

# **Client-Side Web Development**

**Class 2.1**

# Today's Topics

- Flexbox
- **Exercise:** Flexbox Froggy

# **Announcements**

# Recordings

# **In-Class Labs**

# Late Policy

# PA Schedule

**Any Questions?**



# Flexbox

**Display**

The `display` property sets the display type of an element.

**Display types can be put into two categories:**  
***Outer Display Types* and *Inner Display Types*.**

***Outer Display Types*** affect the elements itself  
and how it is displayed in the flow layout

# Outer Display Types

- inline
- block
- inline-block

***Inner Display Types*** affect how the element's  
children are displayed

# Inner Display Types

- Flex
- Grid



# Flexbox

***Flexbox*** is one-dimension layout model that allows control over space distributed and item alignment

Use the `display: flex` to create a **Flexbox** container.

**A *Flexbox container* works with two axes, the main and the cross axis**

**The `flex-direction` property defines which  
is the main axis**

**A *flexbox container* can control the distribution and alignment of the children**

The **justify-content** property is used to set how the children are distributed along the main axis

The `align-items` property defines how children are laid along the cross axis



**By default, children of a *flexbox container* will remain on a single line.**

The `flex-wrap` property is used to allow children to wrap.

# Examples

**By default, children of a flexbox container are laid out in the source order.**

**The `order` property controls the order in which children appear in the parent**

The **order** property takes an integer as a value, negative numbers are allowed, with the default being **0**

**Elements will be displayed in numerical order based on their `order` value. If elements have the same value source order will be used**

The `flex-grow` property gives the ability for the child to "grow" to fill the available space.



The `flex-grow` property takes an integer as a value with the default being `0`

**Elements with a higher `flex-grow`, will grow more. Elements with `2` will grow twice as much as the elements with `1`.**

**If one of the children has a `flex-grow` value of `2`, the child would take up twice as much space as the others**

The `align-self` property allows a child to override the parent `align-items` property

# Examples

# **Exercise: Flexbox Froggy**

# For next class...

- **Lab:** Increase Your Flexibility