COMP 1017

Day 16
Flex Wrap & :nth-child( )

Usually, slapping a class on an element and writing up some rules in CSS is an excellent way to apply styles to something.

But there are a tonne of other ways that we can select different elements!

One of these ways is by targeting the order in which an element appears.

:nth-child( )

Web developers definitely don’t love all their children equally.

Does anyone remember pseudo-class selectors?

Pseudo-class selectors can target an element when it’s in a specific state.

We used them previously to make `<a>` tags look different whenever we hovered our cursor over top of them.

```css
a:hover {
  color: pink;
}
```

```css
/* This says, whenever the user rolls
their cursor over an <a> anchor tag,
change the color to pink. */
```

We can also use pseudo-class selectors to target an element based upon its order or position.

```
<main>
     <div>
      <div></div>
      <div></div>
      <div></div>
    </div>
</main>
```

Without adding any classes, how can we target the innermost `<div>`s?

To choose a specific element in a sequence of siblings, we can use:

```
element:nth-child( )
```

Whatever number we use will be the position of the element that we're selecting.

```html
<main>
  <div>
    <div></div>
    <div></div>
    <div></div>
  </div>
</main>
```

```css
main div div:nth-child(1) {
    color: red;
}
```

```
/* This selects the first <div>
 inside of our wrapper <div>,
 which is inside of <main>. */
```

```html
<main>
  <div>
    <div></div>
    <div></div>
    <div></div>
  </div>
</main>
```

```css
main div div:nth-child(2) {
    color: red;
}
```

```
/* This selects the second <div>
 inside of our wrapper <div>,
which is inside of <main>. */
```

```html
<main>
   <div>
      <div></div>
      <div></div>
      <div></div>
   </div>
</main>
```

```css
main div div:nth-child(3) {
      color: red;
}
```

```
/* This selects the third <div>
 inside of our wrapper <div>,
 which is inside of <main>. */
```

Please note that when we’re targeting the position of an element, we start counting at `1`, not `0`.

Let’s use it in a demo!

Please grab (Exercise) Flexbox I from Day 16.

So far, we’ve been using flexbox to place multiple boxes side-by-side.

But what happens if we need multiple rows of boxes within a single flex container?

We can use flex-wrap.

Flex Wrap

Still a weird flex, but ok.

Flex Wrap is a CSS property for flex containers*.

* **not** the flex items!

It determines whether flex items should be forced onto a single line, or if they can wrap onto multiple lines.

```css
main {
   display: flex;
   flex-wrap: wrap;
   }
```

Flex Wrap will only work if there is enough content (or, if there are enough flex items).