

Style Conference Part IV.II

AIM CODE SCHOOL FOUNDATIONS OF WEB DEVELOPMENT

With text properties under our belt, let go back into our style conference website and put our knowledge to work.

Currently every link on the page is underlined, which is the default style for anchor elements. We'll begin by removing the underline from all anchor elements by way of the **text-decoration** property. Next, we'll select all anchor elements that appear within a paragraph element and give them a bottom **border**.

We could use the **text-decoration** property instead of the **border-bottom** property to underline all the links within each paragraph; however, by using the **border-bottom** property we have more control over the underline's appearance. Here, for example, the underline will be a different color than the text itself.

Our links section, which includes our previous hover styles, should now look like this.

```
320  a {
321    |   color: #6c89db;
322    |   text-decoration: none;
323  }
324
325  a:hover {
326    |   color: #a9b2b9;
327  }
328
329  p a {
330    |   border-bottom: 1px solid #dfe2e5;
331  }
```

Going back to our `<h5>` elements from before, which have slightly different styles than the rest of the headings, let's make them all uppercase using the **text-transform** property. Our new `<h5>` element styles should look like this:

```
281  h5 {
282    |   color: #4568c9;
283    |   font-size: 14px;
284    |   font-weight: 400;
285    |   text-transform: uppercase;
286  }
```

Let's revisit our `<header>` element to apply additional styles to our navigation menu (to which we previously added the `primary-nav` class attribute value). After the existing `font-size` and `font-weight` properties, let's add some slight `letter-spacing` and change our text to all `uppercase` via the `text-transform` property.

Our styles for the `<nav>` element with the `primary-nav` class attribute value should now look like this:

```
402 .primary-nav {
403     font-size: 14px;
404     font-weight: 400;
405     letter-spacing: 0.5px;
406     text-transform: uppercase;
407 }
```

Previously, we floated our logo to the `left` within the `<header>` element. Now our tagline sits directly to the right of the logo; however, we'd like it to appear all the way to the right of the `<header>` element, flush right.

We need to add the `text-align` property with a value of `right` to the `<h3>` element with the class attribute value of `tagline` to get the `tagline` to sit all the way to the right.

When added to the existing `margin` property, our new styles for the `<h3>` element with the class attribute value of `tagline` will look like this:

```
397 .tagline {
398     margin: 66px 0 22px 0;
399     text-align: right;
400 }
```

We'd also like our navigation menus, both in the `<header>` and `<footer>` elements, to sit flush right. Because both the `<header>` and `<footer>` elements have child elements that are floated to the left, we can use the same approach as we did with our tagline.

The floated elements within the `<header>` and `<footer>` elements are taken out of the normal flow of the page, and this causes other elements to wrap around them. In this specific case, our navigation menus are the elements wrapping around the floated elements.

Because we'll be sharing the same styles across both navigation menus, we'll give them each the class of `nav`. Our `<header>` element will now look like this:

```
88 <header class="container group">
89     <h1 class="logo">...</h1>
90     <h3 class="tagline">...</h3>
91     <nav class="nav primary-nav">
92         ...
93     </nav>
94 </header>
```

And our `<footer>` element will now look like this:

```
88 <footer class="primary-footer container group">
89
90   <small>...</small>
91
92   <nav class="nav">
93     ...
94   </nav>
95
96 </footer>
```

Let's not forget, changes to our `<header>` and `<footer>` elements need to be made on every page.

With the `nav` class in place on both navigation menus, let's create a new section within our `main.css` file to add shared navigation styles. We'll begin by adding the `text-align` property with a value of `right` to a `nav` class rule set. We'll expand these styles later on, but this will serve as a great foundation.

```
429 /*
430  |=====
431  | Navigation
432  |=====
433 */
434
435 .nav {
436   text-align: right;
437 }
```

While we're adding the `text-align` property to a few different elements, let's also add the `text-align` property with a value of `center` to our `hero` class selector rule set. For reference, these styles, including our existing `line-height` and `padding` properties, are located within the home page section of our `main.css` file.

```
66 .hero {
67   line-height: 44px;
68   padding: 22px 80px 66px 80px;
69   text-align: center;
70 }
```

Let's also set the paragraph element within our hero section to that weight. We can use the existing selector to do so, and the new rule set will look like this:

```
467 .hero p {  
468     font-size: 24px;  
469     font-weight: 100;  
470 }
```

Great Work!! Our DSC Site is coming along nicely! This is what our website should look like in the browser thus far...

