HTML & CSS: Day 4

Flex! You're Going To Love This
... and, The Amazing Pseudo-Selectors

Lesson Objectives

FLEXBOX

1. Know the difference: flex layout versus the box model
2. Parent/Container properties
3. Item/Children properties

PSEUDO SELECTORS

1. Select elements based on some special states of elements
2. User actions (:hover, :focus, etc)
3. Position in the HTML (nth-child, etc)
4. Relation to other selectors (:not())

 | All content is proprietary and confidential.

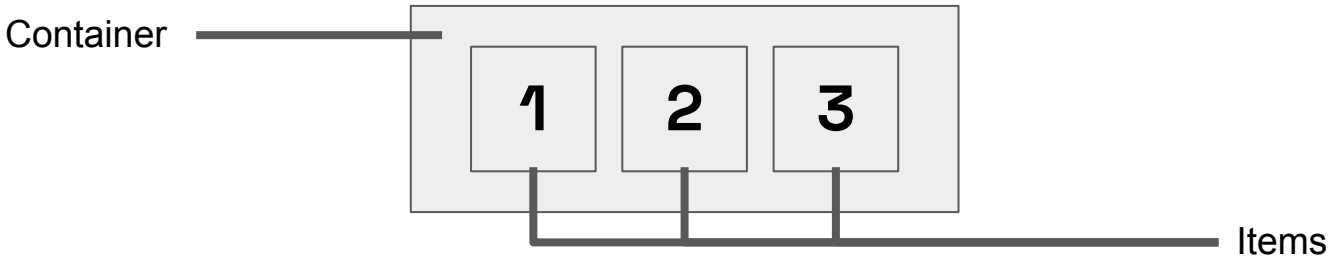
Flexbox:

1. Puts a container into a different layout “mode”

 | All content is proprietary and confidential.

Flexbox:

1. Puts a container into a different layout “mode”
2. Comprised of a container (or parent) and items (or children)



Resources:

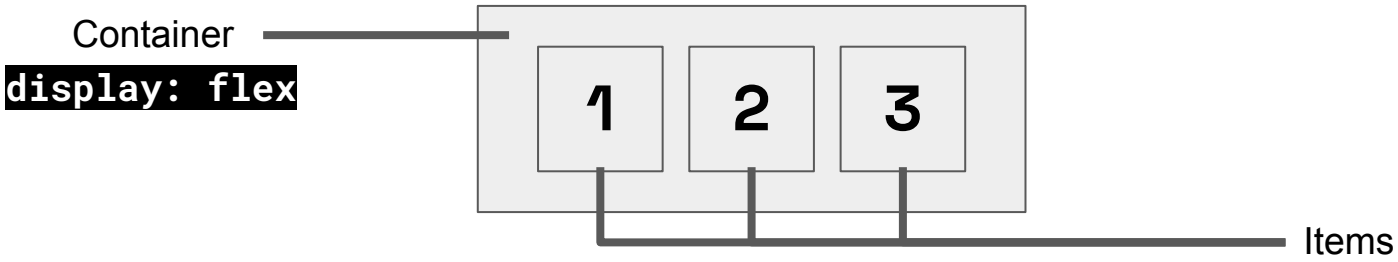
https://css-tricks.com/snippets/css/a-guide-to-flexbox/

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox

 All content is proprietary and confidential.

Flexbox:

1. Puts a container into a different layout “mode”
2. Comprised of a container (or parent) and items (or children)
3. `display: flex` must be set on the container

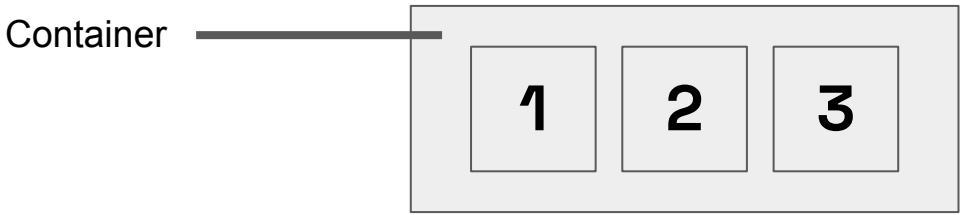


 | All content is proprietary and confidential.

Flexbox **container properties**

Properties for flex **containers** determine how the flex items inside that container are laid out.

Flex containers can automatically make their items grow, shrink, or move around to fit inside them.

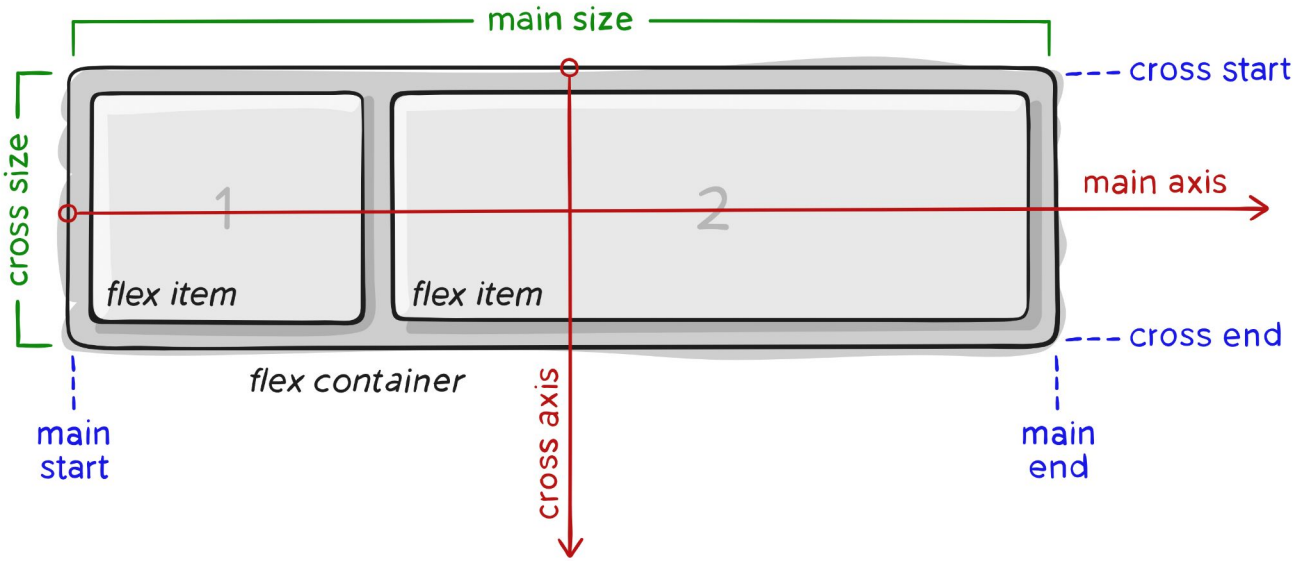


 | All content is proprietary and confidential.

Image credit: css-tricks

Flexbox: **main** axis and **cross** axis

Flexbox lays content out along two axes.
Content flows along the **main** axis, and may expand along the **cross** axis, which is always perpendicular to the main axis.



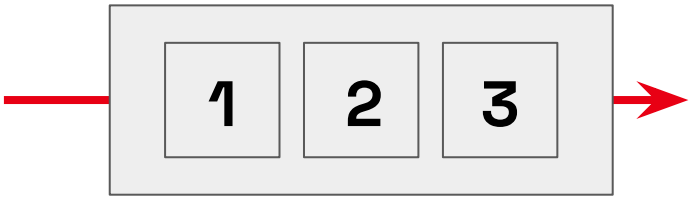
 | All content is proprietary and confidential.

Image credit: css-tricks

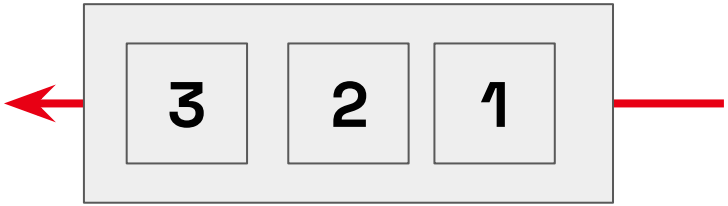
Flexbox container property: `flex-direction`

Controls the direction of the main axis, and the direction of item flow along the main axis

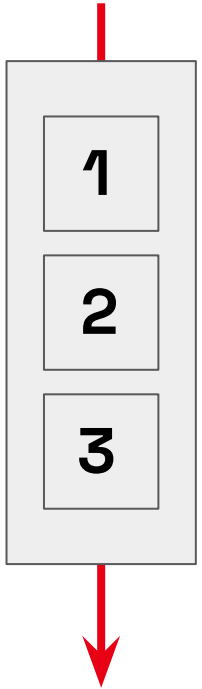
`flex-direction: row`



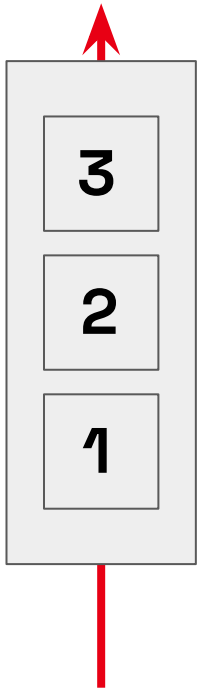
`flex-direction: row-reverse`



`flex-direction: column`



`flex-direction: column-reverse`

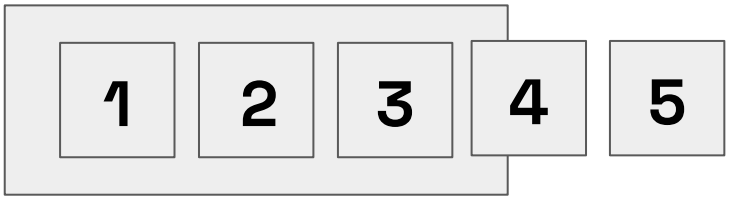


| All content is proprietary and confidential.

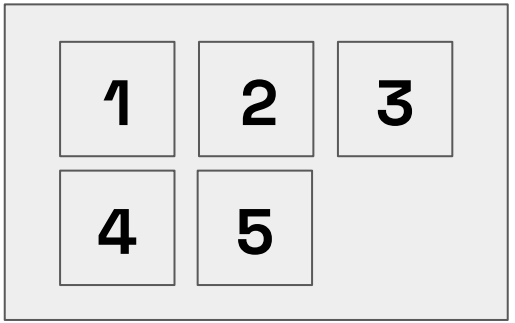
Flexbox container property: `flex-wrap`

Controls how items wrap to multiple lines (or not)

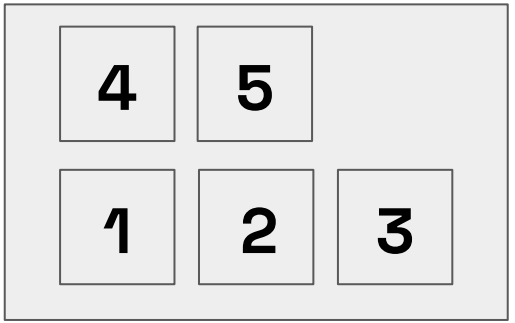
`flex-wrap: nowrap`



`flex-wrap: wrap`



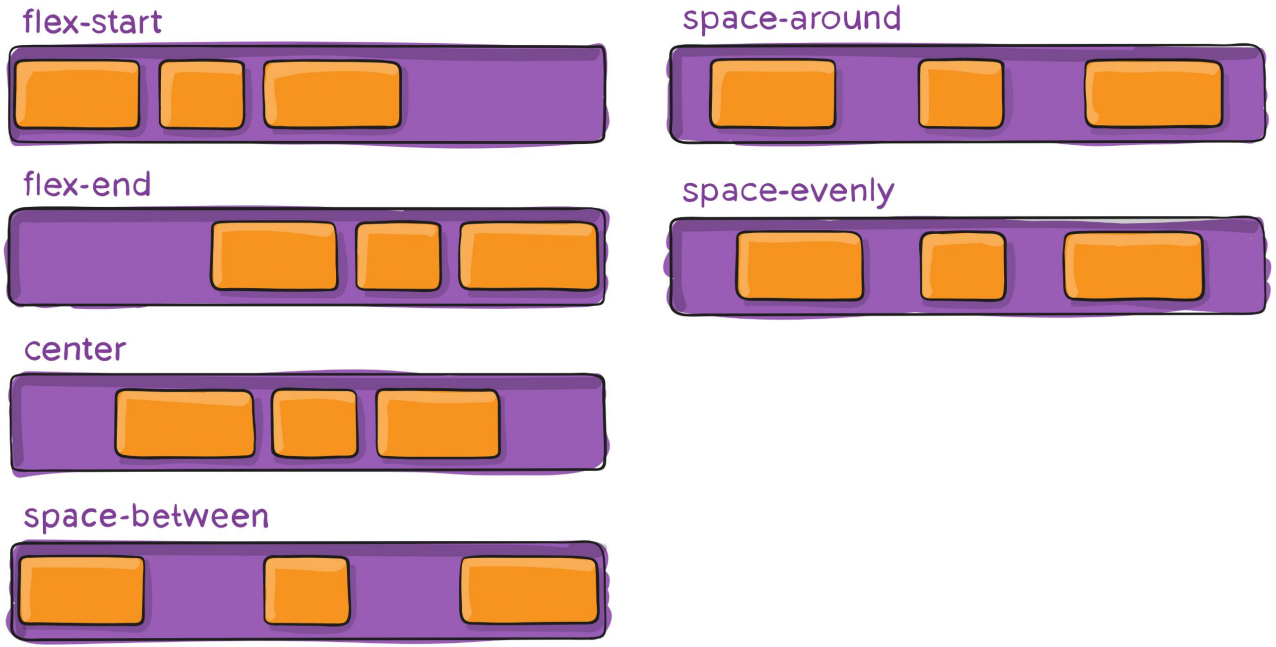
`flex-wrap: wrap-reverse`



 | All content is proprietary and confidential.

Flexbox container property: `justify-content`

Controls how items align along the **main axis**

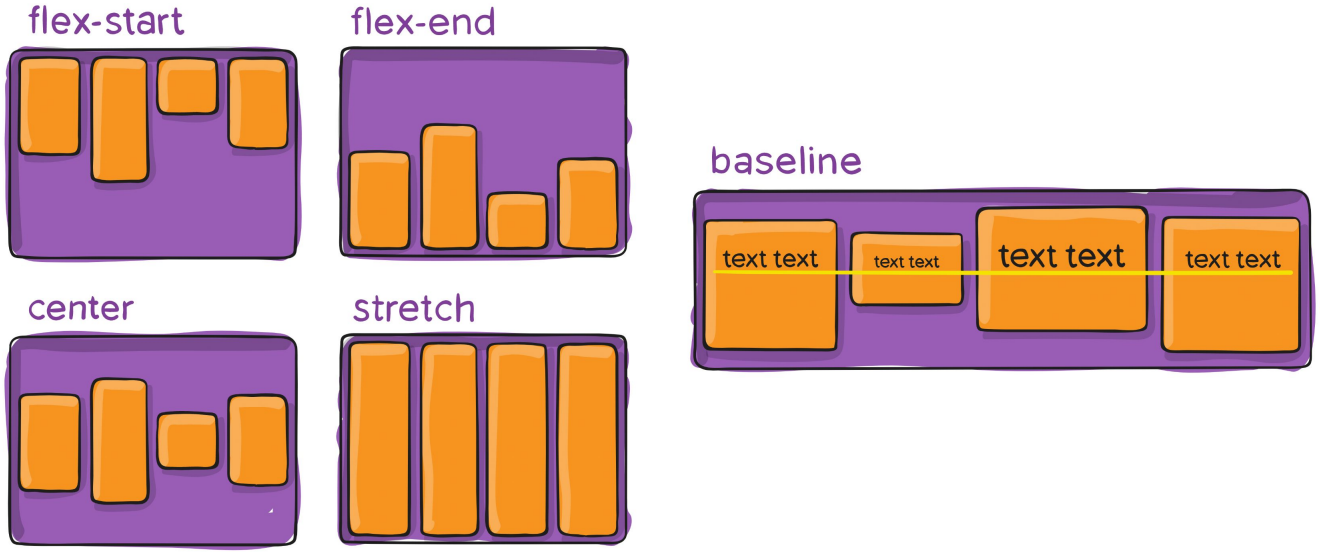


 All content is proprietary and confidential.

Image credit: css-tricks

Flexbox container property: `align-items`

Controls how items align along the **cross axis**

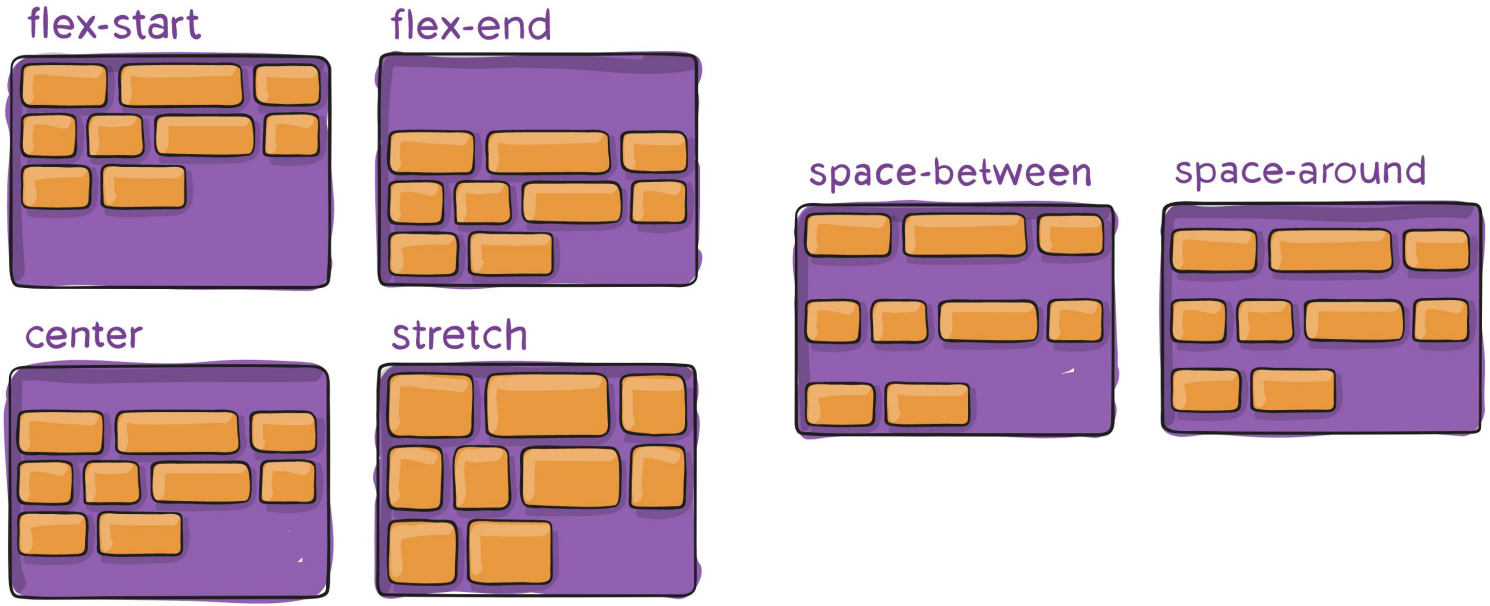


 | All content is proprietary and confidential.

Image credit: css-tricks

Flexbox container property: `align-content`

Controls how the container's lines behave, if there are multiple lines AND there is extra space in the cross-axis



 All content is proprietary and confidential.

Image credit: css-tricks

Flexbox container property: `gap`

Controls the space between flex items.

Think of this like `margin`, except it only applies to the spaces between items, instead of all around an item.

Also: `row-gap`, `column-gap`

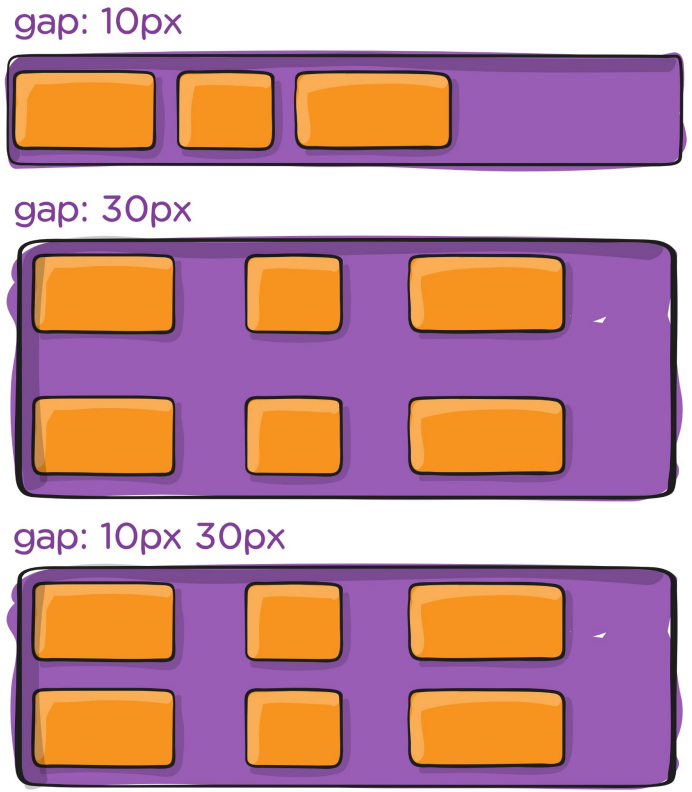


Image credit: css-tricks

Flexbox **item properties**

Properties for flex **items** can control specific things about their own behavior, sometimes overriding their container's properties.

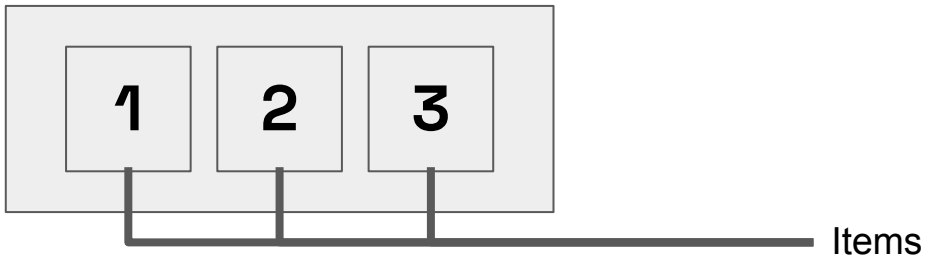
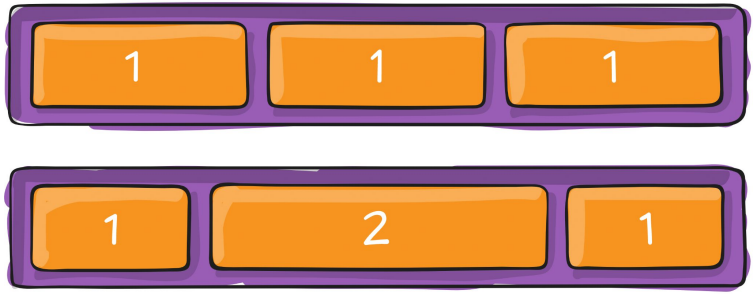


Image credit: css-tricks

Flexbox item property: `flex-grow`

Allows a flex item to grow if necessary, and determines how much space it can grow to occupy relative to other flex items in the same container.



Flexbox item property: `flex-shrink`

Allows flex items to shrink if they are too large to fit in their container, and determines how much they're allowed to shrink relative to other items in the same container.

Image credit: css-tricks

Flexbox item property: `flex-basis`

Sets the initial size of a flex item.

Flexbox item property: `flex`

Shorthand! Lets you define `flex-grow`, `flex-shrink`, and `flex-basis` all in one line!

```
flex: 0 1 auto;
```

Is the same as:

```
flex-grow: 0;
flex-shrink: 1;
flex-basis: auto;
```


Image credit: css-tricks

Flexbox versus the Box Model

1. Defines “boxes” that can align horizontally or vertically

2. Works on two axes

3. Defines “containers” and “items”

4. Uses float, margin, padding, etc to lay out elements

5. Can easily change and reverse direction

FLEX

BOX MODEL

 All content is proprietary and confidential.

Flex Container vs Item Properties

1. flex
2. justify-content
3. align-self
4. flex-direction
5. align-items

CONTAINER

ITEM

 | All content is proprietary and confidential.

Fullstack Academy

BREAK TIME

 All content is proprietary and confidential.

CSS Pseudo Selectors:

Fullstack Academy

Select elements based on special states that often aren't visible in the HTML itself. The most commonly used tend to fall into 4 categories:

1. User Action: When a user performs an action on an element
   a. `:hover, :focus`
2. Tree Structure: How an element is located in the HTML structure
   a. `:nth-child, :first-child, :last-child, :nth-of-type, etc`
3. Input: Related to the state of form elements
   a. `:enabled, :disabled, :checked, :required, :invalid, etc`
4. Negation: `:not()` (yep that's the only one)

Resource: https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes

 All content is proprietary and confidential.

Fullstack Academy

DEMO TIME

 All content is proprietary and confidential.