

Don't Forget React



The Virtual DOM

The "engine" of React, that updates the Real DOM when needed.

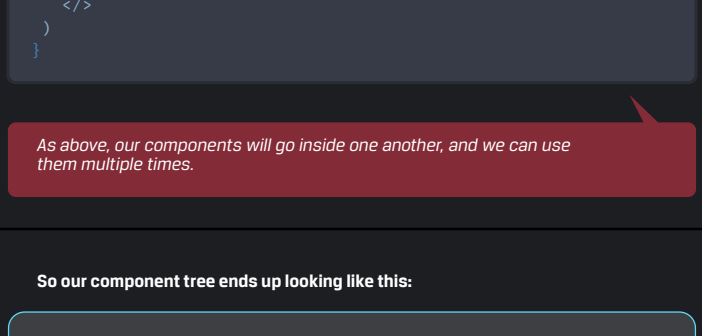
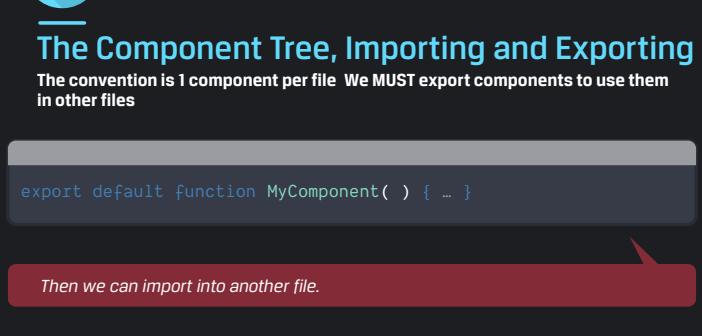
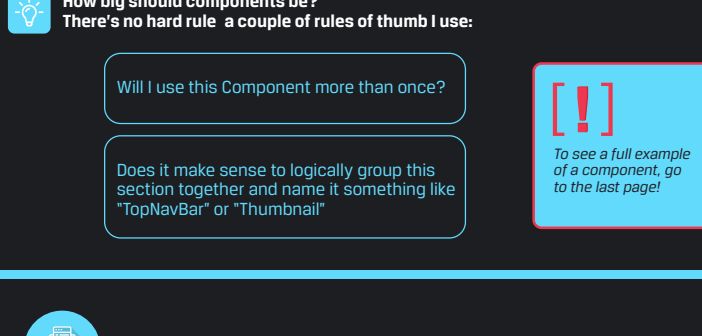
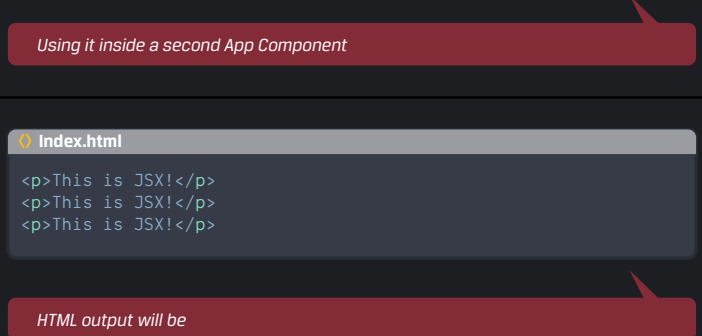
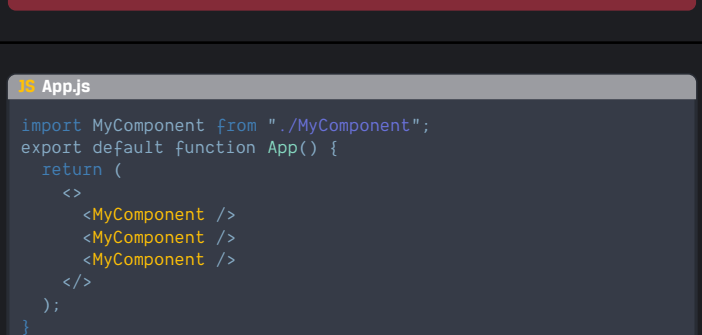
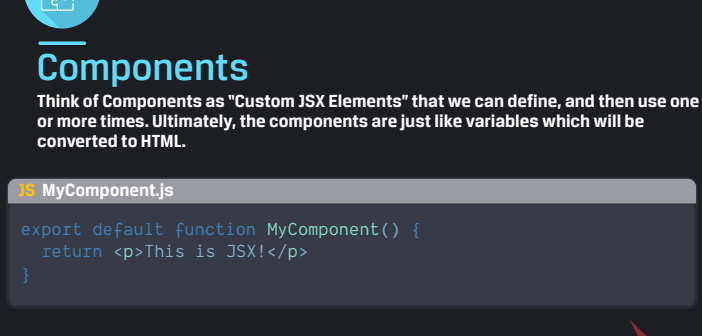
The "Virtual DOM" checks the old "Virtual DOM" - Changing the virtual DOM is much faster, because it's just a JavaScript object.

Then, depending on what changed, it will update the real DOM!



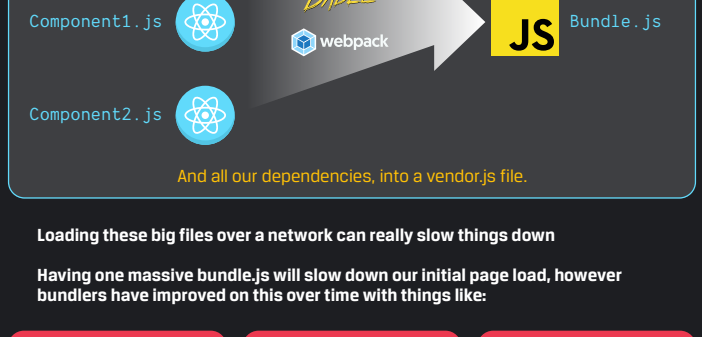
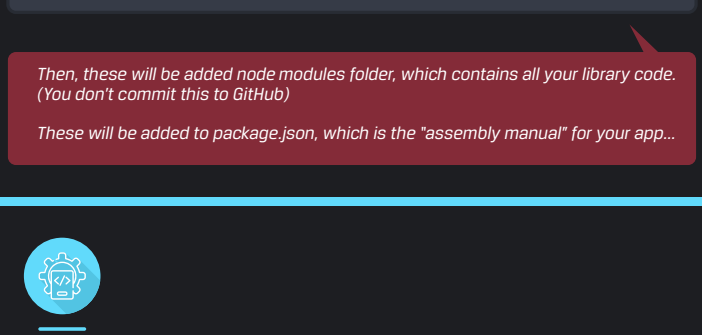
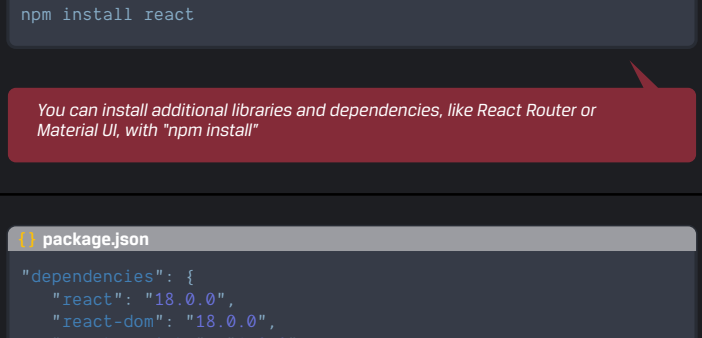
JSX - the special language of React

Almost like HTML, with a few key differences



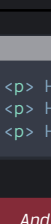
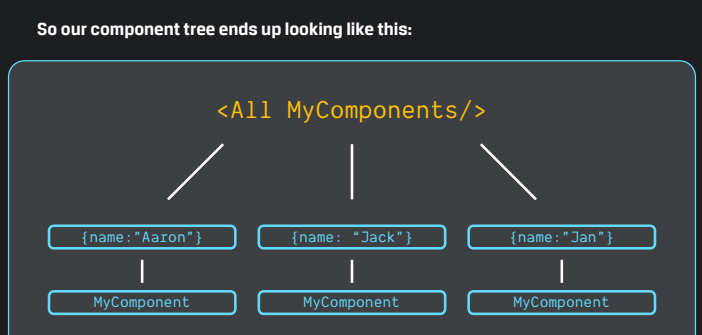
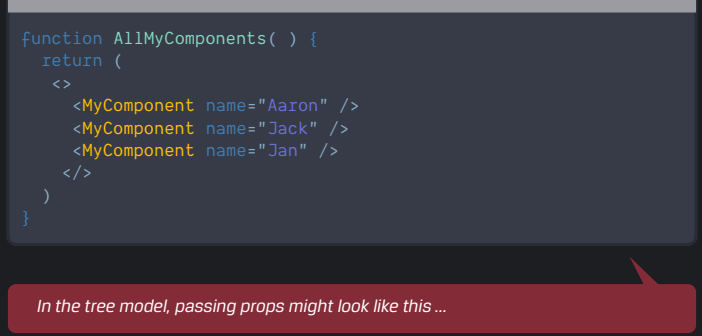
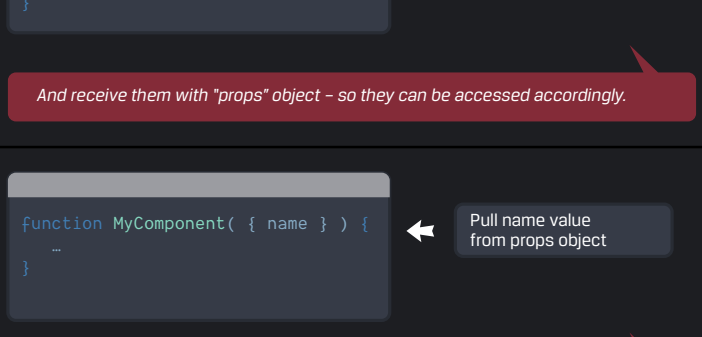
Components

Think of components as "Custom JSX Elements" that we can define, and then use one or more times. Ultimately, the components are just like variables which will be converted to HTML.



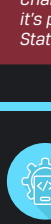
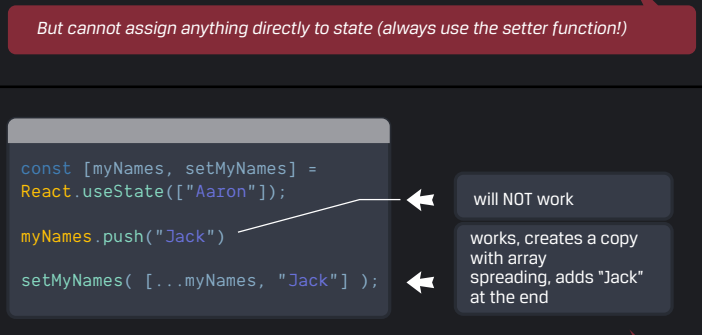
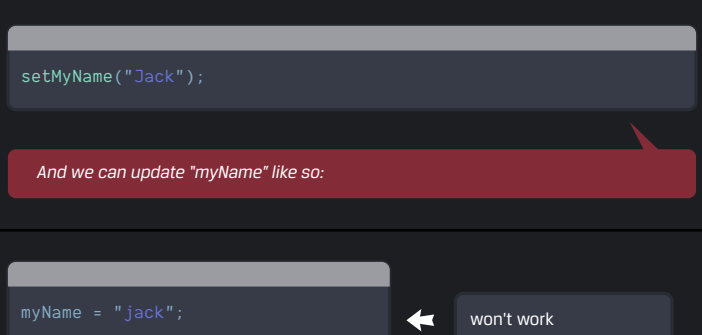
The Component Tree, Importing and Exporting

The convention is 1 component per file. We MUST export components to use them in other files



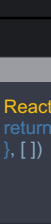
How to Install React & NPM

We can install React with NPM and NodeJS locally. We will need the React library, and its dependencies. It's generally recommended to use a framework like Create-React-App, Vite, or NextJS rather than starting from zero. This will give you boilerplate and save you time.



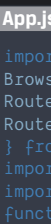
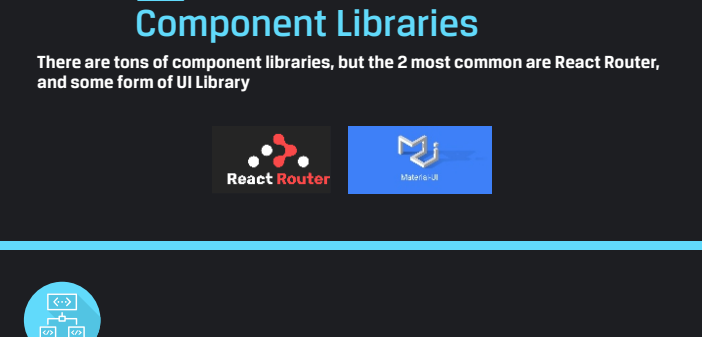
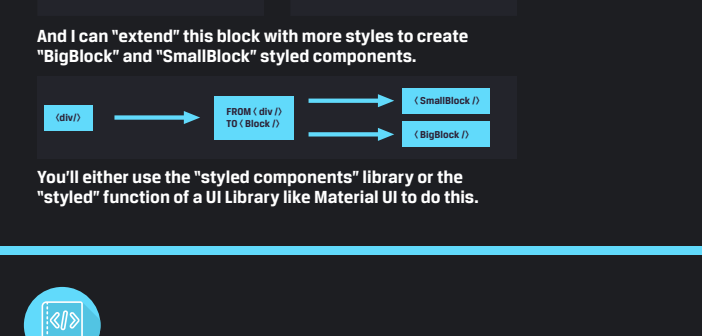
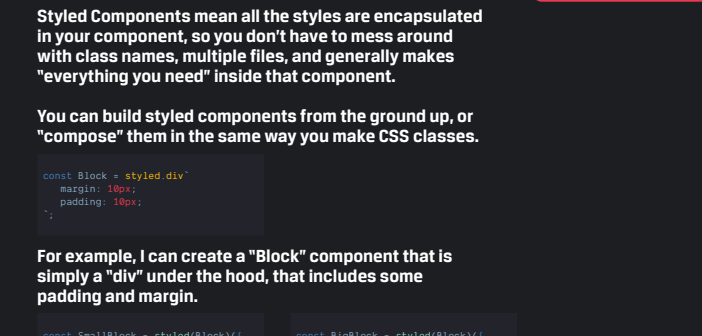
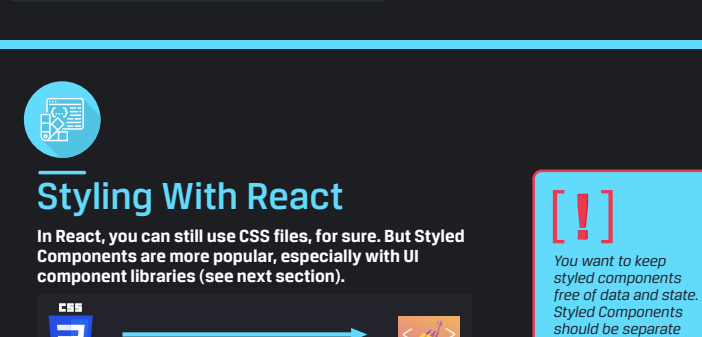
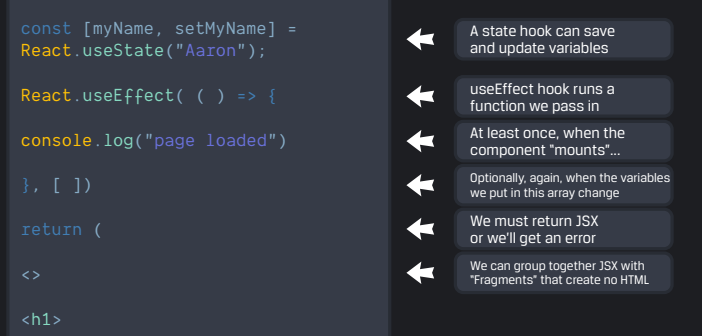
Transpiling

Browsers can't run React code, so it must be TRANS-piled to vanilla JavaScript. HTML and CSS. There are libraries that do this for us. Babel and Webpack.



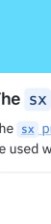
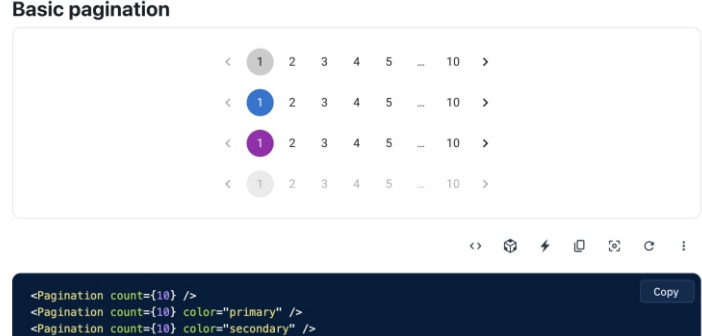
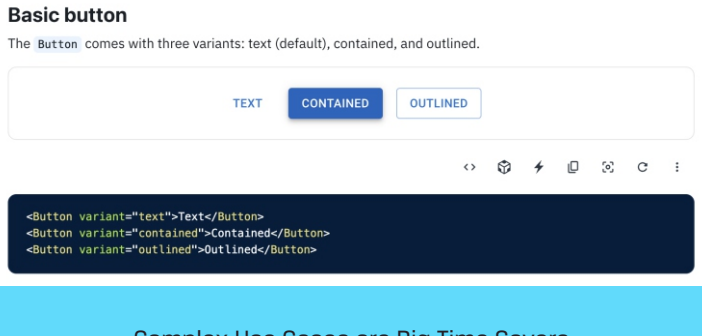
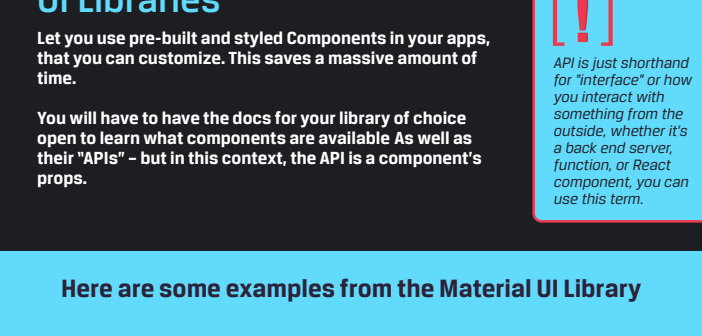
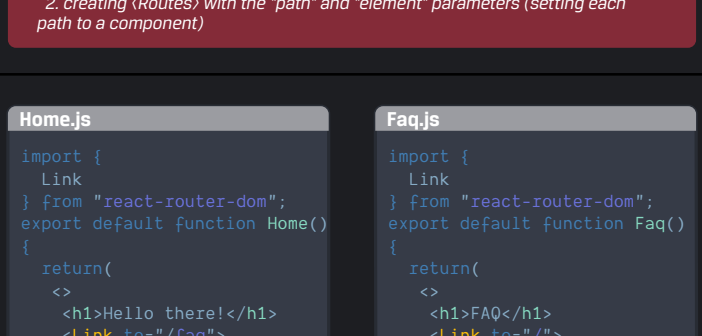
Props

The "arguments" of a component, they're how we pass data from one component to another.



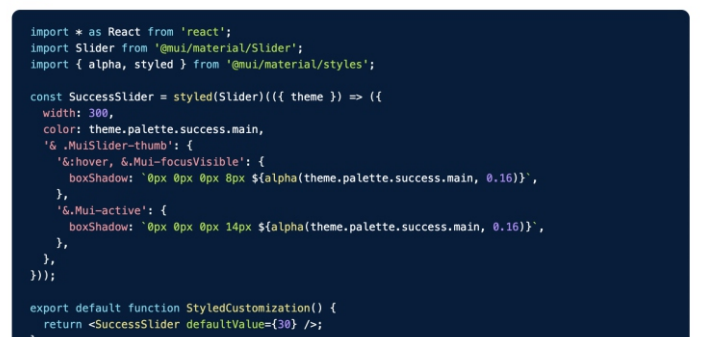
State & State Hook

The "internal variables" of a component.



useEffect & React Life Cycle

When components get added to the page, they start the "component life cycle". There are 3 key events: Mount, Update, Unmount



Full React Component Example

Styling With React

In React, you can still use CSS files, for sure. But Styled Components are more popular, especially with UI component libraries (see next section).

Component Libraries

There are tons of component libraries, but the 2 most common are React Router, and some form of UI Library

React Router

React Router enables us to do "Client Side Routing", so we can have some navigation in our React apps with multiple pages and paths that can be observed in the URL bar.

UI Libraries

Let you use pre-built and styled Components in your apps, that you can customize. This saves a massive amount of time.

Here are some examples from the Material UI Library

Simple Components Are Attractive Out of the Box

To make the same overrides in different locations across your application, create a reusable component using the styled() utility.

