

## basic flexbox

- flex-direction: sets axis - row (default)/column

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

2

1

## wrapping

- flex-wrap: wrap/nowrap/wrap-reverse!

```
.container {  
  display: flex;  
  flex-wrap: wrap;  
}  
  
.item {  
  width: 200px;  
  margin: 10px;  
}
```

1

2

3

4

5

6

7

## ordering

- default is 0, so anything greater than 0 is at the end!

```
.container {
  display: flex;
}

.box {
  flex: 1;
}

.box-3 {
  order: -1;
}

.box-2 {
  order: 4;
}
```

3

1

4

2

## alignment and centering

- **justify** = row axis (left to right) => main axis
- **align** = column axis (top to bottom) => cross axis

justify-content

- flex-start (default), flex-end, center, space-between, space-around

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

1

2

3

4

align-items

- default set to stretch; center, flex-start, flex-end, baseline

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

1

2

3

4

### align-content

- stretch (default), flex-start, flex-end, center, space-between, space-around
- takes extra space on the cross axis (top-to-bottom) when wrapped

```
.container {  
  height: 250px;  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  align-content: space-around;  
}
```

1

2

3

4

5

6

7

8

9

### align-self

- applied on individual item
- center, flex-start, flex-end, baseline, stretch

```
.container {  
  height: 100px;  
  display: flex;  
  align-items: flex-start;  
}  
  
.item-2 {  
  align-self: flex-end;  
}
```

1

3

2

## flex

- goes on individual items
- shorthand: `flex: <flex-grow> <flex-shrink> <flex-basis>;`

```
.container {  
  display: flex;  
}  
  
.item-2 {  
  flex: 2;  
}
```

1

2

3

### flex-grow, flex-shrink, flex-basis

- how to divide extra space between elements
- flex-basis: size in an ideal world :)
- flex-grow - default 0; flex-shrink - default 1: how much to grow/shrink relative to other elements in case of too much/too little space

```

.container {
  display: flex;
}

.item-1 {
  flex: 0 1 200px
}

.item-2 {
  flex: 1 1 200px;
}

```

1

2

term	values	definition - <a href="#">main source</a> and <a href="#">here</a>
flex-direction	row, row-reverse, column, column-reverse	establishes main axis
flex-wrap	nowrap, wrap, wrap-reverse	allows items to wrap if needed
flex-flow		shorthand for flexdirection and flex-wrap
justify-content	flex-start, flex-end, left, right, center, space-between, space-around, space-evenly, start/end (writing mode direction)	defines the alignment along the main axis (Notice that when the flex direction is a column, justify-content changes to the vertical and align-items to the horizontal.)
align-items	flex-start, flex-end, center, baseline	how flex items are laid out along the cross axis on the current line (perpendicular to main axis)
align-content	flex-start, flex-end, center, space-between, space-around, space-evenly, stretch	aligns a flex container's lines within when there is extra space in the cross-axis. no effect if only one line of items
order	(default 0)	order of placement of items
flex-grow	(default 0)	dictates what amount of the available space inside the flex container the item should take up proportionally
flex-shrink	(default 1)	defines the ability for a flex item to shrink if necessary.

term	values	definition - <a href="#">main source</a> and <a href="#">here</a>
flex-basis	length/auto/content	defines the default size of an element before the remaining space is distributed.
flex		shorthand for flex-grow flex-shrink flex-basis (recommended)
align-self	same as align-items	allows the default alignment (or the one specified by align-items) to be overridden for individual flex items.