FINAL PROJECTS

WORK ON FUNCTIONALITY

AGENDA

- Flexbox
- Lab —Work on final project

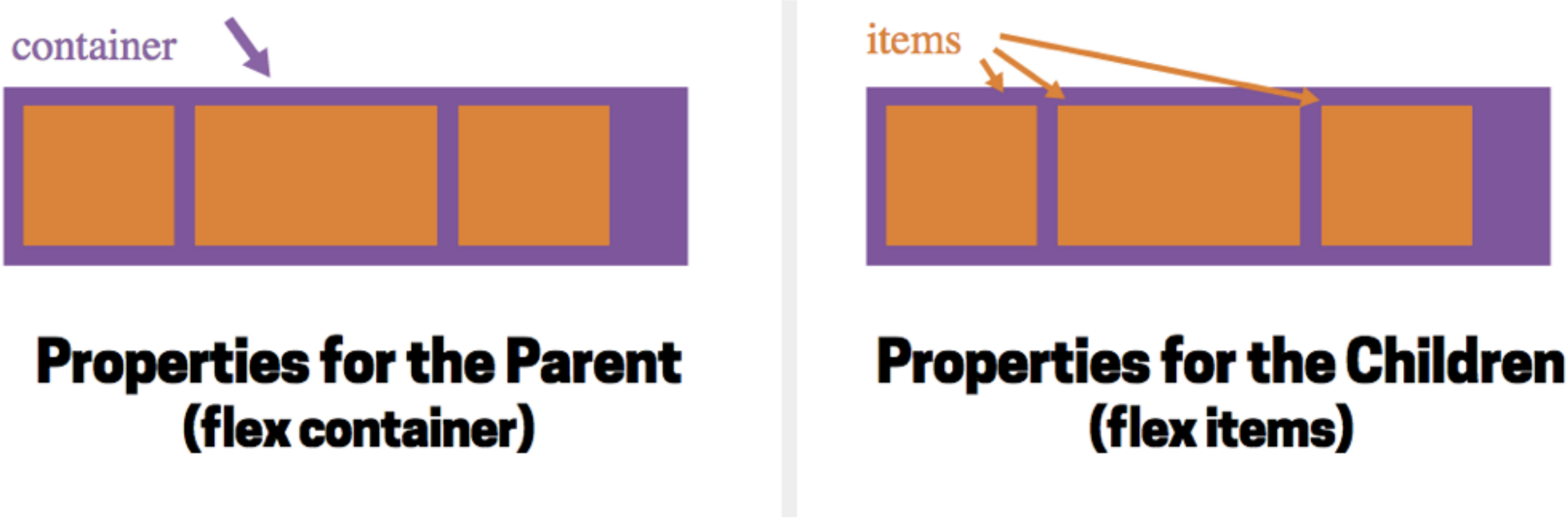
FLEXBOX

FLEXBOX

The **Flexbox Layout** (Flexible Box) aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word "flex").

FLEXBOX

- Parent- Flex Conainer
- Items- Flex Items



Properties for the Parent
(flex container)

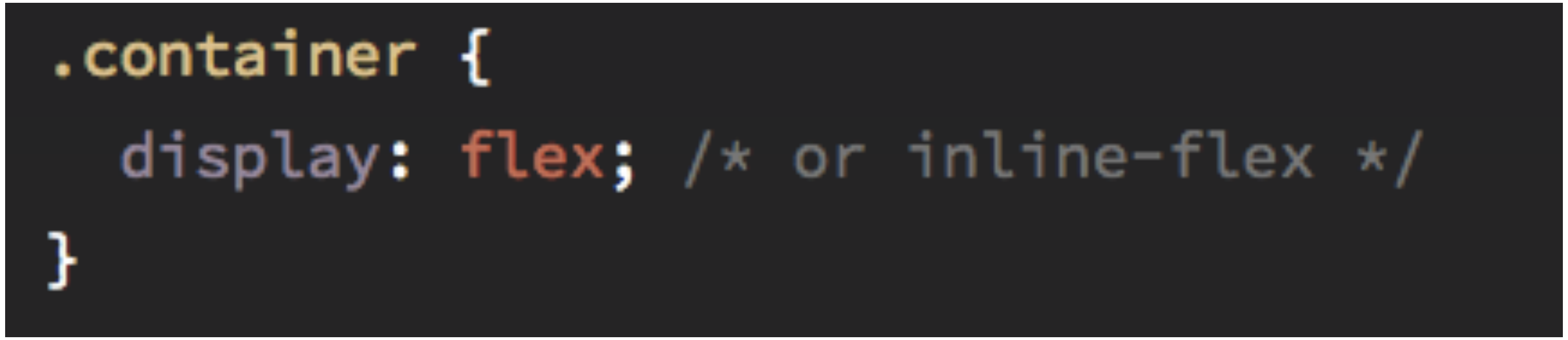
Properties for the Children
(flex items)

PARENT PROPERTIES

FLEXBOX- PARENT

Display

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

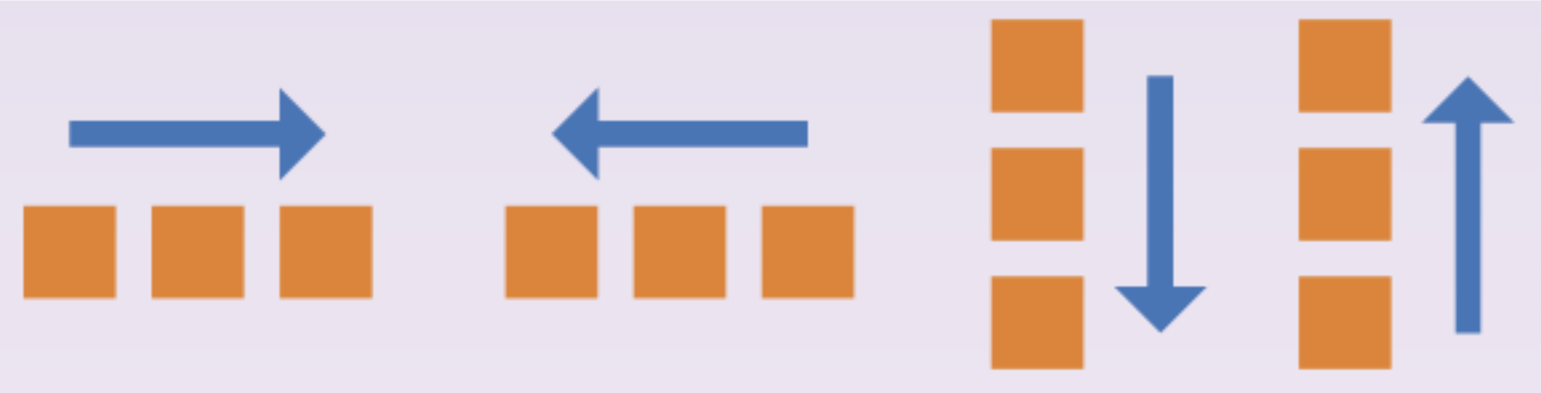


```
.container {
  display: flex; /* or inline-flex */
}
```

FLEXBOX- PARENT

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.



FLEXBOX- PARENT

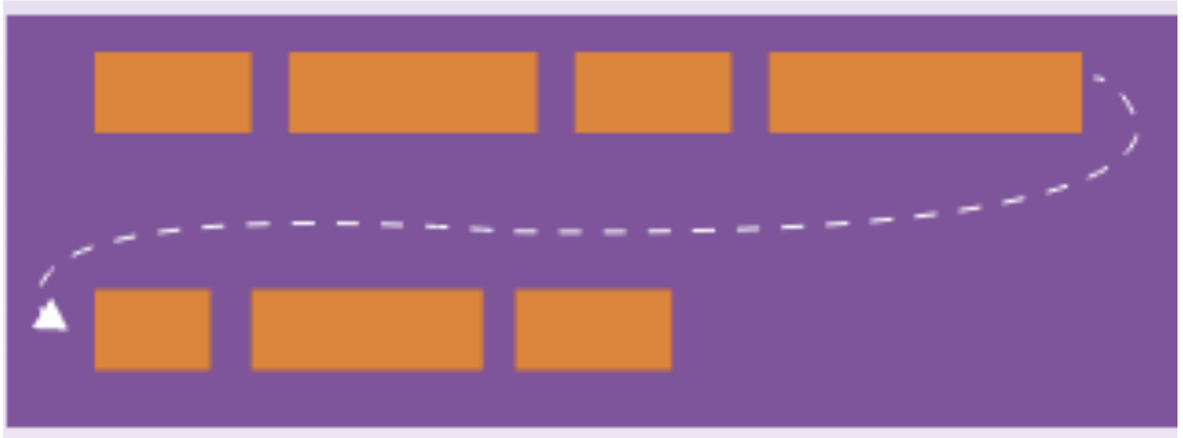
```css
.container {
  flex-direction: row | row-reverse | column | column-reverse;
}
```

row (default): left to right in ltr; right to left in rtl
row-reverse: right to left in ltr; left to right in rtl
column: same as row but top to bottom
column-reverse: same as row-reverse but bottom to top

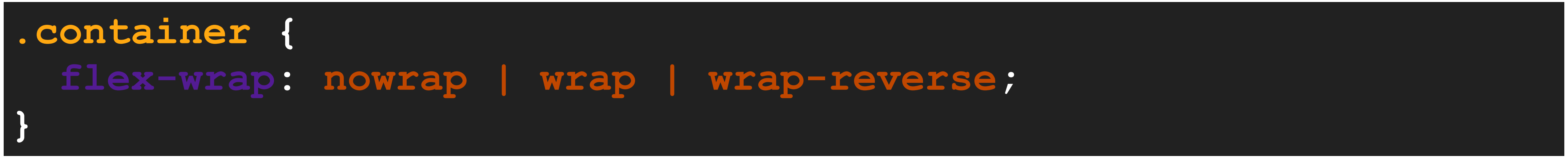
FLEXBOX- PARENT

Flex-wrap

By default, flex items will all try to fit onto one line. You can change that and allow the items to wrap as needed with this property. Direction also plays a role here, determining the direction new lines are stacked in.



FLEXBOX- PARENT



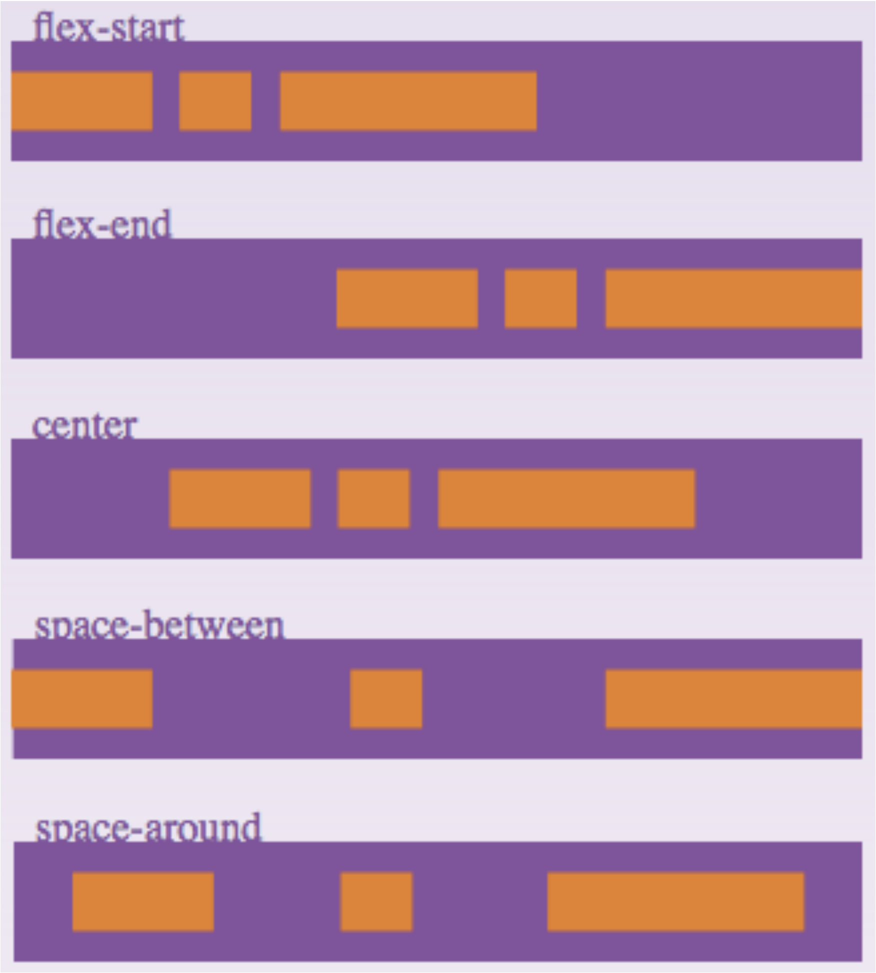
```
.container {
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

nowrap (default): single-line / left to right in ltr; right to left in rtl
wrap: multi-line / left to right in ltr; right to left in rtl
wrap-reverse: multi-line / right to left in ltr; left to right in rtl

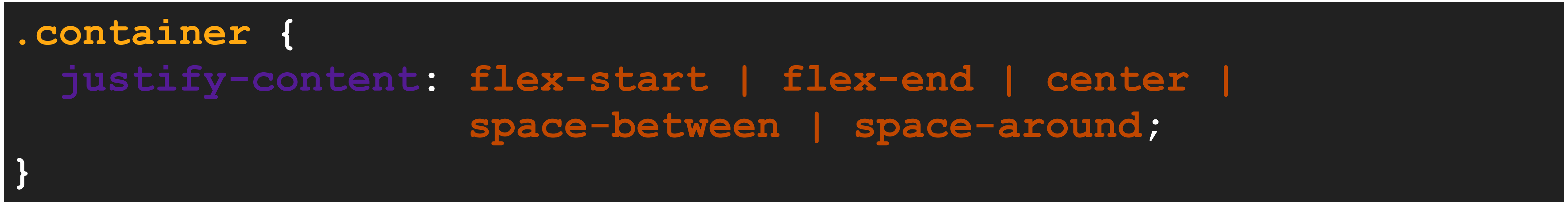
FLEXBOX- PARENT

Justify-content

This defines the alignment along the main axis. It helps distribute extra free space left over when either all the flex items on a line are inflexible, or are flexible but have reached their maximum size. It also exerts some control over the alignment of items when they overflow the line.



FLEXBOX- PARENT



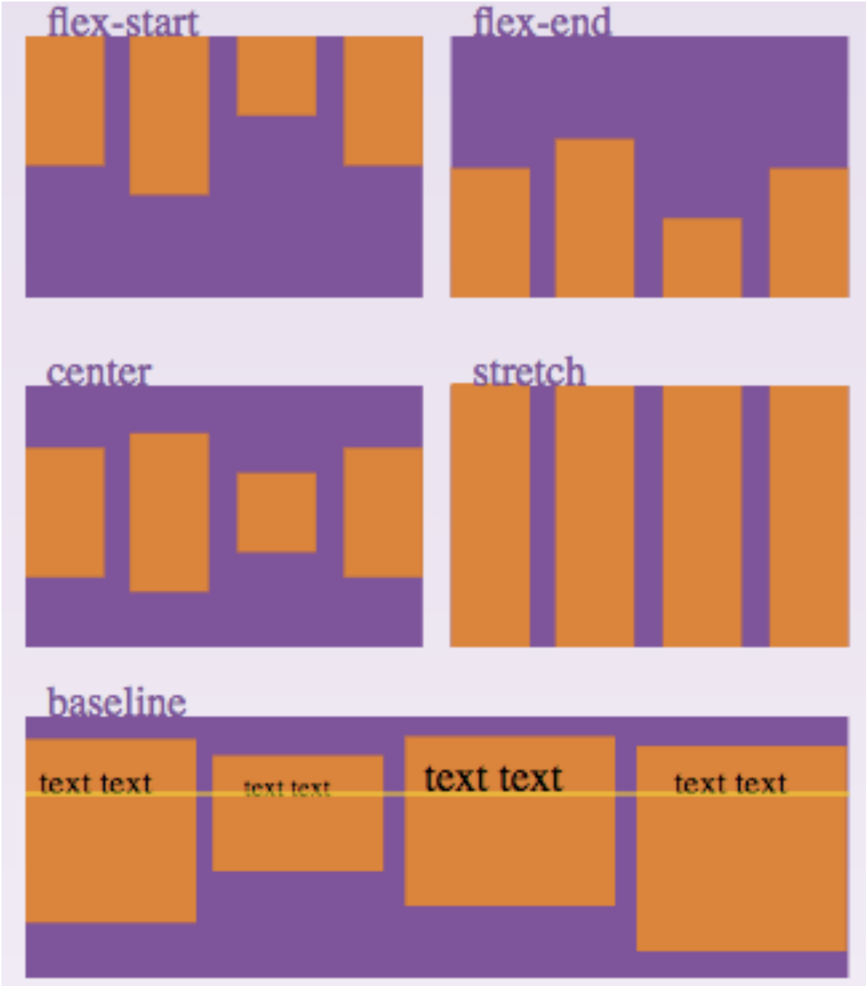
```css
.container {
  justify-content: flex-start | flex-end | center |
                   space-between | space-around;
}
```

flex-start (**default**): items are packed toward the start line
flex-end: items are packed toward to end line
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.

FLEXBOX- PARENT

Align-items

This defines the default behaviour 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).



FLEXBOX- PARENT

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

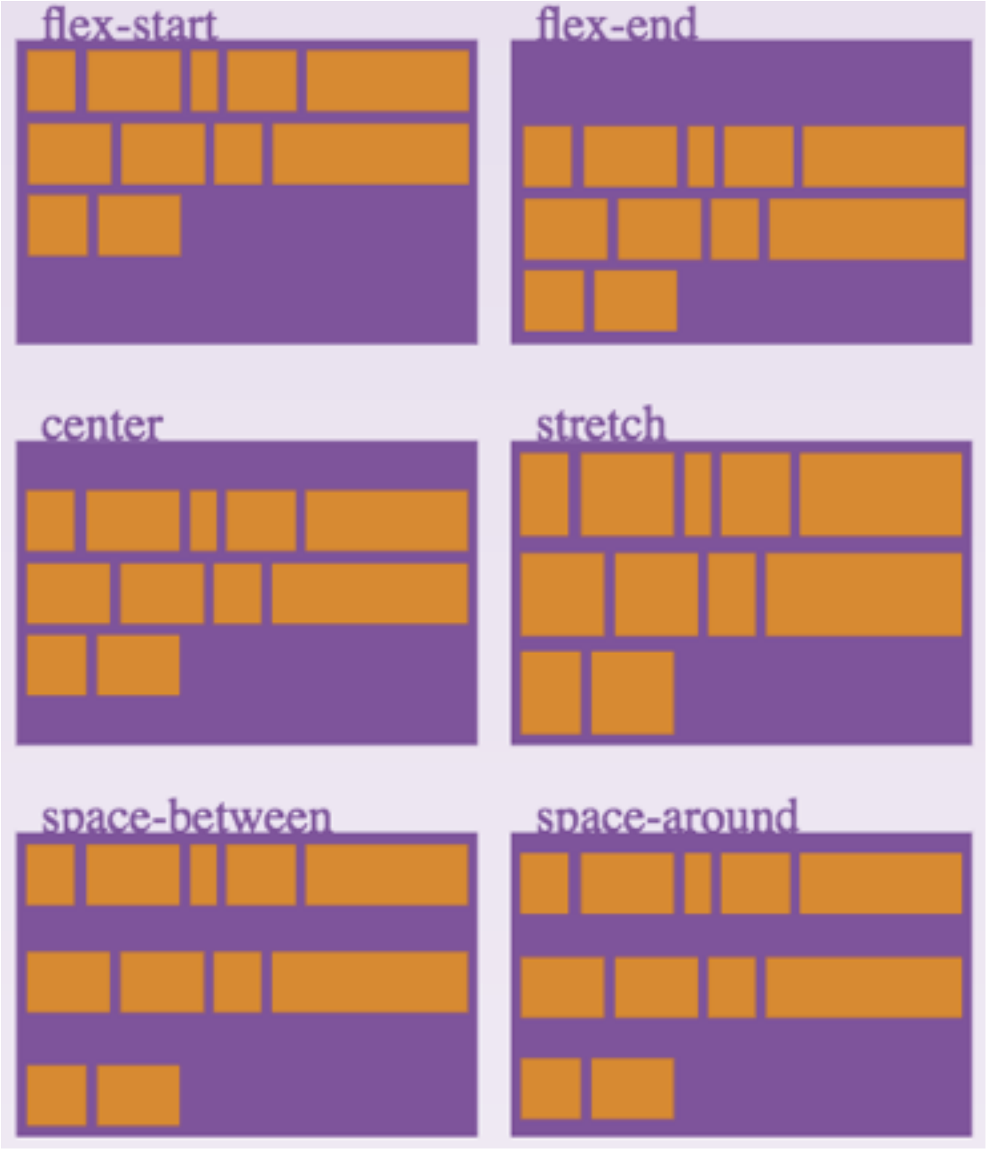
flex-start: cross-start margin edge of the items is placed on the cross-start line
flex-end: cross-end margin edge of the items is placed on the cross-end line
center: items are centered in the cross-axis
baseline: items are aligned such as their baselines align
stretch (**default**): stretch to fill the container (still respect min-width/max-width)

FLEXBOX- PARENT

Align-content

This aligns a flex container's lines within when there is extra space in the cross-axis, similar to how justify-content aligns individual items within the main-axis.

Note: this property has no effect when there is only one line of flex items.



FLEXBOX- PARENT

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

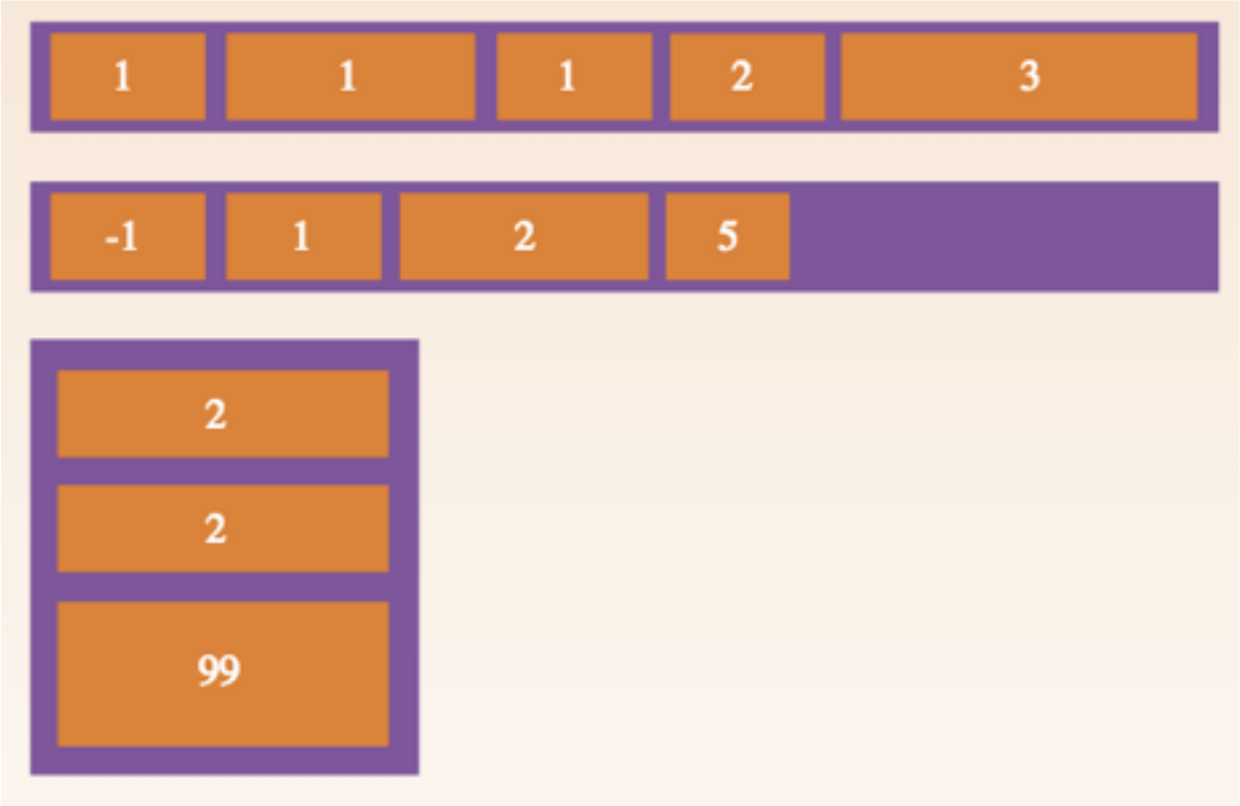
flex-start: lines packed to the start of the container
flex-end: lines packed to the end of the container
center: lines packed to the center of the container
space-between: lines evenly distributed; the first line is at the start of the container while the last one is at the end
space-around: lines evenly distributed with equal space around each line
stretch (default): lines stretch to take up the remaining space

CHILDREN PROPERTIES

FLEXBOX-CHILDREN

Order

By default, flex items are laid out in the source order. However, the order property controls the order in which they appear in the flex container.

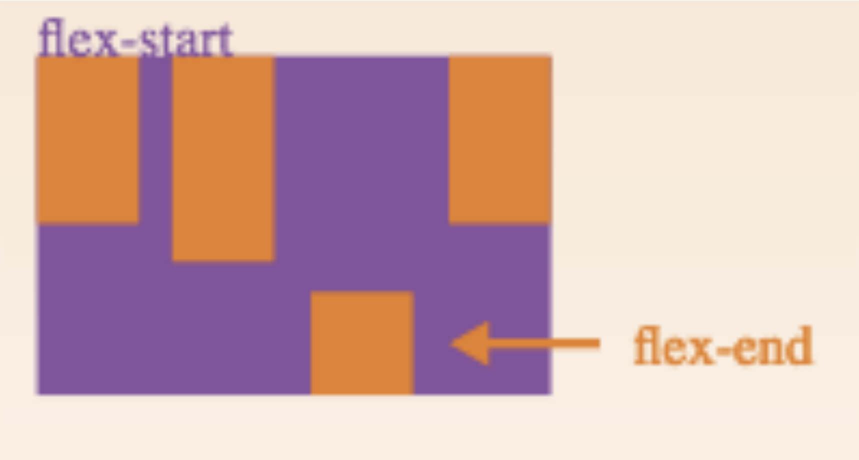


FLEXBOX-CHILDREN

```css
.child {
  order: integer;
}
```

Align-self

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



Note that float, clear and vertical-align have no effect on a flex item.

FLEXBOX-CHILDREN

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

flex-start: cross-start margin edge of the items is placed on the cross-start line
flex-end: cross-end margin edge of the items is placed on the cross-end line
center: items are centered in the cross-axis
baseline: items are aligned such as their baselines align
stretch (**default**): stretch to fill the container (still respect min-width/max-width)

HOMEWORK

FINISH FUNCTIONALITY

EXIT TICKETS