**CSS in JS**

CSS in JS is a distinct approach to styling. The main idea is that styling is handled by JavaScript objects rather than traditional CSS. Styles can be written inline or accessed via object variables, but React Native offers a StyleSheet API that provides a performant and compositional way to style components.

Now that we've seen React Native handle *styling*, how do we manage the *layout* of a mobile application? We'll take a look at CSS's **flexbox** in the next section to do just that!

**Further Learning**

* [**How can I use CSS-in-JS securely?**](https://reactarmory.com/answers/how-can-i-use-css-in-js-securely)

**Flexbox**

React Native leverages a version of **flexbox** to build component layout. This is primarily due to flexbox's ability to provide consistent layouts across different screen sizes.

Flexbox containers comprise of two axes: a **main axis**, as well as a **cross axis**. Some of the more critical properties to consider when building layouts with flexbox include flex-direction, justify-content, and align-items. React Native's implementation of flexbox \_is\_ a bit different, however. We'll see just how in the very next section!

**Further Research**

* [**A Complete Guide to Flexbox**](https://css-tricks.com/snippets/css/a-guide-to-flexbox/)
* [**Flexbox Froggy**](http://flexboxfroggy.com/)