



Rev. 08Aug2019

Exercise 17

A PEEK INTO CSS POSITIONING

You've seen how we can use CSS to affect the styling of elements, with things like `font-family`, `text-align`, and `color`. That's cool, but we can do more. We can actually affect where HTML elements appear on the page. This aspect of CSS is often referred to as *CSS positioning*. It's a broad topic — one that's best approached gradually, learning a little each time you approach it. Here's our initial approach.

Run the repl for **Exercise 17**. Now compare that with what we want to end up with for this exercise:

Products			Your Cart		
Red widget	12.95	2	Red widgets	25.90	
Blue widget	9.95	1	Blue widget	9.95	
Green widget	14.95				

The available values for "display" are "block", "inline", and "inline-block". I might have used "inline" but for one problem: inline does not permit me to set the width of an element, which I'll want to do. "inline-block" gives us the display characteristics of inline elements — with some added flexibility.

In a previous exercise, I mentioned that all HTML elements have two types of display properties naturally. *Block-level* elements (such as `<p>`, `<div>`, and the `<h1-6>` elements) automatically start themselves on a new line. *Inline* elements (such as `` and `<output>`) snuggle themselves against the previous element.

This presents us with our first problem: we have two `<div>`s: one with an `id` of `products`, the other with an `id` of `cart` that we want next to each other, even though their natural display type is block-level. Luckily, we can redefine this display property of an element with this CSS property:

```
#products {  
  display: inline-block;  
}
```

```
#cart {
  display: inline-block;
}
```

- Go ahead and try this in your `index.css`.

I'm sure there are CSS masters who write flawless code the first time. I'm not one of them. For me, CSS takes a good deal of trial-and-error, aided by frequent calls to Google. One thing I do to help me as I bungle my way to a solution is to define an `outline` for elements so that I can see where they sit on the page.

- Try adding this to `#products` and `#cart` CSS: `outline: 1px solid red`.

Products	Your Cart
Red widget	2
12.95	Red widgets
Blue widget	25.90
9.95	1
Green widget	Blue widget
14.95	9.95

- I know I want the products to be wider than the cart. I can set this with the `width` property. I set the `width` of `#products` to `300px` and the `width` of `#cart` to `200px`. Try those values. Feel free to change them.
- Back to the HTML. Inside of the `<div>`s with a `class` of `product`, we have two `<p>` elements, one with a `class` of `description` and the other with a `class` of `price`. `<p>` elements have a natural `display` property of `block`. Since we want them next to each other, change the `display` to `inline-block`.
- Run the repl. I've got a problem. Although the two `<p>` elements now align vertically, the `#products` and `#cart` `<div>`s are vertically aligned on the bottom. I want them aligned on the top. A quick consult with Google and I find there's a `vertical-align` property I can use.

```
vertical-align: top.
```

And that fixes it!

Products	Your Cart
Red widget 12.95	2 Red widgets
Blue widget 9.95	25.90
Green widget 14.95	1 Blue widget
	9.95

- ❑ Looking at Products, I want the description and price to be in nice columns, instead of mashed up against each other. Use the `width` property to fix this.
- ❑ Looks better, but we're used to see prices being *right*-aligned. You've used the `text-align` property in a previous exercise. Use it here to right-align prices.
- ❑ Much better! Now, using what we've just done with products as a pattern, fix the cart so that it looks like my initial screenshot above.

NOTE

You might think I'm being self-deprecating when it comes to CSS. Surely, you can't be a successful, professional programmer and still be relying on Google so much? I assure you there's no false humility at play. In fact, successful, professional programmers use Google a *lot*. Humans aren't great at keeping reams of information in their heads. Google is. (Just don't tell anyone. We wouldn't want to break the spell...)