

# CSS Flexbox

Anthoniraj Amalanathan

# CSS Flexbox

CSS Flexbox, or Flexible Box Layout, is a layout model that allows you to create responsive and dynamic layouts with less code.

## 1. Flex Container (`display: flex;`)

To use Flexbox, you need to define a container as a flex container. You can do this by applying `display: flex;` to the container element. The `display: flex;` property makes all direct children of the container into flexible items.

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

## 2. Flex Direction (**flex-direction**)

The **flex-direction** property defines the direction in which the items within the flex container are placed. It can be set to one of the following values:

- **row** : Default value. Items are placed in the same direction as the text.
- **row-reverse** : Items are placed in the opposite direction as the text.
- **column** : Items are placed top to bottom.
- **column-reverse** : Items are placed bottom to top.

```
.container {  
  flex-direction: row; /* or column, row-reverse, column-reverse */  
}
```

### 3. Justify Content ( `justify-content` )

This property aligns items along the main axis of the container. It defines how space is distributed between and around the items. Common values include:

- `flex-start` : Items are packed at the start of the main axis.
- `flex-end` : Items are packed at the end of the main axis.
- `center` : Items are centered along the main axis.
- `space-between` : Items are evenly distributed with the first item at the start and the last item at the end.
- `space-around` : Items are evenly distributed with equal space around them.
- `space-evenly` : Items are evenly distributed with equal space around them, including the space before the first and after the last item.

```
.container {  
  justify-content: center; /* or flex-start, flex-end, space-between, space-around, space-evenly */  
}
```

## 4. Align Items (`align-items`)

This property aligns items along the cross-axis (perpendicular to the main axis). It determines how items are positioned in the flex container. Common values include:

- `flex-start` : Items are aligned at the start of the cross axis.
- `flex-end` : Items are aligned at the end of the cross axis.
- `center` : Items are centered along the cross axis.
- `baseline` : Items are aligned based on their baseline.
- `stretch` : Items are stretched to fill the container (default).

```
.container {  
  align-items: center; /* or flex-start, flex-end, baseline, stretch */  
}
```

## 5. Flex Property (`flex`)

The `flex` property is shorthand for three individual properties: `flex-grow`, `flex-shrink`, and `flex-basis`. It's used to specify how a flex item will grow or shrink to fit the available space.

```
.item {  
  flex: 1; /* flex-grow: 1, flex-shrink: 1, flex-basis: 0% */  
}
```