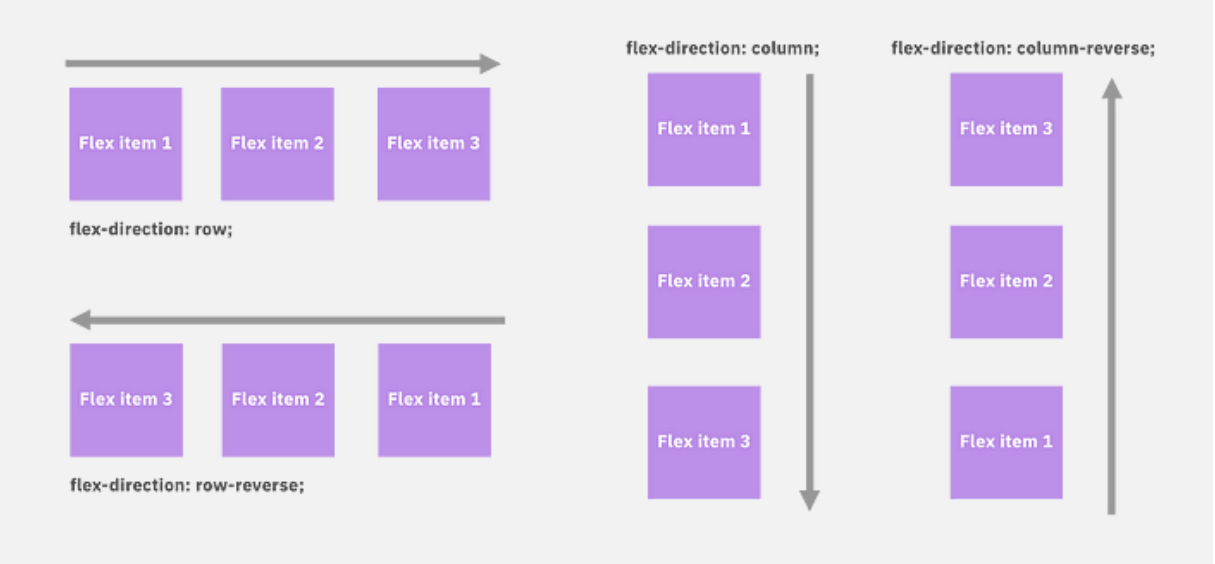
Flex Container Properties

display: Defines a flex container. It can be inline, or block, depending on the value.

```
.container {

 display: flex

}
```

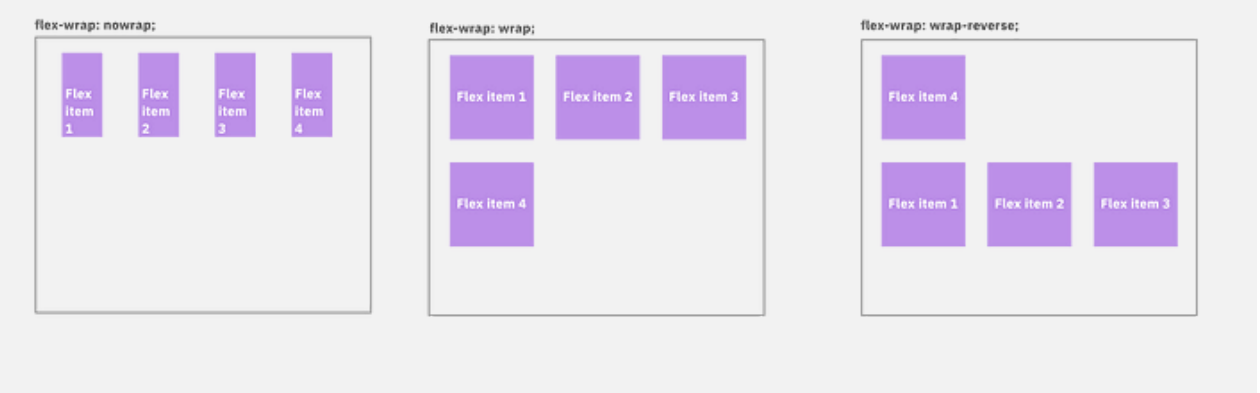
flex-direction: Establishes the main-axis, defining the direction flex items are placed in your flex container.

```
.container {

 flex-direction: row | row-reverse | column | column-reverse;

}
```



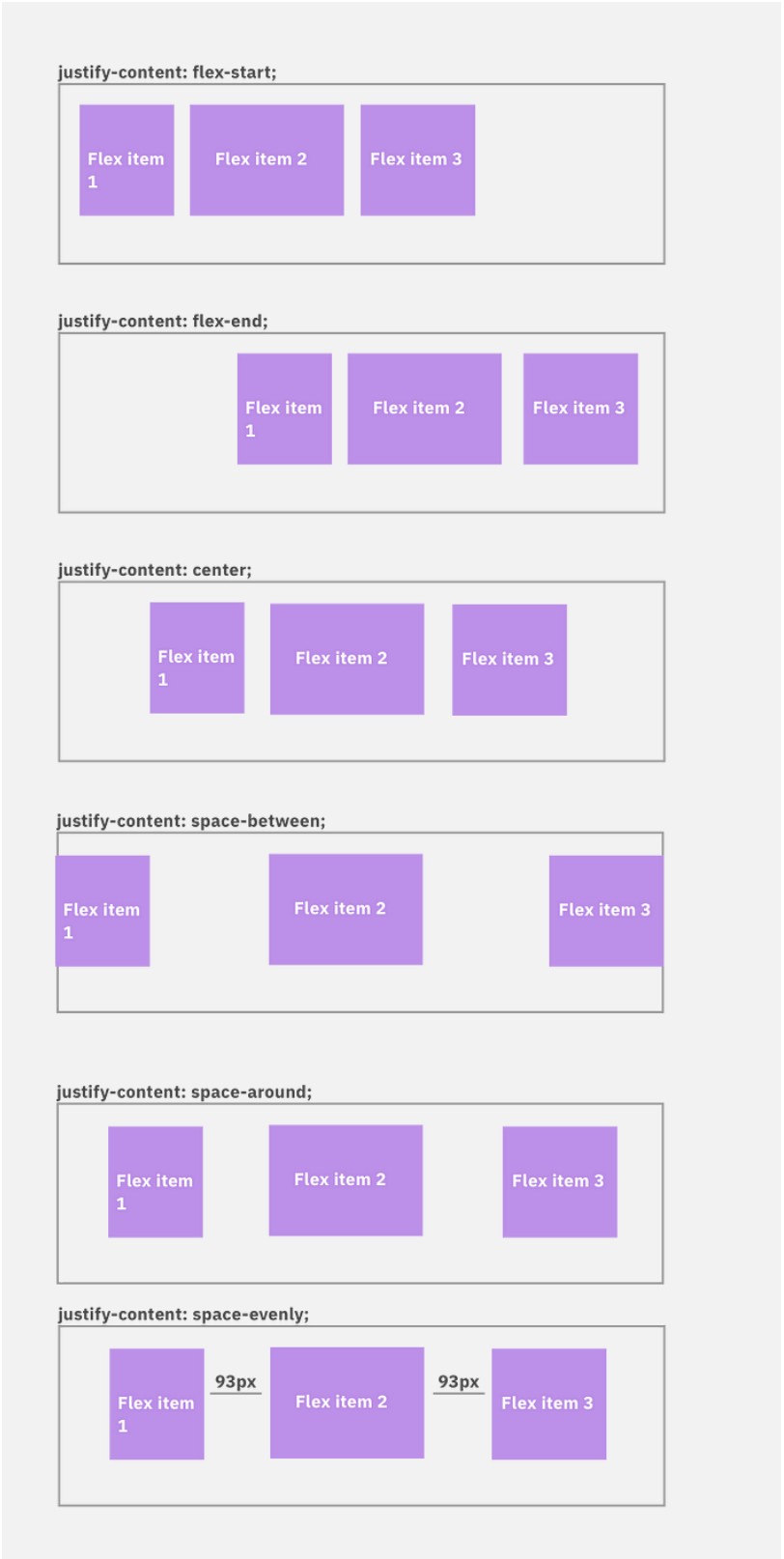
flex-wrap: Normally, flex items will attempt to fit onto one line. This can be changed such that flex items start wrapping to a new line when the width decreases.

```
.container {

 flex-wrap: nowrap | wrap | wrap-reverse;

}
```



justify-content: This defines the alignment of flex items along the main axis. It helps the developer distribute the extra white space.

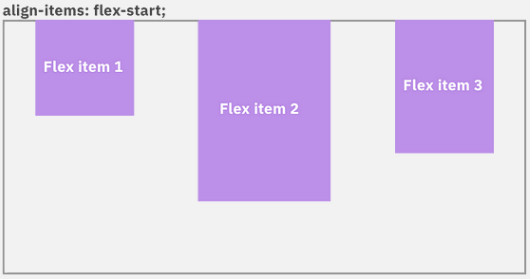
```
.container {

 justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;

}
```



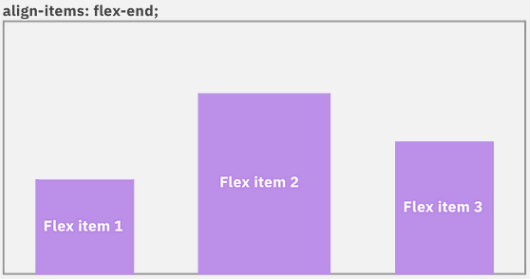
align-items: This defines how flex items should be laid out along the cross-axis.

```
.container {

 align-items: flex-start | flex-end | center | baseline | stretch;

}
```

align-items: flex-start;

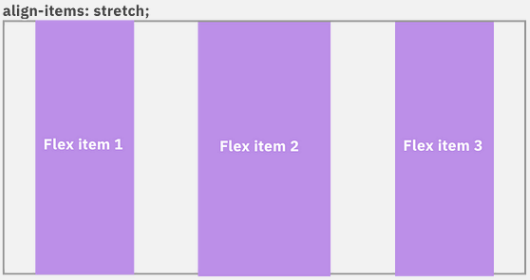


align-items: flex-end;

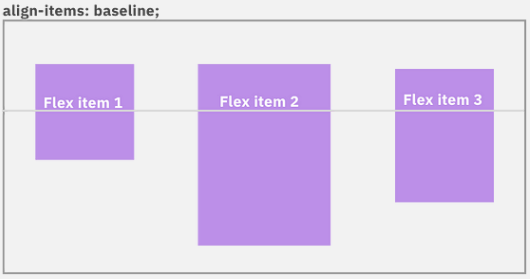


align-items: center;

align-items: stretch;



align-items: baseline;

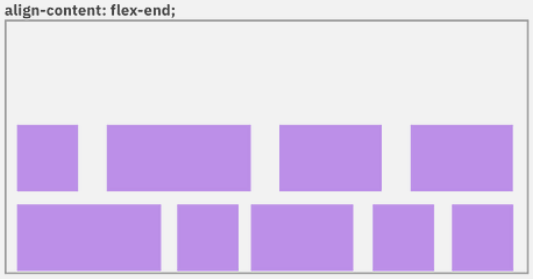


align-content: This aligns a flex container's internal boundaries when there is extra white space in the cross-axis. This won't affect the content when there is one single line.

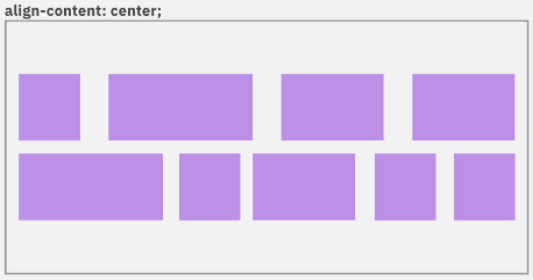
```
.container {

 align-content: flex-start | flex-end | center |
space-between | space-around;

}
```

align-content: flex-start;

align-content: flex-end;



align-content: center;



align-content: space-between;

align-content: space-around;