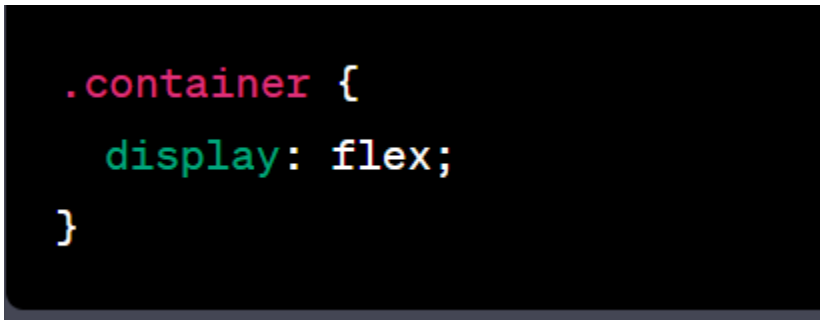
Flexbox is a layout mode in CSS that provides a more efficient way to design and align items within a container. With Flexbox, you can easily adjust the layout and positioning of elements, regardless of their size or content.

To use Flexbox, you first need to define a container element, and then specify the layout and alignment properties for the items within that container. Here are the main properties you can use:

display: flex;

This property turns the container element into a flex container and enables the use of flex properties.

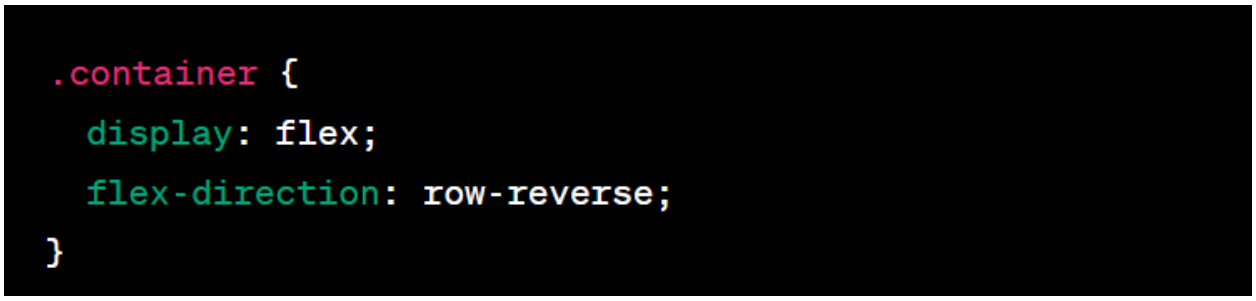
Example:



Flex-direction:

This property specifies the direction in which the items within the container should be laid out. It can take one of four values: row (default), row-reverse, column, or column-reverse.

Example:



justify-content:

This property aligns the items along the main axis (horizontal for row and row-reverse, and vertical for column and column-reverse). It can take one of five values: flex-start (default), flex-end, center, space-between, or space-around.

Example:

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

align-items:

This property aligns the items along the cross-axis (vertical for row and row-reverse, and horizontal for column and column-reverse). It can take one of five values: stretch (default), flex-start, flex-end, center, or baseline.

Example:

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

flex-wrap:

This property specifies whether the items should wrap to the next line when they exceed the width or height of the container. It can take one of three values: nowrap (default), wrap, or wrap-reverse.

Example:

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

Flex-flow:

This property is a shorthand for flex-direction and flex-wrap.

Example:

```
.container {
  display: flex;
  flex-flow: column wrap;
}
```

align-content:

This property aligns the lines of items when there is extra space in the cross-axis. It can take one of six values: stretch (default), flex-start, flex-end, center, space-between, or space-around.

Example:

```
.container {
  display: flex;
  align-content: space-around;
}
```

order:

This property specifies the order in which the items should be displayed within the container. The default value is 0.

Example:

```
.item {
  order: 2;
}
```

flex-grow:

This property specifies how much the item should grow relative to the other items in the container. The default value is 0.

Example:

```
.item {
  flex-grow: 2;
}
```

flex-shrink:

This property specifies how much the item should shrink relative to the other items in the container. The default value is 1.

Example:

```
.item {
  flex-shrink: 1;
}
```

flex-basis:

This property specifies the initial size of the item before any remaining space is distributed. The default value is auto.

Example:

```
.item {
  flex-basis: 50%;
}
```

flex:

This property is a shorthand for flex-grow, flex-shrink, and flex-basis.

Example:

```
.item {
  flex: 1 1 auto;
}
```