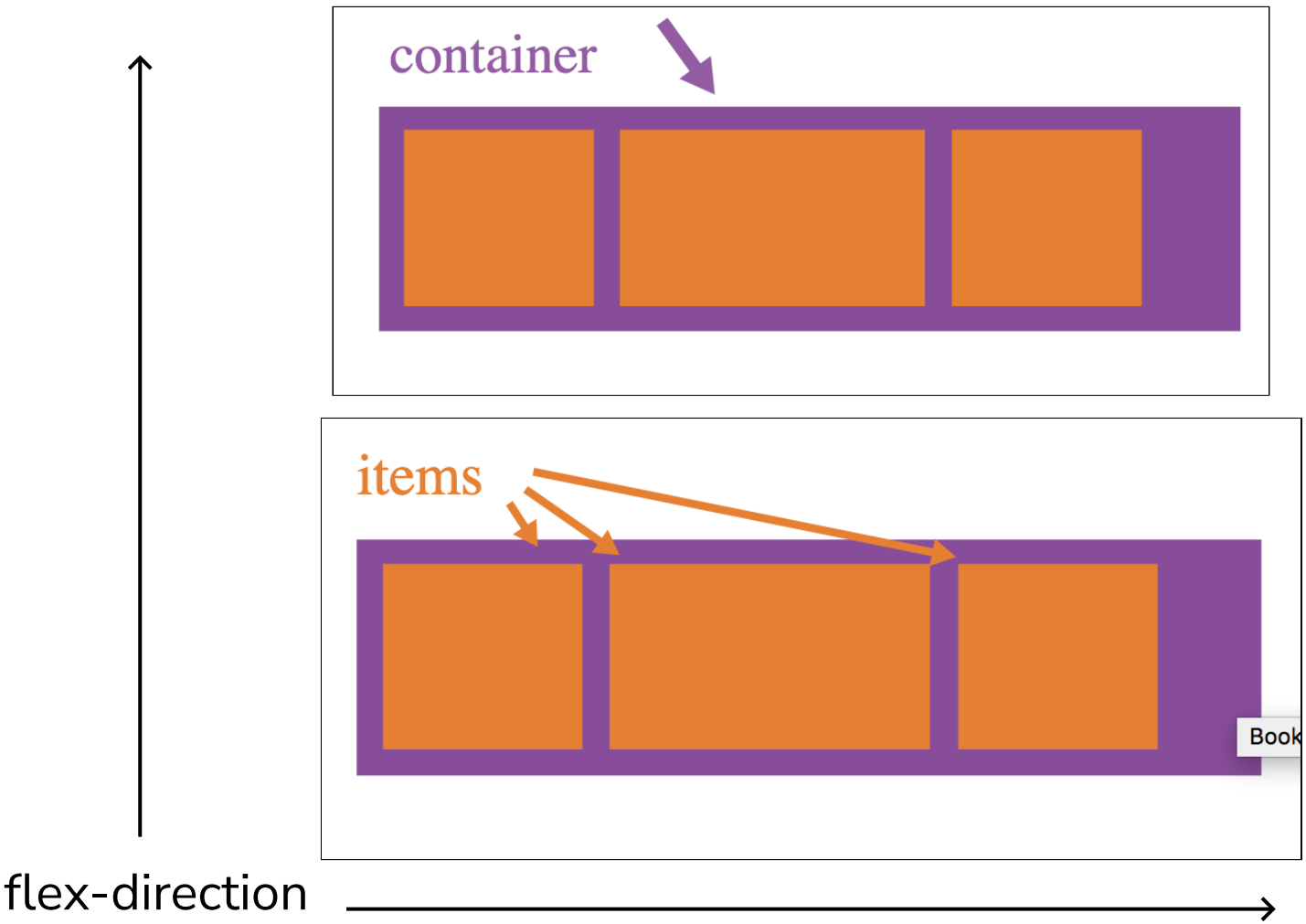
FLEXBOX

Tessa Joseph-Nicholas

COMP 126: Practical Web Design & Development for Everyone

BASIC UNITS: FLEX-CONTAINER, FLEX-ITEM, FLEX-DIRECTION



1. TURN A CONTAINER INTO A FLEX CONTAINER WITH THE DISPLAY PROPERTY

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

2. PUT CHILD ELEMENTS IN THAT CONTAINER

```
<div class="container">
    <div class="item1">item one</div>
    <div class="item1">item two</div>
    <div class="item1">item three</div>
</div>
```

NOW THE CHILD ELEMENTS ARE FLEX-ITEMS AND READY FOR FLEXBOX!

WITHOUT FLEXBOX

//codepen.io/tkjn/embed/MPOBjE/?height=265&theme-id=0&default-tab=css,result

DISPLAY: FLEX; DEFAULT BEHAVIOR

//codepen.io/tkjn/embed/LqXgOr/?height=265&theme-id=0&default-tab=css,result

DISPLAY: INLINE-FLEX MAKES THE FLEX CONTAINER BEHAVE LIKE AN INLINE ELEMENT

FLEX PROPERTIES FOR FLEX-CONTAINER (CONTAINER ELEMENT)

These properties can only be applied to a *container* element to which you've applied display: flex; or display: inline-flex;

- display (flex or inline-flex)
- flex-direction (sets main axis)
- flex-wrap (do items wrap from line to line inside container or just shrink/expand?)
- flex-flow (shorthand for flex-direction and flex-wrap)
- justify-content (aligns items along main axis)
- align-items (aligns items along cross axis)
- align-content (aligns multiple lines of items along cross axis)
- gap, row-gap, column-gap (adds gaps between flex items)

FLEX-DIRECTION: THE DIRECTION OF FLEX-ITEMS WITHIN A FLEX-CONTAINER; SETS THE MAIN AXIS

By default, X (horizontal) is the main axis and Y (vertical) is the cross axis, but you can change this

https://codepen.io/tkjn/embed/mdwaymq?default-tab=html%2Cresult

SO...

MAIN-AXIS VS. CROSS-AXIS

ESTABLISHED BY FLEX-DIRECTION VALUE

FLEX-DIRECTION: ROW; (DEFAULT) = X IS MAIN, Y IS CROSS

FLEX-DIRECTION: COLUMN; = Y IS MAIN, X IS CROSS

FLEX-WRAP: WHETHER FLEX-ITEMS WRAP FROM LINE TO LINE IN A FLEX-CONTAINER

https://codepen.io/tkjn/embed/jOwXEej?default-tab=html%2Cresult

FLEX-FLOW: DIRECTION/WRAP SHORTHAND SYNTAX

```
flex-flow: row nowrap; (default)

flex-flow: column wrap;

flex-flow: column-reverse wrap-
reverse;

etc.
```

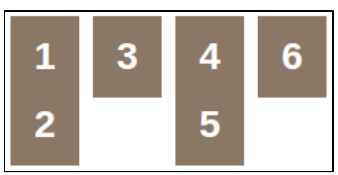
JUSTIFY-CONTENT: ALIGNS FLEX-ITEMS ALONG THE MAIN AXIS OF THE FLEX-CONTAINER

https://codepen.io/tkjn/embed/zYzyGrZ?default-tab=html%2Cresult

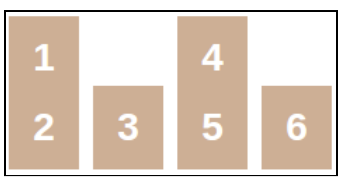
Used primarily when the flex-items' sizes are fixed or at their max-height/max-width

ALIGN-ITEMS: ALIGNS FLEX-ITEMS ALONG THE CROSS AXIS

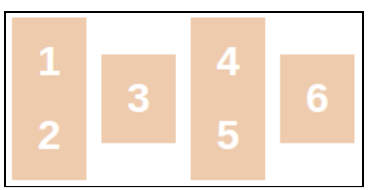
https://codepen.io/tkjn/embed/BaZvNze?default-tab=html%2Cresult



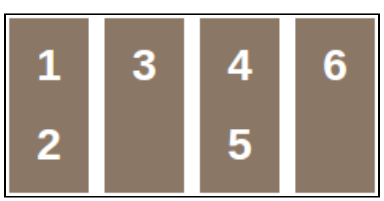
align-items: flex-start;



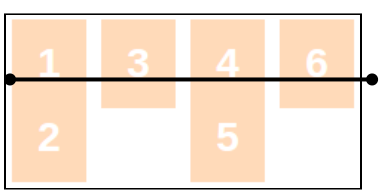
align-items: flex-end;



align-items: center;



align-items: stretch;



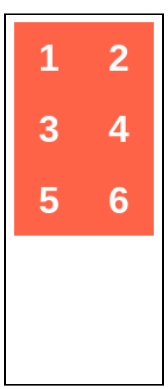
align-items: baseline;

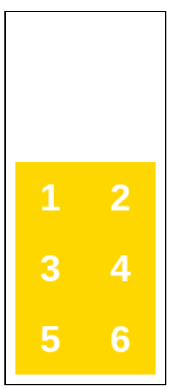
ALIGN-CONTENT: ALIGNS MULTIPLE LINES OF FLEX-ITEMS IN THE CROSS-AXIS

https://codepen.io/tkjn/embed/eYRbNWv?default-tab=html%2Cresult

align-content:

flex-start;



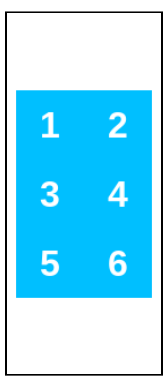


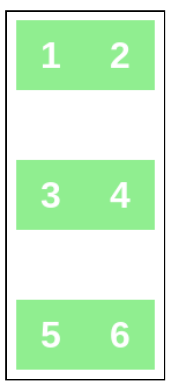
align-content:

flex-end;

align-content:

center;



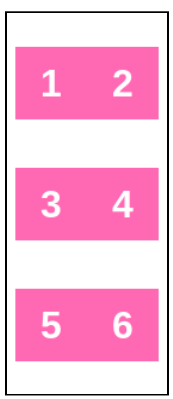


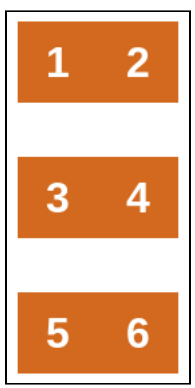
align-content:

space-between;

align-content:

space-around;





align-content:

stretch;

FLEX-ITEM PROPERTIES

NOTE: these properties can only be applied to individual *items* within a flex-container

- **order:** affects the order in which single items appear in a flex container
- **align-self:** overrides the default or assigned (with align-items) alignment for an individual item in a flex container
- **flex-grow:** how much should a flex item grow/what proportion of the available space should it take up, if space is available?
- **flex-shrink:** how much/by what proportion can a flex item shrink if necessary?
- **flex-basis:** default size of a flex item *before* any growing or shrinking or redistribution of available space in a flex container

ORDER

//codepen.io/tkjn/embed/BqmOWJ/?height=265&theme-id=0&default-tab=css,result

ALIGN-SELF: OVERRIDES DEFAULT OR ALIGN-ITEMS ALIGNMENT FOR INDIVIDUAL FLEX-ITEMS

FLEX-GROW: HOW MUCH FLEX ITEMS EXPAND WHEN THERE'S ROOM

FLEX-SHRINK: HOW MUCH FLEX ITEMS SHRINK IF CONSTRAINED

//codepen.io/tkjn/embed/QZaOrp/?height=265&theme-id=0&default-tab=css,result

FLEX-BASIS: DEFAULT SIZE OF FLEX-ITEMS BEFORE SPACE IS DISTRIBUTED

https://codepen.io/tkjn/embed/RwgEvYv?default-tab=html%2Cresult

FLEX: SHORTHAND FOR GROW, SHRINK, BASIS

https://codepen.io/tkjn/embed/JjJwVKo?default-tab=html%2Cresult

IN SUMMARY

- Use display: flex; to turn a container into a flex-container and its items into flex-items.
- Use justify-content to set the main-axis alignment of flex-items.
- Use align-items to define the cross-axis alignment of flex-items.
- Use flex-direction to switch between columns and rows of flex-items.
- Use the row-reverse or column-reverse values to flip the flex-items' order.
- Use order to customize the order of individual elements.
- Use align-self to vertically align individual flex-items.
- Use flex-grow, -shrink, and -basis (or the "flex" property, which is shorthand for those three) to create flexible boxes that can stretch and shrink.

STUFF YOU CAN DO WITH FLEXBOX

CENTER ITEMS IN A CONTAINER

//codepen.io/tkjn/embed/rRNjJP/?height=265&theme-id=0&default-tab=css,result

DISTRIBUTE ITEMS EQUALLY

//codepen.io/tkjn/embed/exQPQG/?height=265&theme-id=0&default-tab=css,result

RESPONSIVE NAVIGATION

//codepen.io/tkjn/embed/PVxyXo/?height=265&theme-id=0&default-tab=css,result

RESPONSIVE LAYOUT

//codepen.io/tkjn/embed/WPYaLx/?height=265&theme-id=0&default-tab=css,result

FLEXBOX WEBPAGE DEMO

https://codepen.io/tkjn/embed/xxxbdZj?height=265&theme-id=0&default-tab=css,result

FLEXBOX FROGGY

USES OF FLEX:

https://codepen.io/tkjn/embed/OJJypOv?height=265&theme-id=0&default-tab=css,result