

Properties for the Parent

(flex container)

display

CSS

```
.container {  
  display: flex; /* or inline-flex */  
}
```

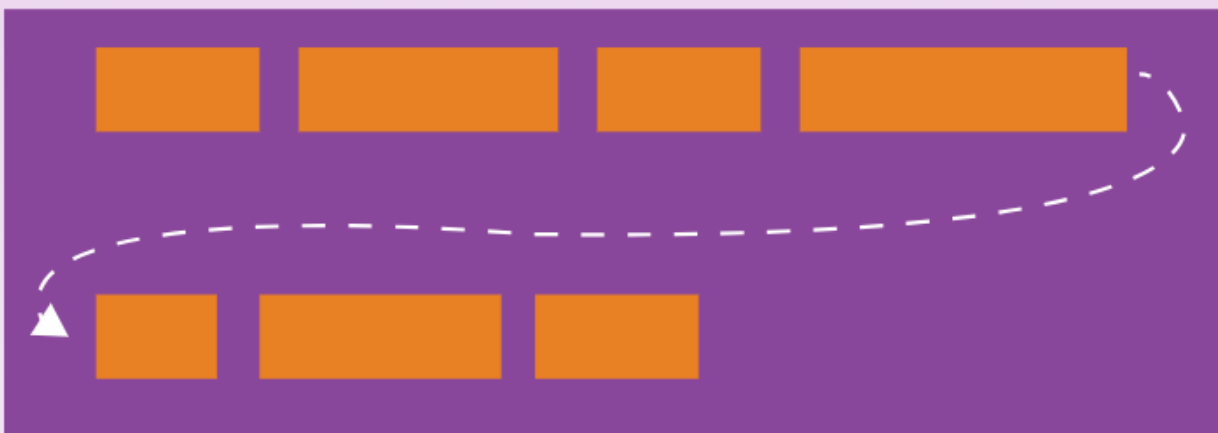
flex-direction



CSS

```
.container {  
  flex-direction: row | row-reverse | column | column-reverse;  
}
```

flex-wrap



CSS

```
.container{  
  flex-wrap: nowrap | wrap | wrap-reverse;  
}
```

flex-flow (Applies to: parent flex container element)

This is a shorthand `flex-direction` and `flex-wrap` properties, which together define the flex container's main and cross axes. Default is `row nowrap`.

CSS

```
flex-flow: <'flex-direction'> || <'flex-wrap'>
```

justify-content

flex-start



flex-end



center



space-between



space-around



space-evenly



CSS

```
.container {  
  justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly;  
}
```

align-items

flex-start



flex-end



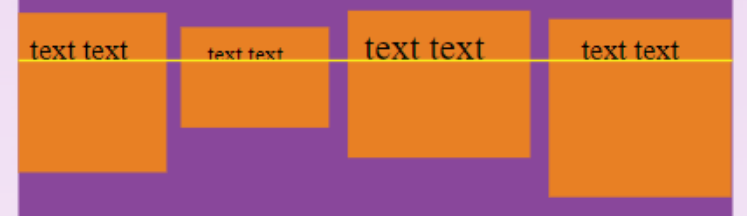
center



stretch



baseline



CSS

```
.container {  
  align-items: flex-start | flex-end | center | baseline | stretch;  
}
```

align-content

flex-start



flex-end



center



stretch



space-between



space-around



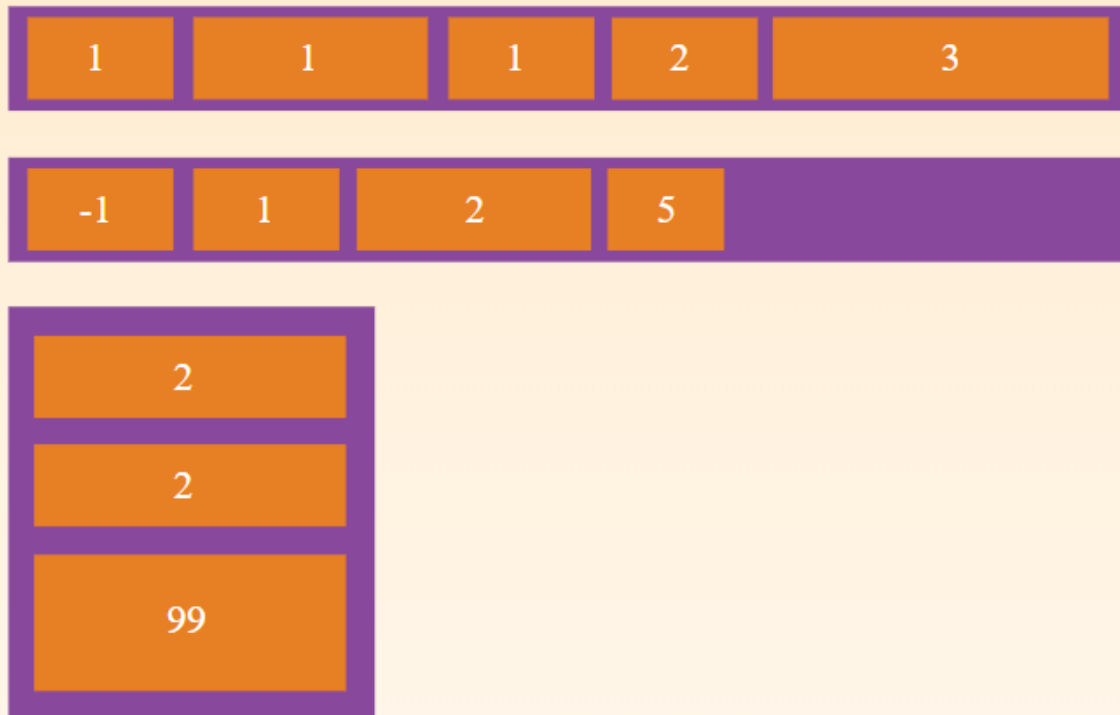
CSS

```
.container {  
  align-content: flex-start | flex-end | center | space-between | space-around | space-evenly;  
}
```

Properties for the Children

(flex items)

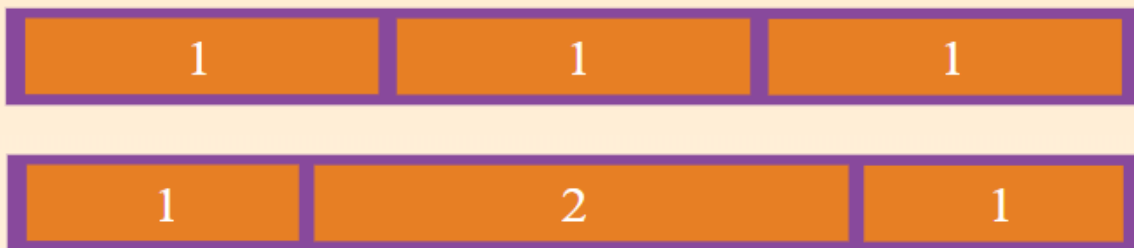
order



CSS

```
.item {
  order: <integer>; /* default is 0 */
}
```

flex-grow



CSS

```
.item {
  flex-grow: <number>; /* default 0 */
}
```

flex-shrink

CSS

```
.item {
  flex-shrink: <number>; /* default 1 */
}
```

flex-basis

This defines the default size of an element before the remaining space is distributed. It can be a length (e.g. 20%, 5rem, etc.) or a keyword. The `auto` keyword means "look at my width or height property" (which was temporarily done by the `main-size` keyword until deprecated). The `content` keyword means "size it based on the item's content" - this keyword isn't well supported yet, so it's hard to test and harder to know what its brethren `max-content`, `min-content`, and `fit-content` do.

CSS

```
.item {
  flex-basis: <length> | auto; /* default auto */
}
```

If set to `0`, the extra space around content isn't factored in. If set to `auto`, the extra space is distributed based on its `flex-grow` value.

flex

This is the shorthand for `flex-grow`, `flex-shrink` and `flex-basis` combined. The second and third parameters (`flex-shrink` and `flex-basis`) are optional. Default is `0 1 auto`.

CSS

```
.item {
  flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]
}
```

align-self

flex-start



flex-end

This allows the default alignment (or the one specified by `align-items`) to be overridden for individual flex items.

Please see the `align-items` explanation to understand the available values.

CSS

```
.item {
  align-self: auto | flex-start | flex-end | center | baseline | stretch
}
```