

# CSS Flexbox Cheat Sheet

Our guide for CSS Flexbox layout

## 1. a Flexbox intro

Flexbox is a great way to get more flexibility in your layouts and simplify responsiveness throughout your design. It provides a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and dynamic.

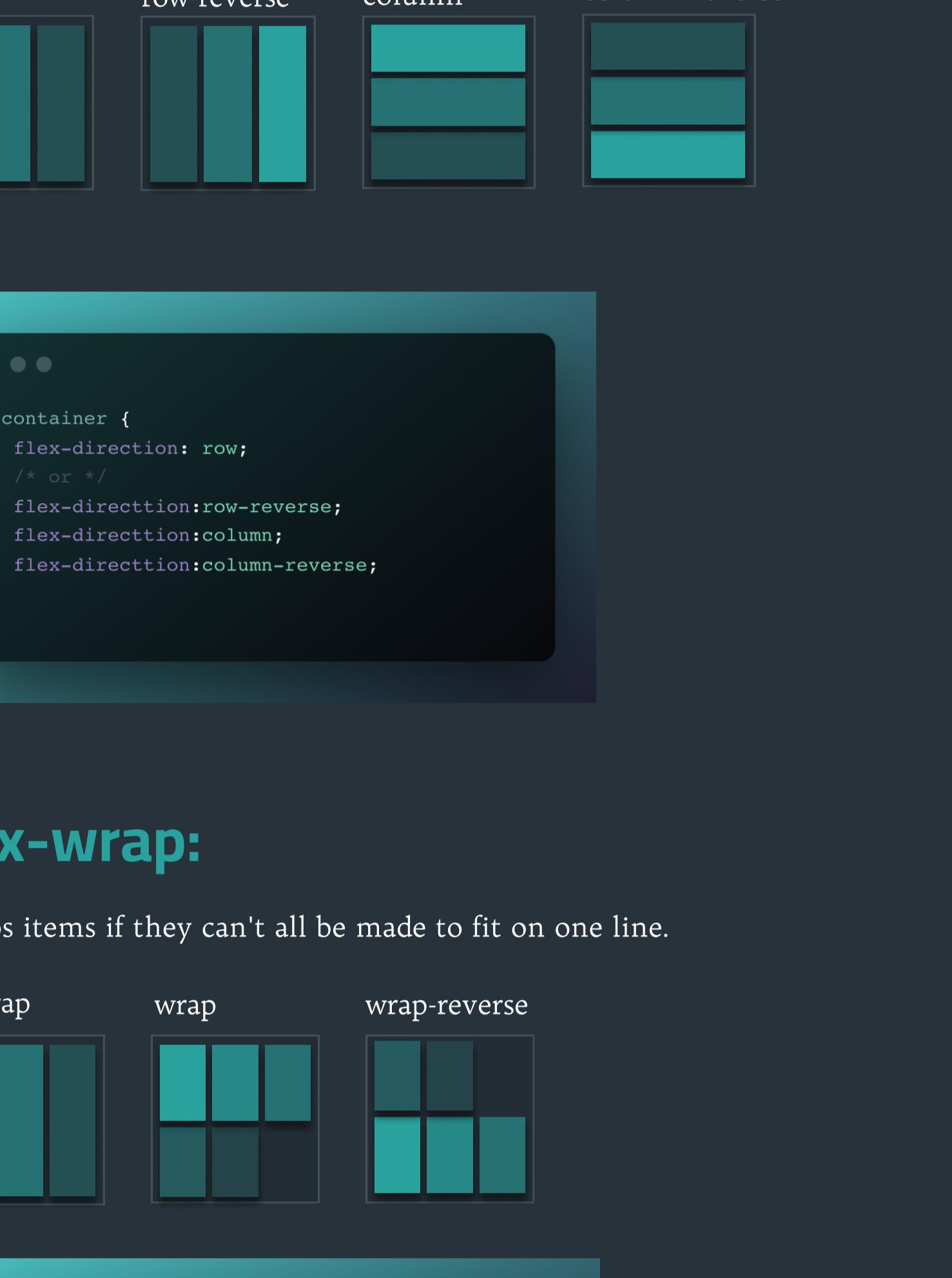
The main idea behind the flex layout is to give the container the ability to alter its items' width/height (and order) to best fill the available space (mostly to accommodate to all kind of display devices and screen sizes). A flex container expands items to fill available free space or shrinks them to prevent overflow.

## b Main axis and cross axis

Since flexbox is a whole module and not a single property, it involves a lot of things including its whole set of properties. Some of them are meant to be set on the container (parent element, known as "flex container") whereas the others are meant to be set on the children (said "flex items").

If "regular" layout is based on both block and inline flow directions, the flex layout is based on "flex-flow directions". Please have a look at this figure from the specification, explaining the main idea behind the flex layout.

Items will be laid out following either the main axis (from main-start to main-end) or the cross axis (from cross-start to cross-end).



## c Flexbox vs Grid:

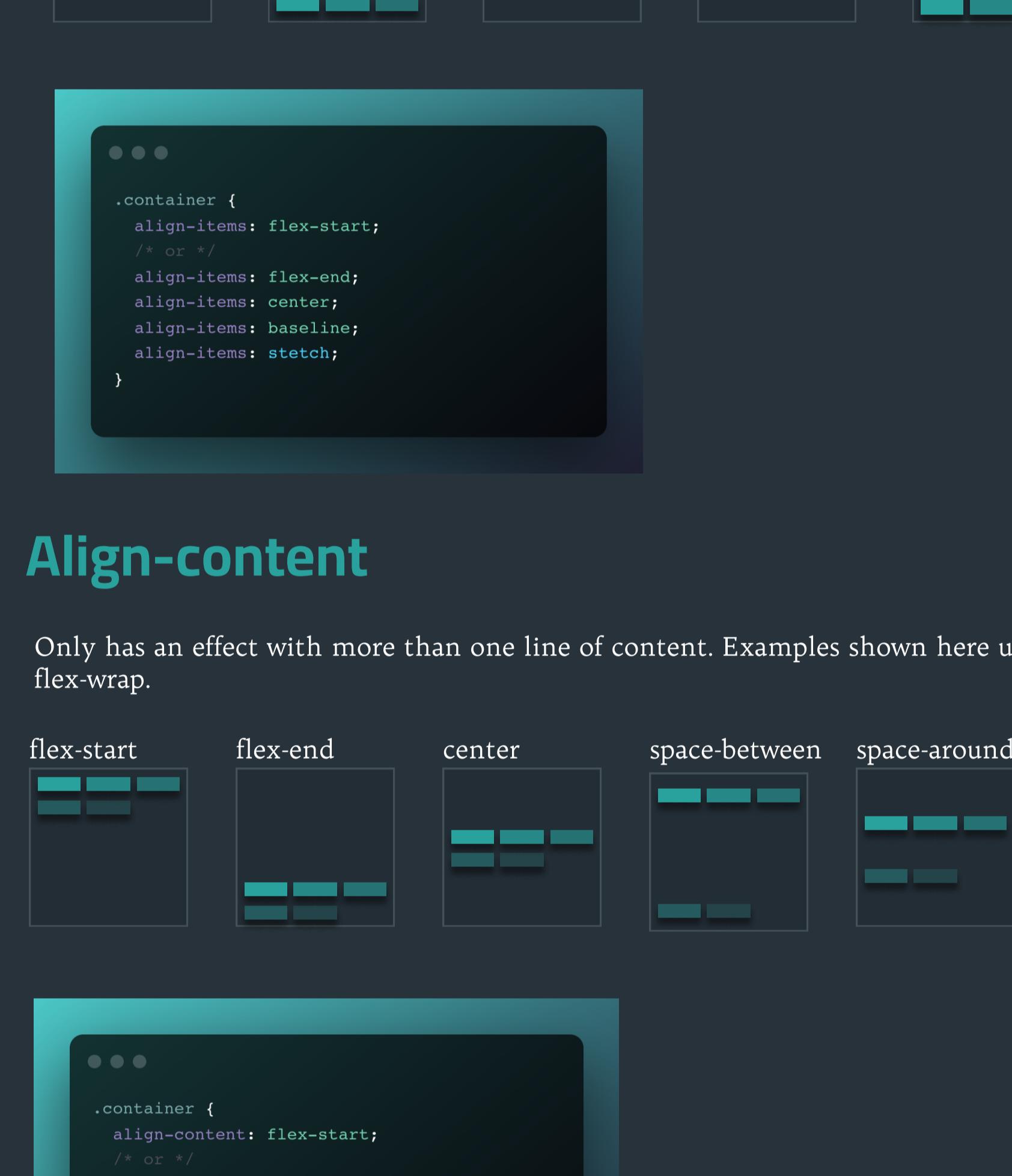
Flexbox layout is most appropriate to the components of an application, and small-scale layouts, while the Grid layout is intended for larger scale layouts. CSS-Grids are two-dimensional and contain columns and rows. So it's useful to use it for bigger layouts or the main layout as itself.

-> Addresses the Layout

Flexbox is one-dimensional and can act like a column or a row. It should be used for components inside the grid/layout or sometimes for some smaller layouts inside an already existing layout.

-> Addresses the Content.

So the best way is to combine both. But be careful. You can use the flex-box-container within a grid-construction, but not in the other direction.



## 2. Flex container (parent)

### a Display

This defines a flex container; inline or block depending on the given value. It enables a flex context for all its direct children.

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

### b Flex-direction

This establishes the main-axis, thus defining the direction flex items are placed in the flex container. Flexbox is (aside from optional wrapping) a single-direction layout concept. Think of flex items as primarily laying out either in horizontal rows or vertical columns.



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

### c Flex-wrap:

Wraps items if they can't all be made to fit on one line.

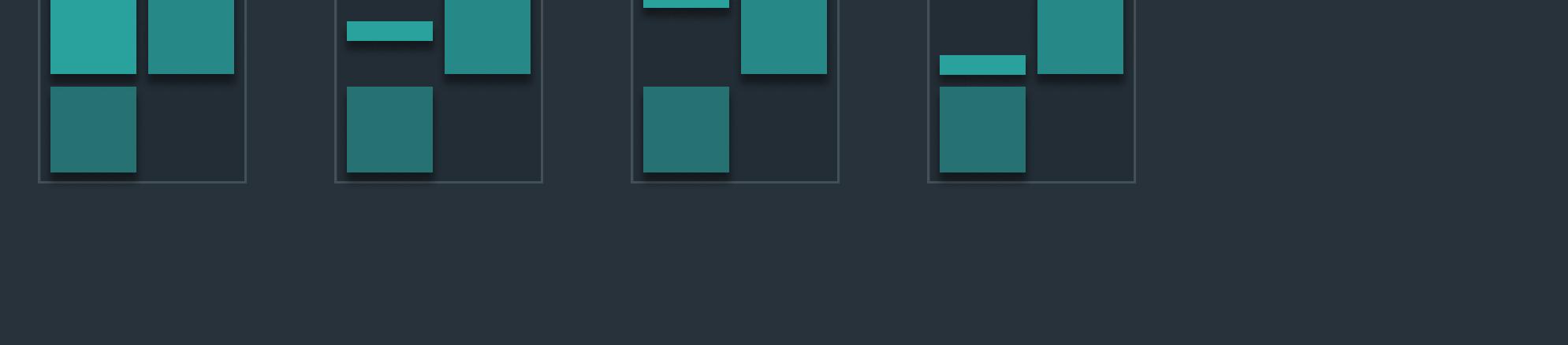


```
/* Container */
.container {
  flex-wrap: nowrap;
  /* or */
  flex-wrap: wrap;
  flex-wrap: wrap-reverse;
}
```

### d Justify-content:

Attempts to distribute extra space along the main axis

- flex-start (default): items are packed toward the start of the flex-direction.
- flex-end: items are packed toward the end of the flex-direction.
- center: items are centered along the line
- space-between: items are evenly distributed in the line; first item is on the start line, last item on the end line
- space-around: items are evenly distributed in the line with equal space around them. Note that visually the spaces aren't equal, since all the items have equal space on both sides. The first item will have one unit of space against the container edge, but two units of space between the next item because that next item has its own spacing that applies.
- space-evenly: items are distributed so that the spacing between any two items (and the space to the edges) is equal.



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

### e Align-items

This defines the default behavior for how flex items are laid out along the cross axis on the current line. Think of it as the justify-content version for the cross-axis (perpendicular to the main-axis).



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

### f Align-content

Only has an effect with more than one line of content. Examples shown here use flex-wrap.



```
/* Container */
.container {
  align-content: flex-start;
  /* or */
  align-content: flex-end;
  align-content: center;
  align-content: space-between;
  align-content: space-around;
}
```

### g Flex-flow:

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

```
/* Container */
.container {
  flex-flow: row nowrap;
  /* or */
  flex-flow: column nowrap;
}
```

```
/* Container */
.container {
  flex-flow: row;
  /* or */
  flex-flow: column;
}
```

```
/* Container */
.container {
  flex-flow: row wrap;
  /* or */
  flex-flow: column wrap;
}
```

```
/* Container */
.container {
  flex-flow: row reverse-wrap;
  /* or */
  flex-flow: column reverse-wrap;
}
```

### h Gap:

The gap property explicitly controls the space between flex items. It applies that spacing only between items not on the outer edges.

```
/* Container */
.container {
  gap: 10px 15px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   */
}
```

```
/* Container */
.container {
  gap: 10px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 10px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   * row-gap: 60px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px 65px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   * row-gap: 60px;
   * column-gap: 65px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px 65px 70px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   * row-gap: 60px;
   * column-gap: 65px;
   * row-gap: 70px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px 65px 70px 75px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   * row-gap: 60px;
   * column-gap: 65px;
   * row-gap: 70px;
   * column-gap: 75px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px 65px 70px 75px 80px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   * row-gap: 60px;
   * column-gap: 65px;
   * row-gap: 70px;
   * column-gap: 75px;
   * row-gap: 80px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px 65px 70px 75px 80px 85px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   * row-gap: 60px;
   * column-gap: 65px;
   * row-gap: 70px;
   * column-gap: 75px;
   * row-gap: 80px;
   * column-gap: 85px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px 65px 70px 75px 80px 85px 90px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   * row-gap: 60px;
   * column-gap: 65px;
   * row-gap: 70px;
   * column-gap: 75px;
   * row-gap: 80px;
   * column-gap: 85px;
   * row-gap: 90px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px 65px 70px 75px 80px 85px 90px 95px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   * row-gap: 60px;
   * column-gap: 65px;
   * row-gap: 70px;
   * column-gap: 75px;
   * row-gap: 80px;
   * column-gap: 85px;
   * row-gap: 90px;
   * column-gap: 95px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px 65px 70px 75px 80px 85px 90px 95px 100px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   * row-gap: 60px;
   * column-gap: 65px;
   * row-gap: 70px;
   * column-gap: 75px;
   * row-gap: 80px;
   * column-gap: 85px;
   * row-gap: 90px;
   * column-gap: 95px;
   * row-gap: 100px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px 65px 70px 75px 80px 85px 90px 95px 100px 105px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   * row-gap: 60px;
   * column-gap: 65px;
   * row-gap: 70px;
   * column-gap: 75px;
   * row-gap: 80px;
   * column-gap: 85px;
   * row-gap: 90px;
   * column-gap: 95px;
   * row-gap: 100px;
   * column-gap: 105px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px 65px 70px 75px 80px 85px 90px 95px 100px 105px 110px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25px;
   * row-gap: 30px;
   * column-gap: 35px;
   * row-gap: 40px;
   * column-gap: 45px;
   * row-gap: 50px;
   * column-gap: 55px;
   * row-gap: 60px;
   * column-gap: 65px;
   * row-gap: 70px;
   * column-gap: 75px;
   * row-gap: 80px;
   * column-gap: 85px;
   * row-gap: 90px;
   * column-gap: 95px;
   * row-gap: 100px;
   * column-gap: 105px;
   * row-gap: 110px;
   */
}
```

```
/* Container */
.container {
  gap: 10px 15px 20px 25px 30px 35px 40px 45px 50px 55px 60px 65px 70px 75px 80px 85px 90px 95px 100px 105px 110px 115px;
  /* Is equivalent to:
   * row-gap: 10px;
   * column-gap: 15px;
   * row-gap: 20px;
   * column-gap: 25
```