# Flexbox and Responsive Design

# Flexbox layouts

- Flexbox makes it easier to have responsive design without using float or positioning (or media-queries).
- Gives container the ability to alter its items' width/height (and order) to best fill the available space
  - Expands items to fill available free space
  - Shrinks them to prevent overflow.
- Is direction-agnostic
  - (block which is vertically-based and inline which is horizontally-based).

# Container Properties

- flex-direction – how to stack items.
- flex-wrap - whether items should wrap or not
- flex-flow – shorthand for direction and wrap
- justify-content –horizontal alignment
- align-items – vertical alignment
- align-content – where the flex lines should go

# Values for the Parent properties

```
container{
  display: flex | inline-flex;
  flex-direction: row | row-reverse | column | column-reverse;
  flex-wrap: nowrap | wrap | wrap-reverse;
  flex-flow: <flex-direction> || <flex-wrap>;
  justify-content: flex-start | flex-end | center | space-between|
     space-around | space-evenly; }
  align-items: flex-start | flex-end | center | baseline | stretch;
  align-content: flex-start | flex-end | center | space-between |
      space-around | stretch;
}
```

# Item Properties

- order -can change the order of the items
- flex-grow - how much the item will grow relative to the rest of the items.
- flex-shrink - similar
- flex-basis - initial length of the item
- flex – shorthand for grow, shrink, and basis
- align-self - specifies the alignment (overrides parents align-items value).

# Values for the Child Properties

```
.item{
  order: <integer>;
  flex-grow: <integer>;
  flex-shrink: <integer>;
  flex-shrink: <length> | auto;
  flex : none | [ <'flex-grow'> | <'flex-shrink'> |
     <'flex-basis’> ]
  align-self: auto | flex-start | flex-end | center |
     baseline | stretch;
}
```

# Example

- Code Pen Flex Box example
- Flex Example #2

- Let's take some time and change each of the property values.

- Flex is great, but is still one-dimensional
- Best used within containers and small applications

# When to use Grid/Flex/Bootstrap

- When would you use each?

- How do they relate to responsive design?

- What should you do for your portfolio?

# Acknowledgements

- Initial Development: Colleen van Lent , University of Michigan School of Information