In this section, we will implement the top four layers of the ITCSS architecture: the settings layer, the tools layer, the generic layer and the elements layer.

As we learned in the last section, our constants will go to the settings.scss file. We are going to define the base font size, base-line-height and other settings belonging to our document:

```css $base-font-size: 12px; $base-line-height: 18px;

// https://coolors.co/0a2342-b9d6f2-4f7cac-662020-cacfd6 $ci-darkblue: #0A2342; $ci-lightblue: #B9D6F2; $ci-midblue: #4F7CAC; $ci-red: #662020; $ci-grey: #CACFD6; $ci-lightblue-bg: #E9F6FF; ```

You can use the coolors.co service to create some colors that make sense.

In the tools layer, we are going to create some mixins that we will use later in the other layers. This layer is still not responsible for generating any code whatsoever:

```css @mixin rounded-border($thickness, $radius) { border-radius: $radius; border: $thickness solid $ci-midblue; }

@mixin sky-gradient() { background: #2c3e50; /* fallback for old browsers */ background: -webkit-linear-gradient(to right, #2c3e50, #3498db); /* Chrome 10-25, Safari 5.1-6 */ background: linear-gradient(to right, #2c3e50, #3498db); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */ } ```

Notice that we're using some of the Sass variables from the top layer already. This is why the settings layer is strictly before the tools layer.

Next, generic. This time we're not taking care of supporting any specific browsers. Obviously the browser that you specify depends on your project requirements. Further to your product requirements, oftentimes it makes sense to check your analytics data to figure out which browsers you can get rid of. After all, there is nothing worse than spending a significant amount of time to support a browser that no one ever watches, or maybe 0.01% of your users watch.

```
css @import "normalize";
```

Download Normalize.css at https://necolas.github.io/normalize.css/.

We will now create our components HTML file. I will not go into the details of the file structure and the modeling and the chosen classes because you will see them in action quite soon when we will style the top layers. You can view the components HTML file in the code examples attached to this course.

```css

Textfields, Textareas
Label Placeholder Label Placeholder
Label Placeholder
Placeholder
Long Label
Checkboxes, Radio Buttons, Dropdowns, Buttons
Label One Label One Two
Label Option 1
Press me

```

Jump into the elements.scss file and start styling:

```css html { font-size: $base-font-size; line-height: $base-line-height; }

body { font: 1rem "Segoe UI", Helvetica, Arial; background-color: $ci-grey; color: $ci-darkblue; }

form { width: 80%; margin: 4rem auto; font-size: 1.5rem; }

label { cursor: pointer; }

fieldset {
font-size: 2.25rem; line-height: 1.5;
}

label, input, button, textarea, select { font-size: 1.5rem; } ```

This concludes all the generic styling belonging to our component library. In the next section, we're going to take care of the next two layers, namely objects and components.
```