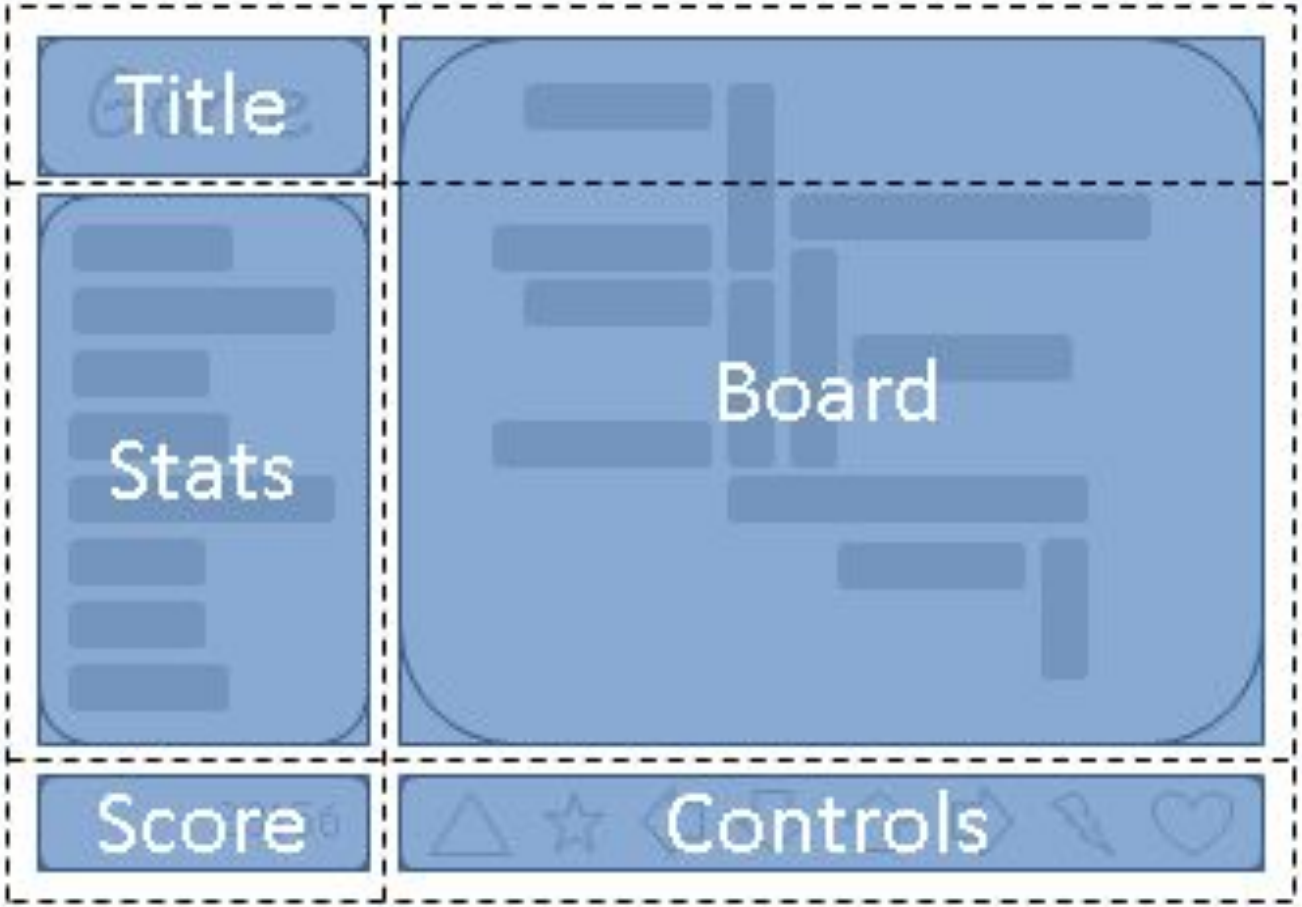
Module 3 - Lecture 3

CSS Grid & Flexbox

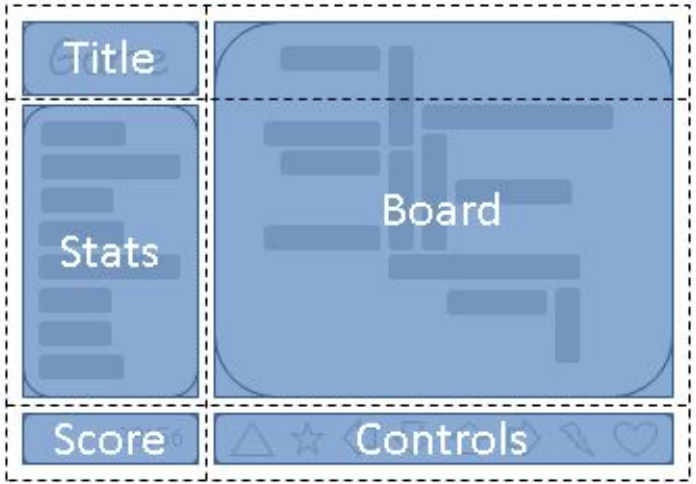
CSS Grid

https://www.w3.org/TR/css-grid-1/


Title
Stats
Board
Score
Controls

Terminology

- **Grid lines** are the vertical and horizontal dividing lines of the grid.
- A **grid item** is one of the items (e.g. Stats) and it may occupy more than one cell.
- A **grid cell** refers to one block within a grid.
- A **grid track** is a term referencing an entire column or row.
- A **grid area** is any rectangular area of one or more cells.
- The **gutters** or **gaps** are the spaces between adjacent grid tracks.



Starting a Grid Layout

```
section {
	display: grid;
}

<section>
	<div>
		<ul>
			<li>Item one</li>
			<li>Item two</li>
		</ul>
	<div>

	<p>Text goes here</p>
</section>
```

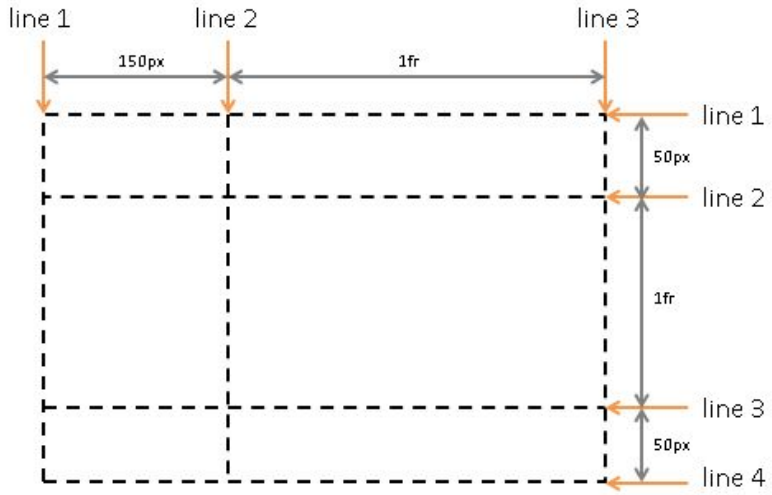
- Grid items are defined by the direct descendants of the element that is displayed as a grid.
  - In the example above, and <p> become the grid items.

- The grid will be 1 column by default include as many rows as there are grid items.
  - In the example above, this would result in a 2 row, 1 column grid.

Defining a Grid Layout

```
{
      display: grid;
      grid-template-columns: 150px 1fr;
      grid-template-rows: 50px 1fr 50px;
}
```

* The fr unit is a flexible length representing a fraction of the remaining space.

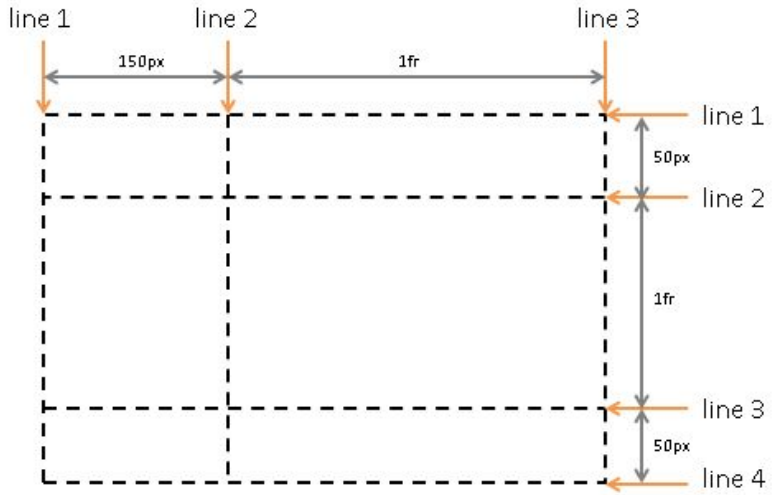


Placing Items

```
{
      display: grid;
      grid-template-areas: “. a”
                           “b .”
                           “. c”;
      grid-template-columns: 150px 1fr;
      grid-template-rows: 50px 1fr 50px;
}

#item1 { grid-area: a }
#item2 { grid-area: b }

#item3 { grid-area: c }
```

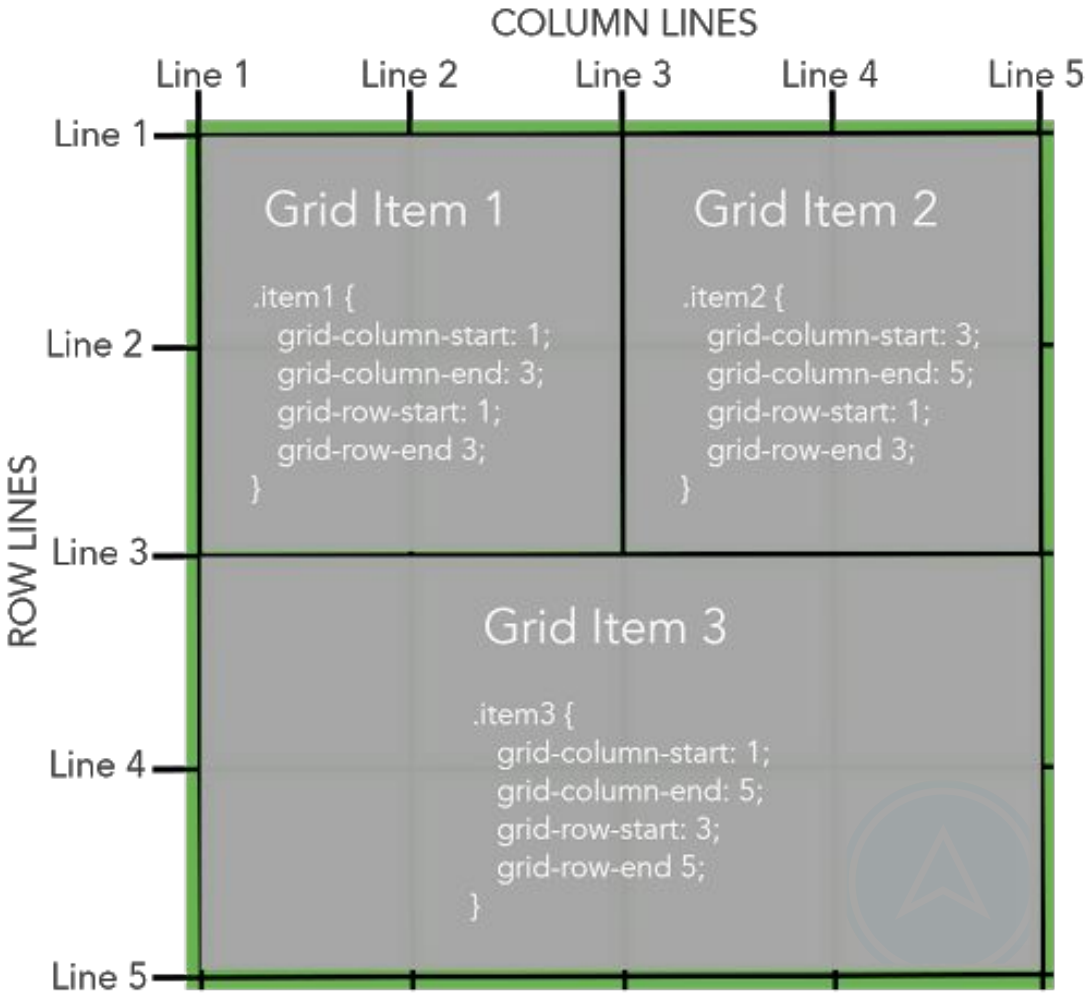


* A . is used within grid-template-areas to leave a grid cell empty.

grid-column-start,
grid-column-end,

grid-row-start,
grid-row-end

control the starting and ending location within the grid where a grid item appears.



Aligning Content

*Justify can be thought of as left to right alignment.
*Align can be thought of as vertical alignment.
*Options for alignment include start, center, end.

- **justify-items** is applied to the grid container to define justification of grid items along the row axis, within the individual grid cells

- **justify-self** is applied to any grid item to define row-axis justification within its individual grid cell

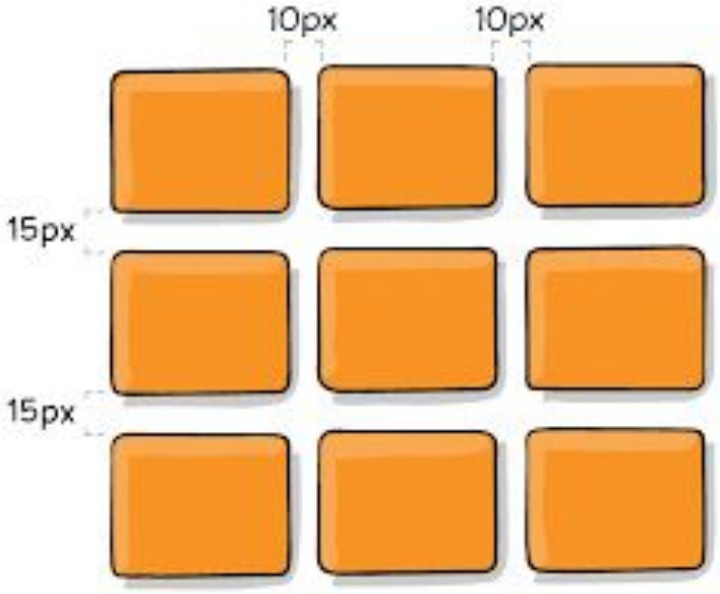
- **align-items** is applied to the grid container to define justification of grid items along the column axis, within the individual grid cells

- **align-self** is applied to any grid item to define column-axis justification within its individual grid cell

Grip Gap

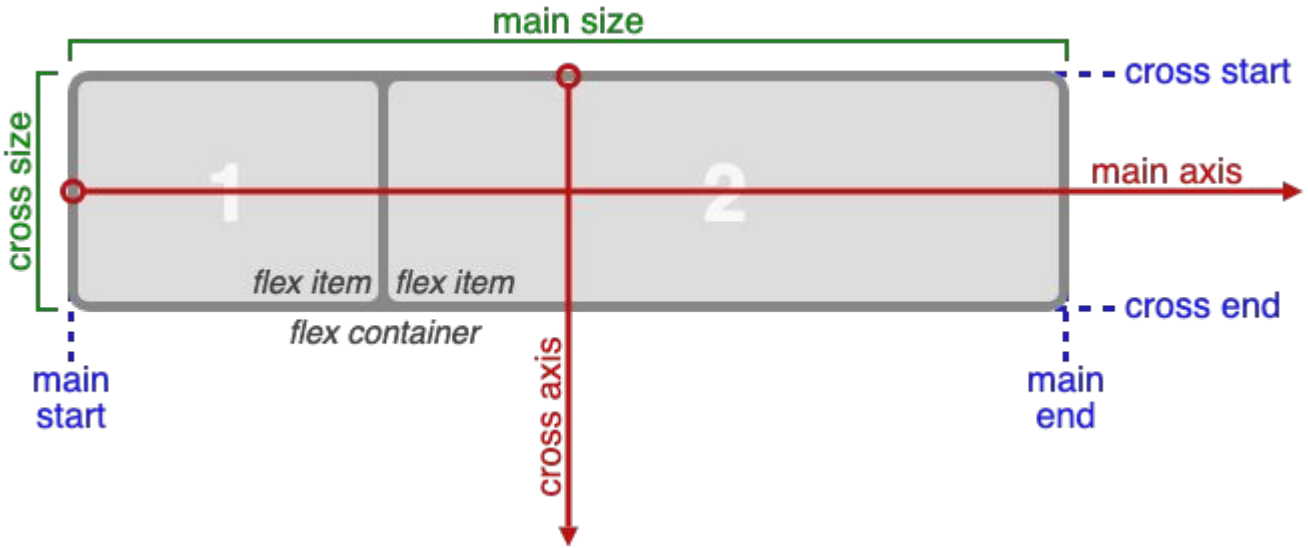
The space between the grid tracks. The gutter.

```
{
      grid-template-columns: 1fr 1fr 1fr;
      grid-column-gap: 10px
      grid-row-gap: 15px;
}
```



CSS Flexbox

https://www.w3.org/TR/css-flexbox-1/



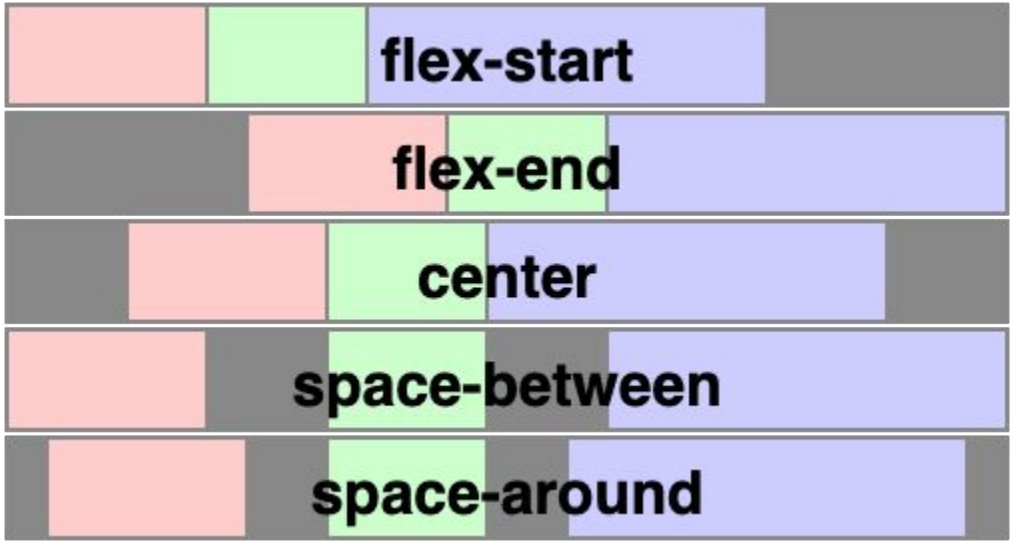
A flex container takes away some of the functionality of the block container

- For example: float and vertical alignment do not apply.

Main Axis Alignment

justify-content

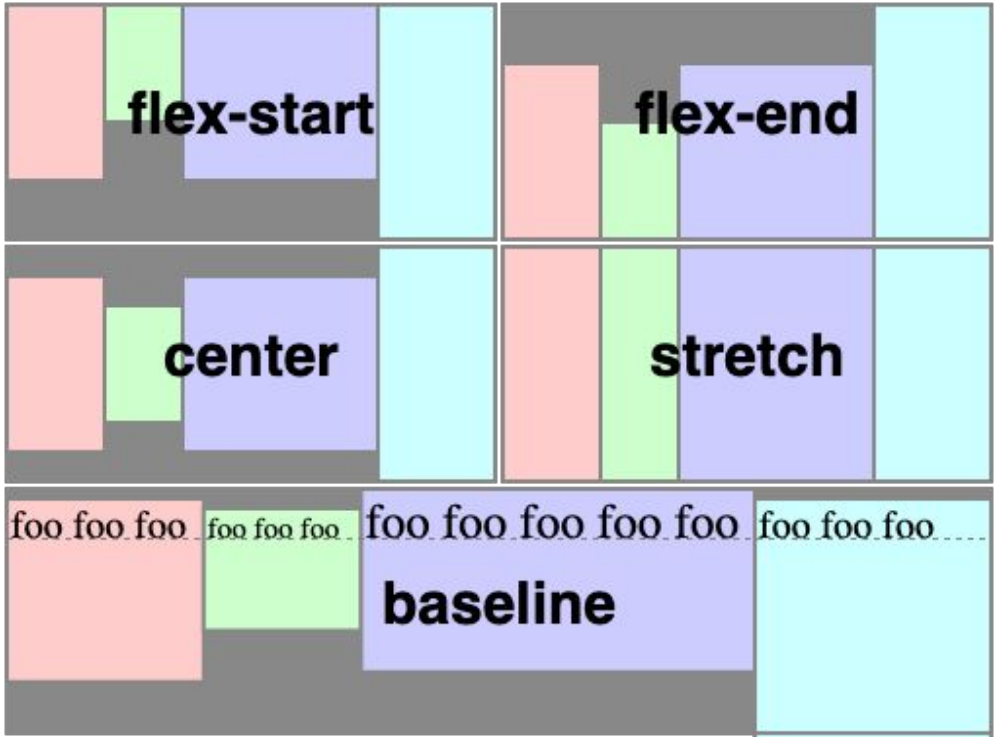
*`flex-start` is the default behavior



Cross Axis Alignment

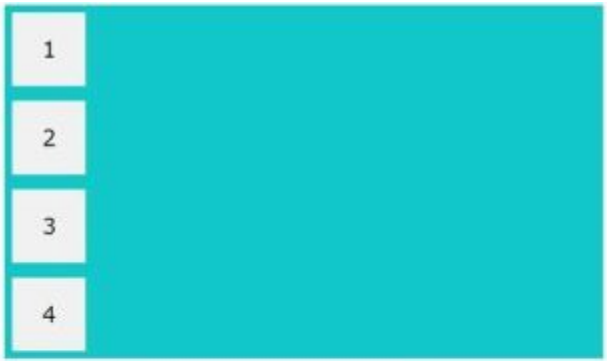
align-items, align-self

*`stretch` is the default behavior

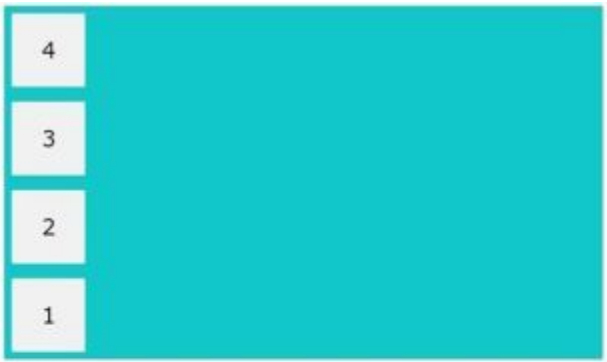


Direction of Flex Items

** Changing the direction from row to column will also change the main and cross axes



flex-direction: column



flex-direction: column-reverse

flex-direction: row

flex-direction: row-reverse

```
flex-container{
  display: flex;
  flex-direction: ____;
}
```

Customized Order of Flex Items

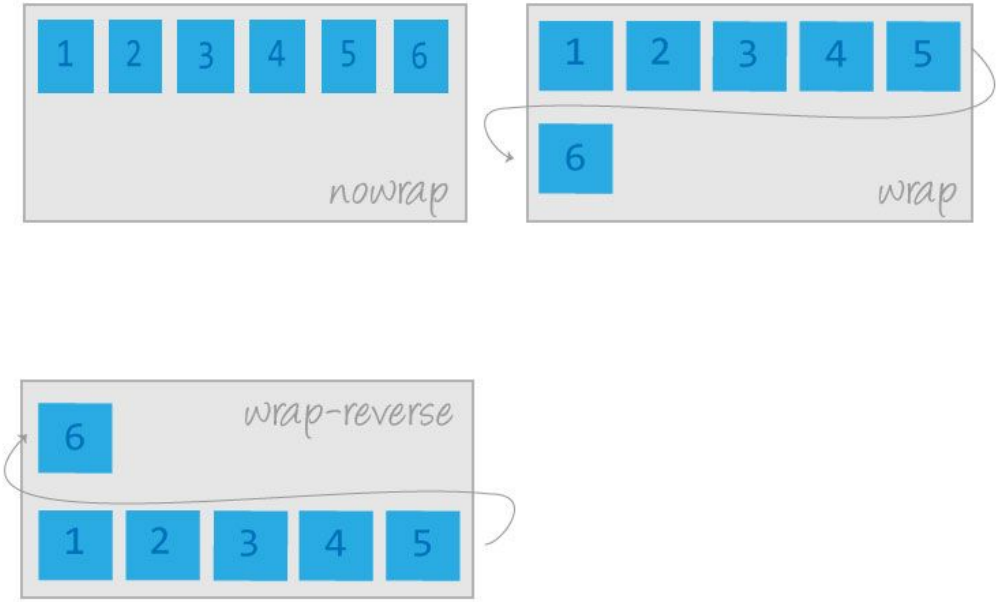
- All items have a default order of 0.
- The order property can be altered to achieve a different arrangement on the screen. It allows for positive or negative integers.

```
.items {
  display: flex;
}

.item-three {
  order: -1;
}

.item-five: {
  order: -1;
}
```

Wrapping



Flex-Wrap

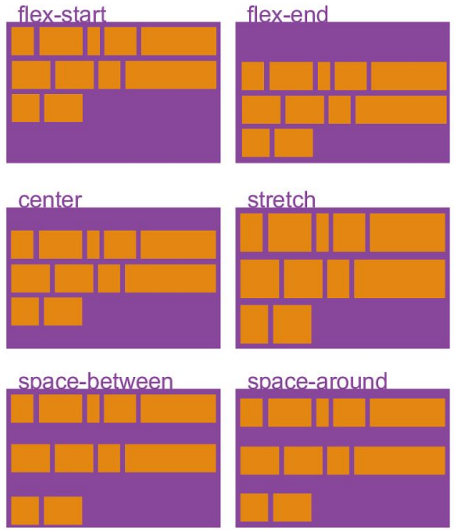
Item alignment when wrapped

align-content

Defines **line** alignment and distributes extra space along the cross axis

Has no effect when the flexbox does not wrap

Values: **flex-start**, **flex-end**, **center**, **space-between**, **space-around**, and **stretch** (default)



css-tricks.com/almanac/properties/a/align-content

Sizing Flex Items

flex-basis: The base size of the flex item.

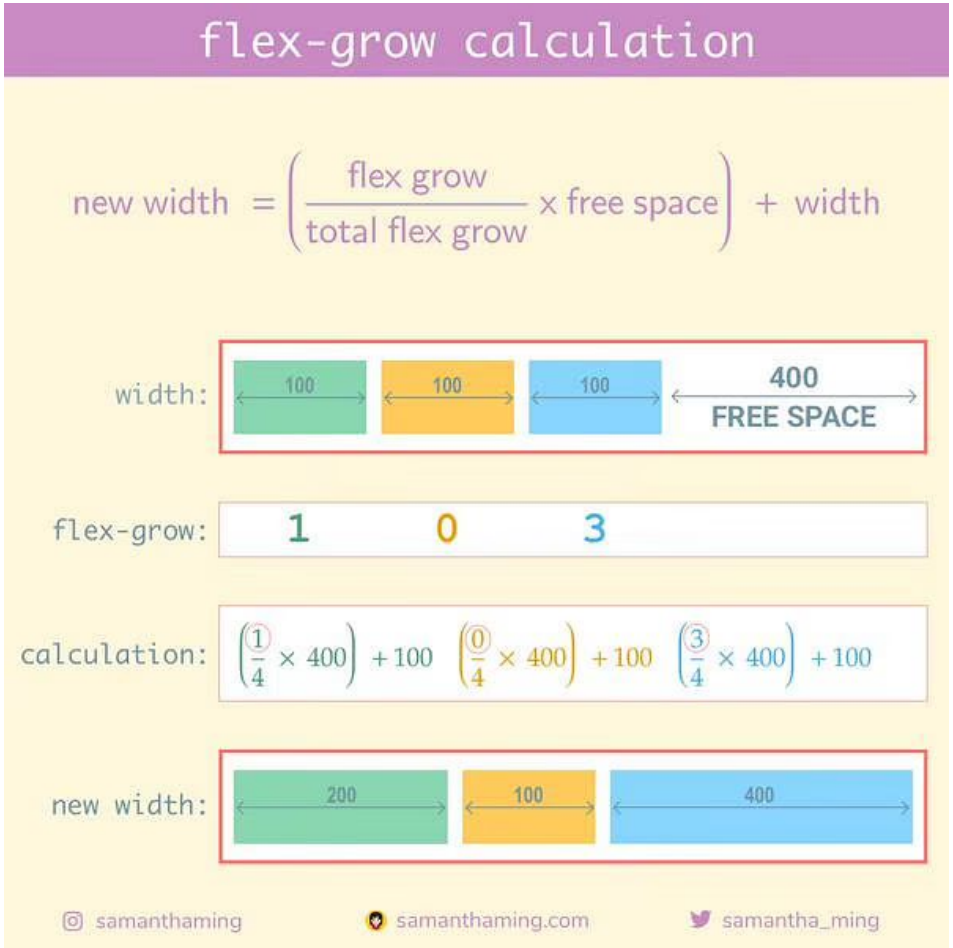
- Functionally similar to width or height, depending on the direction of the flex container, however, it will override width or height.

- Like width and height, flex-basis can be defined in absolute pixels or relative sizing using percentages or the viewport.

flex-grow: How much a flex item will grow relative to other flex items.

flex-shrink: How much a flex item will shrink relative to other flex items.

Relative sizing



Responsive Design

Media Queries

```
@media only screen and (min-width: 1024px) {
  /* Target screen sizes 1024px and above */
}


@media only screen and (max-width: 1023px) {
  /* Target screen sizes 1023px and below */
}
```

QUESTIONS?