Meet the React Component

Let's learn about the react component.

Our first React component is in the *src/App.js* file, which should look similar to the example below. The file might differ slightly because create-react-app will sometimes update the default component's structure.

```
import React from 'react';
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          Href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;
```

src/App.js

This file will be our focus throughout this tutorial unless otherwise specified. Let's start by reducing the component to a more lightweight version for getting you started without too much boilerplate code from create-react-app.

```
import React from 'react';

function App() {
```

```
  return (
    <div>
      <h1>Hello World</h1>

    </div>
  );
}

export default App;
```

src/App.js

- First, this React component, called App component, is just a JavaScript function. It's commonly called **function component**, because there are other variations of React components (see **component types** later).
- Second, the App component doesn't receive any parameters in its function signature yet (see **props** later).
- Third, the App component returns code that resembles HTML which is called JSX (see **JSX** later).

The function component possesses implementation details like any other JavaScript function. You will see this in practice in action throughout your React journey:

```
import React from 'react';

function App() {
  // do something in between
  return (
    <div>
      <h1>Hello World</h1>
    </div>
  );
}

export default App;
```

src/App.js

Variables defined in the function's body will be re-defined each time this function runs, like all JavaScript functions:

```
import React from 'react';

function App() {

  const title = 'React';

  return (
```

```
    <div>
      <h1>Hello World</h1>
    </div>
  );
}

export default App;
```

src/App.js

Since we don't need anything from within the App component used for this variable – e.g. parameters coming from the function signature – we can define the **variable outside of the App component** as well:

```
import React from 'react';

const title = 'React';

function App() {
  return (
    <div>
      <h1>Hello World</h1>
    </div>
  );
}

export default App;
```

src/App.js

The complete code:

```
import React from 'react';

const title = 'React';

function App() {
  return (
    <div>
      <h1>Hello World</h1>
    </div>
  );
}

export default App;
```

Let's use this variable in the next section.

Exercises:

- If you are unsure when to use `const`, `let` or `var` in JavaScript (or React)

for variable declarations, make sure to read more about their differences.
    - var
    - const
    - let
- Think about ways to display the `title` variable in your App component's returned HTML. In the next section, we'll put this variable to use.