1.7.8 Create New Styles

We're nearing the finish line! We just need to add a few more styles to the Privacy Policy page to get it looking like the mock-up.

Using everything we've learned about CSS, let's tackle the page's title first. We'll start by defining which selector will be used. The quickest way to do this is to select the element by its class, `page-title`.

Here are some specifications for how it should look:

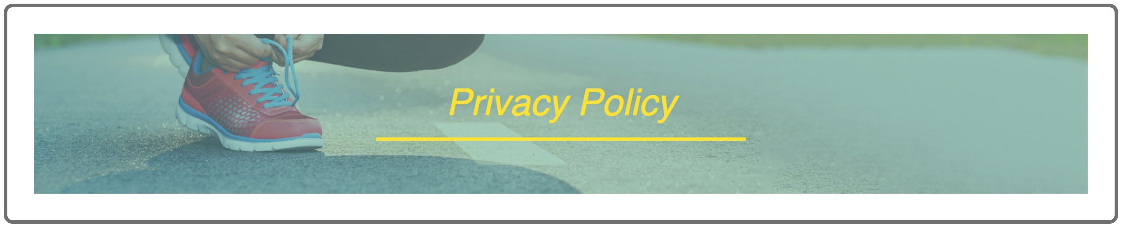
- The font `color` should be `#fce138`.
- The `<h2>` tag should not go full-width, so change its `display` property to make it an `inline-block` element.
- Give its `border-bottom` a value with a `4px` width, a `solid` style, and the same color as the `color` property above.
- Make the border run wider than the text and give it some space by applying the following `padding` to its sides:
    - `top: 0`
    - `right: 80px`

  - `bottom: 15px`
  - `left: 80px`

- Lastly, adjust the font styles as follows:
  - Set the `font-weight` to `normal`; this makes the default bold `<h2>` not bold anymore.
  - Change the `font-size` to 42px.
  - Here's a new one: set the `font-style` to `italic`. As you can probably assume, `font-style` is a CSS property that creats slanted (italicized) text.

If any of the spacing seems off, remember that you can always use Chrome's DevTools to adjust and see how certain styles will look before actually applying them. This saves a lot of trial-and-error time.

The result should look like this:



Now we'll add styles to the `secondary-content` class and its child elements using class selectors and nested selectors. Add the following styles into `secondary-styles.css` so you don't accidentally overwrite any styles for `index.html`!

The styles for `secondary-content` are as follows:

- Set the `width` to 80%.

- Center it on the page by using `margin`. Think back to how you centered elements in previous lessons using this property; the values here will be very close if not the same.

- Set the default font `color` for all text in this `<article>` to have a value of `#024e76`.

Style the `<h3>` tags in `secondary-content` (use nested selectors to make sure it only applies to these tags and no other `<h3>` tags):

- Give it a `font-size` of 25px.

- Set its `margin` to have 20px on the top and bottom, and 0 on the left and right.

Style the `<p>` tags in `secondary-content` (use nested selectors to target only these `<p>` tags):

- Give it a `font-size` of 16px.

- Make its `line-height` a little bit bigger by giving it a value of 1.5.

- Give it the same `margin` values you gave the `<h3>` tag above.

Style the `<ul>` tags in `secondary-content`:

- Set the `margin` to have 15px on the top and bottom and 20px on the left and right.

Finally, style the `<li>` tags in `secondary-content`:

- Make them stand out from the rest of the text by giving them a `color` value of `#39a6b2`.

- Give them some space by adding a `margin` of 10px to the top and bottom, and 0 for left and right.

HIDE HINT

Make sure that you save and refresh the page often to track your progress.

Also don't forget to keep Chrome's DevTools open while you work to confirm that the styles you define get the results you want.

And there you have it. You just used HTML and CSS to create your first project! Our friends at Run Buddy will be thrilled to see what we've put together for them.

The last thing you need to do is get this finished product onto the internet for the world to see. Go ahead and do another `git add`, `git commit`, and `git push` and then admire the results of all your hard work on the GitHub project page.

But first, let's check your knowledge about `<head>` and `<body>` elements.

Final score: 100%

Retake