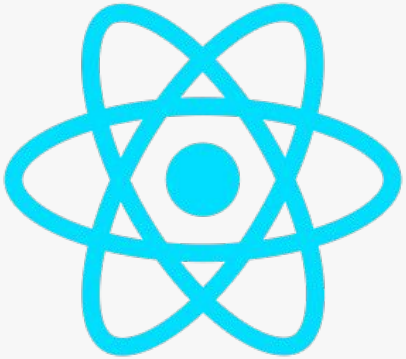
Newton School of Technology



The Ultimate React Course

Lecture 13: Introduction to React

-Vishal Sharma

JS

Prerequisites:

- Basic Web Development Knowledge
  - HTML
  - CSS
  - Javascript
- Text Editor
  - VS Code

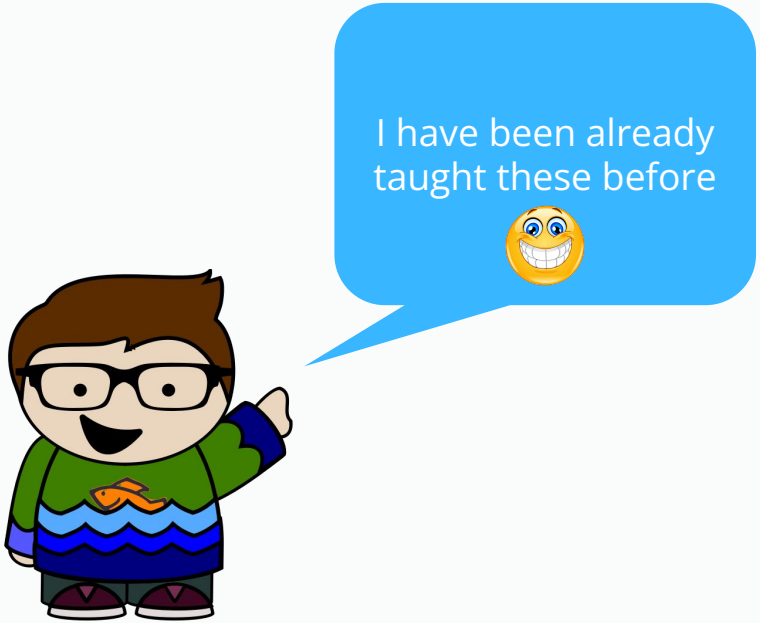
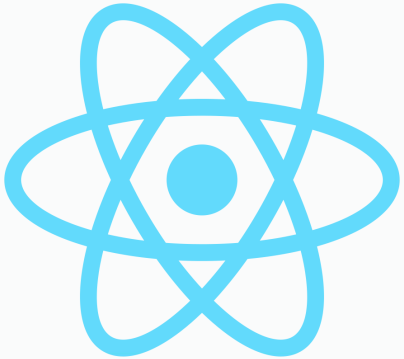
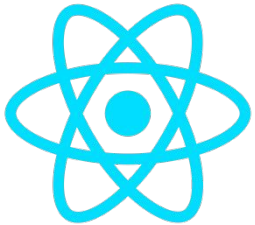


Table of Contents





What is React?

A brief Overview

Have you ever visited a restaurant?

Newton School of Technology

In a local restaurant, the chef makes each dish from scratch, providing flexibility but it can be slow and labour intensive. In a franchise, standardized recipes and pre-made ingredients make the process faster and more consistent.

Local Restaurant

Franchise Restaurant

JavaScript: Local Restaurant of Web Dev

Newton School of Technology

Just like a local restaurant, JavaScript is flexible but can be slow but difficult to maintain and labor-intensive as the app grows.

Can’t make from scratch

Fed up rewriting the code..

React: The Franchise Model

Newton School of Technology

React just like a franchise restaurant, it offers standardized processes, reusable components, and faster, more consistent app development.

Cooked this perfect dish in few moments..

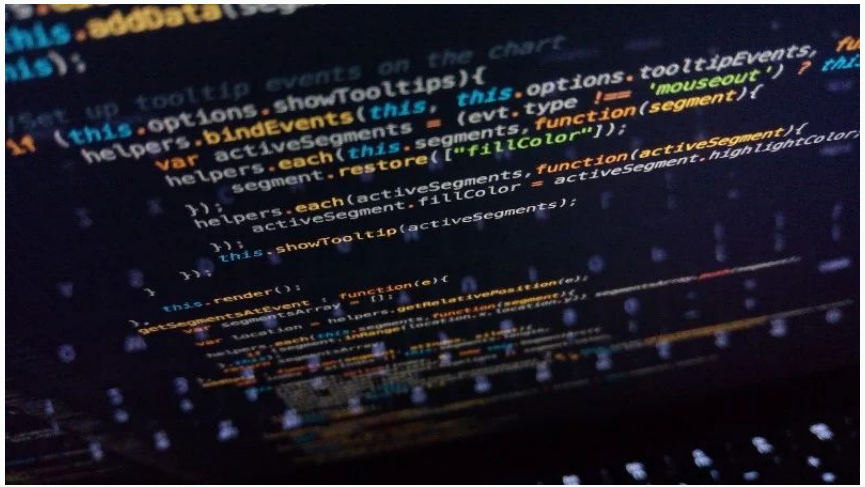
Wrote two day code in 2 hrs..

Struggles of JavaScript Development

Newton School of Technology

As web applications became more dynamic, codebase grew larger javascript struggled with low performance, inefficient updates, and complex code was hard to manage and maintain.

It’s Struggle to maintain JavaScript websites



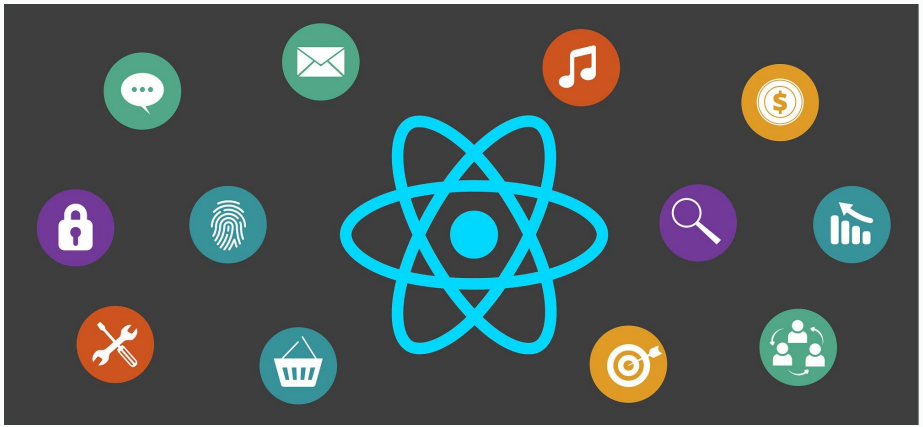
Code getting too Complex

React: The Savior of Web Development

Newton School of Technology

In 2011, Jordan Walke at Facebook, tired of the ongoing frustrations, set out to find a better way. He envisioned a tool that could make the development process smoother and more efficient.

Jordan Walker



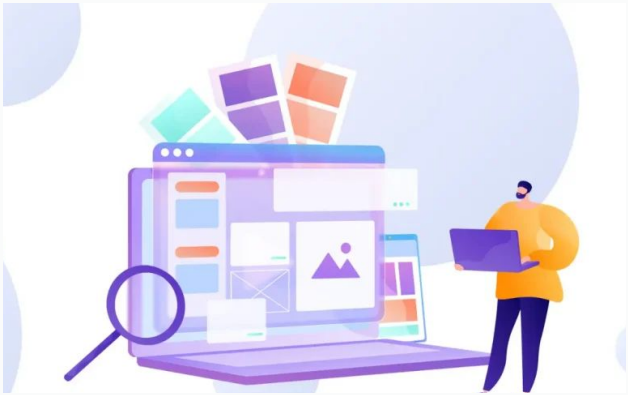
React making life easy for developers

What is React?

Newton School of Technology

React is a JavaScript library for building user interfaces, maintained by Facebook.

Library for building web interfaces

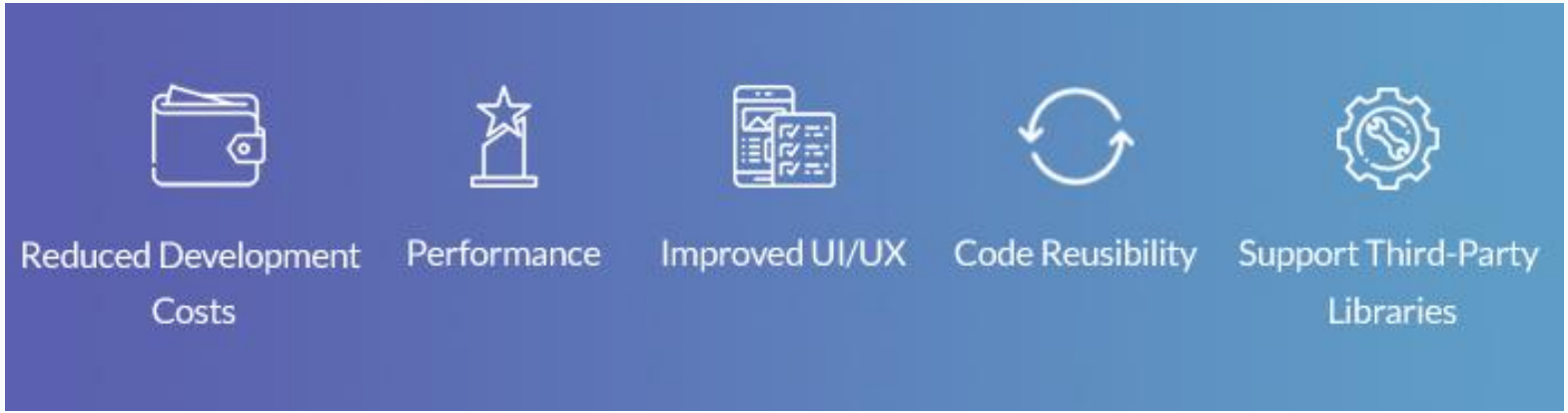


Build beautiful interfaces with react

Why use React?

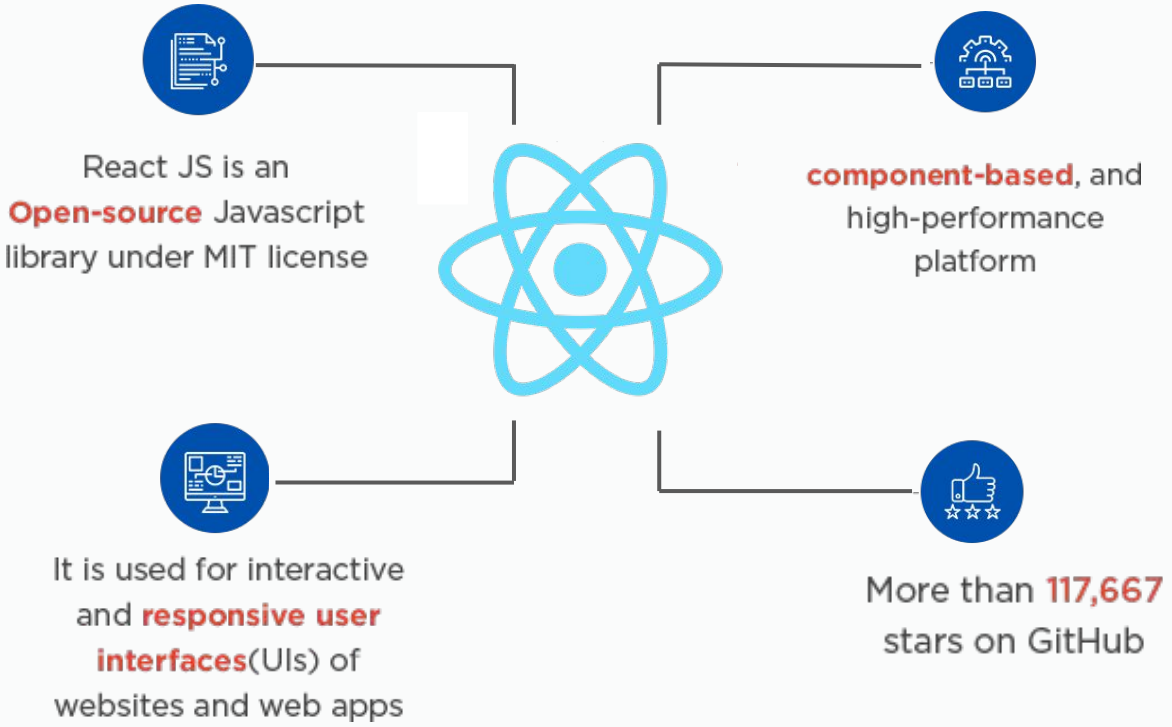
Newton School of Technology

React is most widely library for building single page websites. For the following reasons:-



Key Features of React

The following are the key features of react:-



Don't worry if you don't understand few of the terms, we will explain each of them throughout our course.

Popular websites built on React

The following are few popular websites which uses React:-

Have you identified your dream company among these??

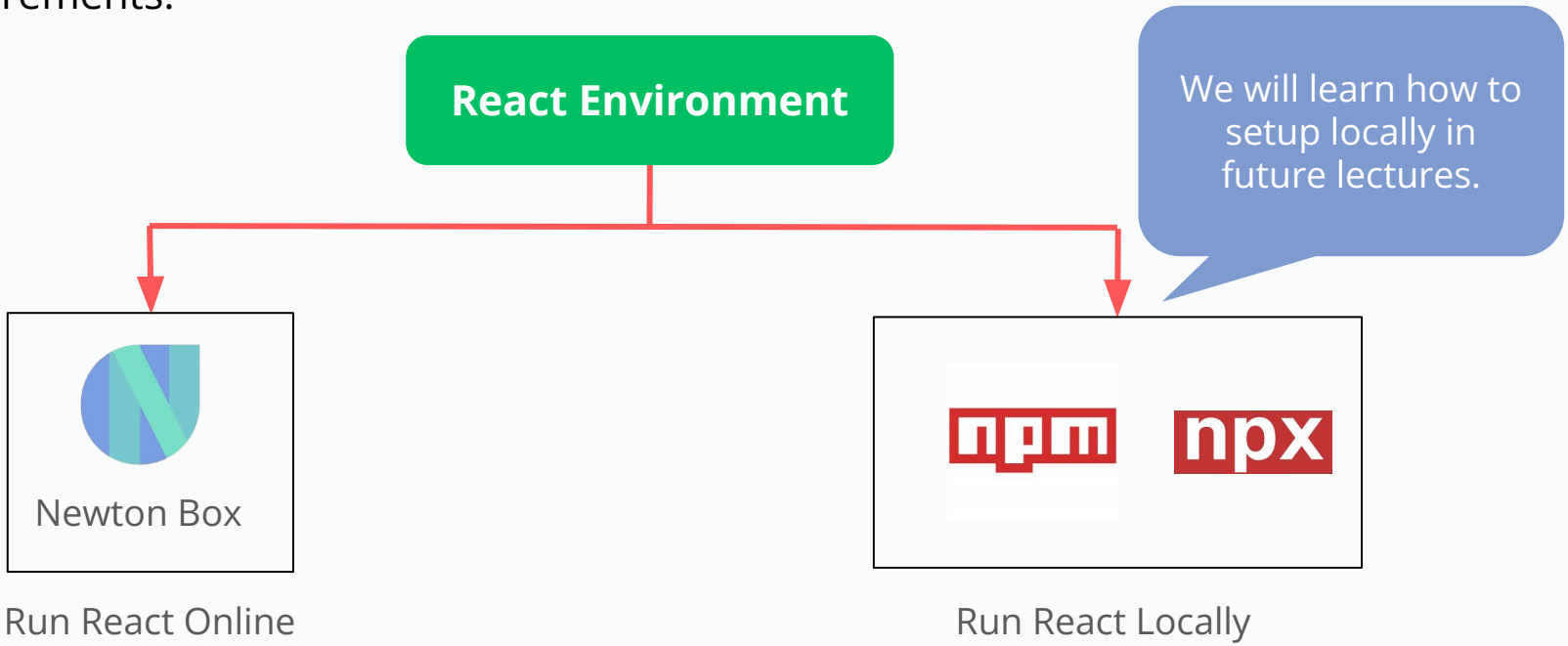
Newton School of Technology

Setting up a React Project

How to Run React Code

Newton School of Technology

React applications can be run in various ways, depending on the development setup and requirements.

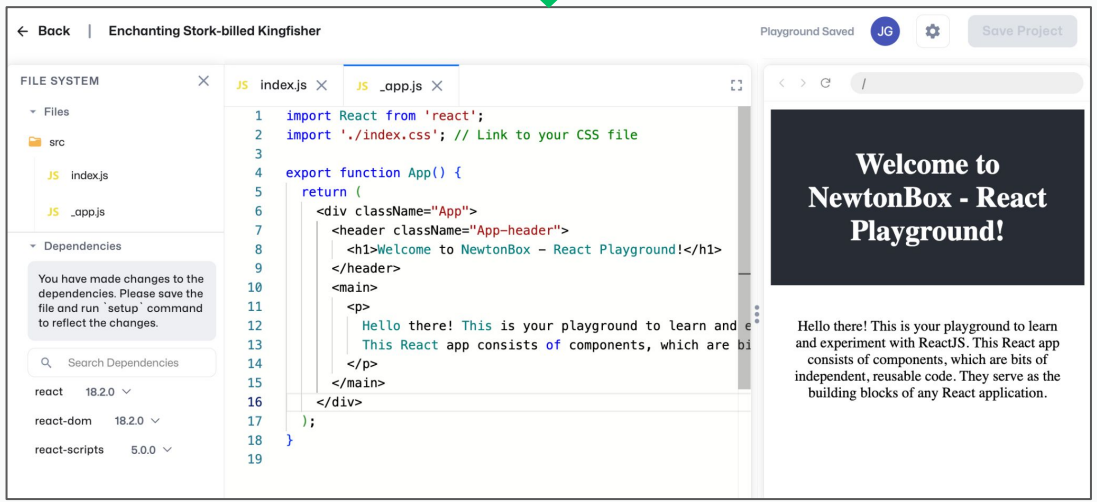


Run React Online: Newton Box Playground

We can run React code effortlessly without worrying about setup by using remote tools. Let's explore how to run React code using Newton Box.

https://my.newtonschool.co/playground/newton-box/3mmvu006csv0

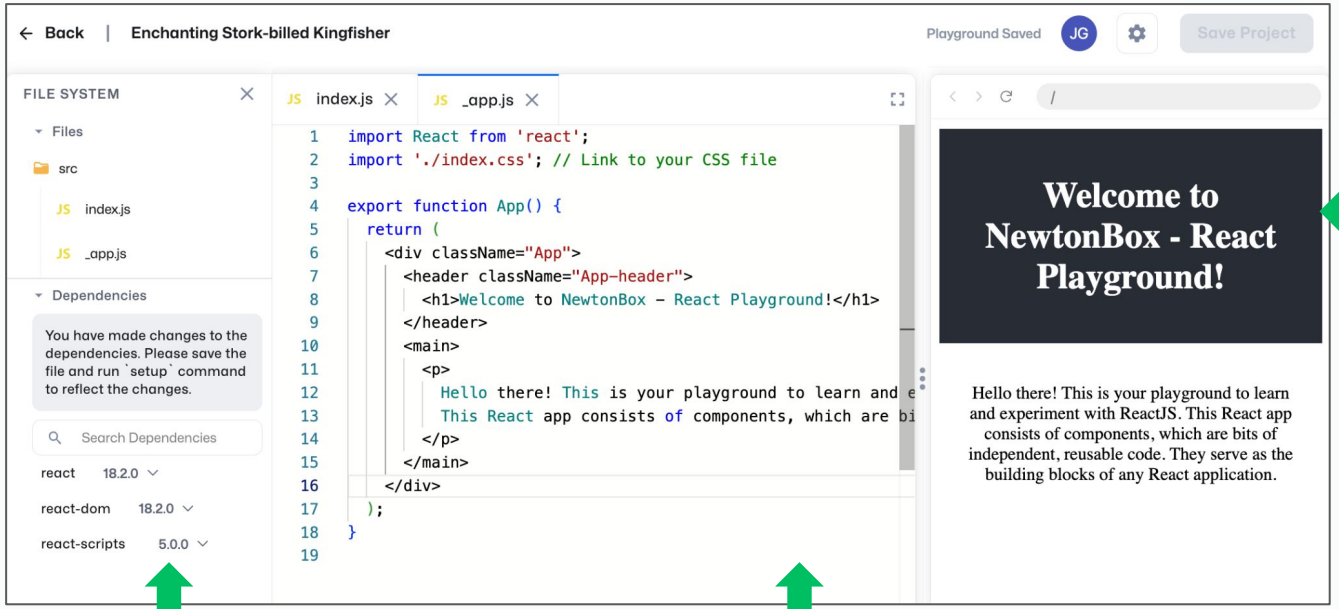
Click on this Link to open Newton Box



In Newton Box you can directly write React Code and view its preview in side browser.

Run React Online: Newton Box

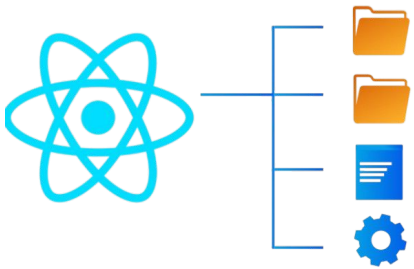
Let’s analyse UI of Newton Box:-



In built browser to see the changes

Here you can open Project files/folders and install dependencies

Here you will write your React code

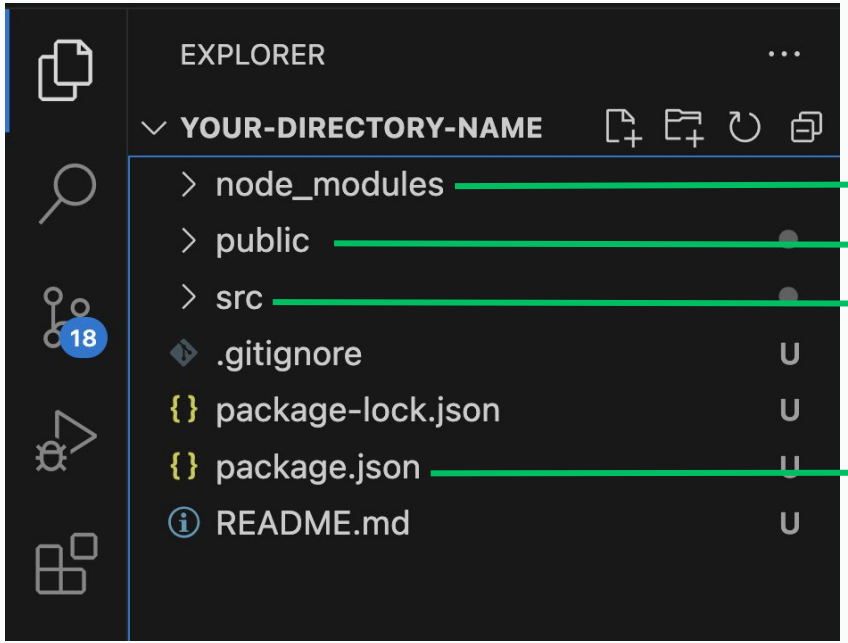


Understanding Project Structure

Project Structure

Newton School of Technology

In file explorer you can see multiple folders and files being created, let's understand each one by one:-



Contains all third-party dependencies installed via npm or yarn.

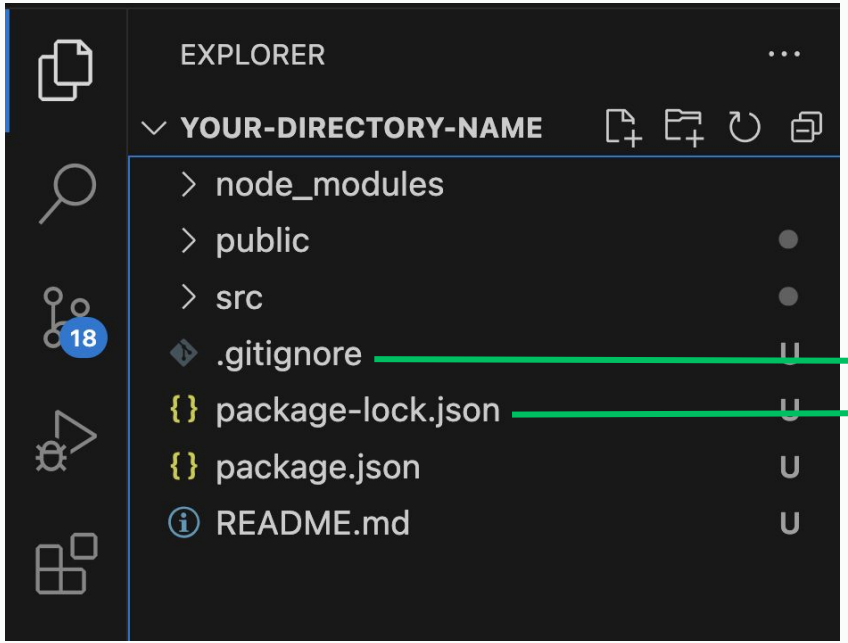
Contains static files directly served by the browser.

Contains all the application code

Lists dependencies, scripts, and project metadata

Project Structure

Here we have two more files:-

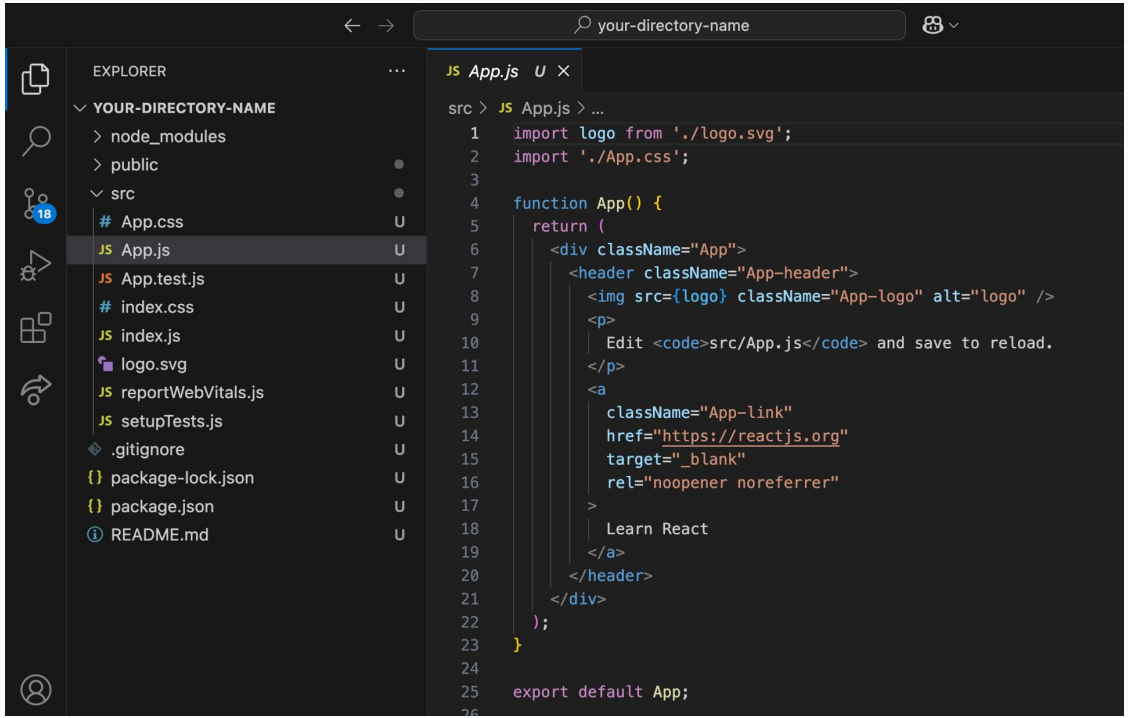


Here you can specify which files needs to be ignored by git.

Its primary purpose is to ensure consistent installations across different environments. You would be rarely updating this file manually.

Project Structure: src folder

Let’s open the src folder and see its contents:-

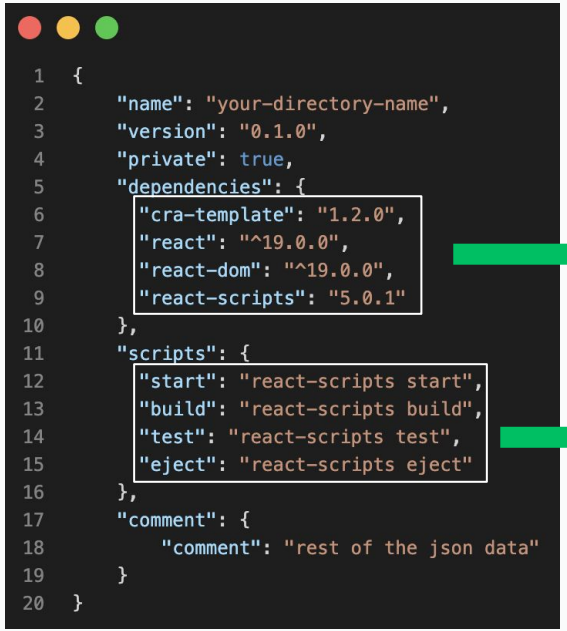


Here, you'll find multiple files. App development begins with App.js, the main component of the application. The App.css file is used to style and enhance the visual appearance of the content in App.js.

Project Structure: package.json

Newton School of Technology

📦 package.json in a React project is a file that manages project settings, dependencies, and scripts to run, build, and test the app.



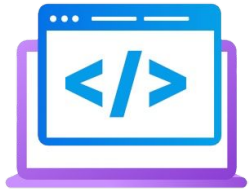
```
{
    "name": "your-directory-name",
    "version": "0.1.0",
    "private": true,
    "dependencies": {
      "cra-template": "1.2.0",
      "react": "^19.0.0",
      "react-dom": "^19.0.0",
      "react-scripts": "5.0.1"
    },
    "scripts": {
      "start": "react-scripts start",
      "build": "react-scripts build",
      "test": "react-scripts test",
      "eject": "react-scripts eject"
    },
    "comment": {
        "comment": "rest of the json data"
    }
}
```

Dependencies installed automatically with React

Script commands with an alias for npm

We can execute these scripts with npm:-

npm start
npm build
.....

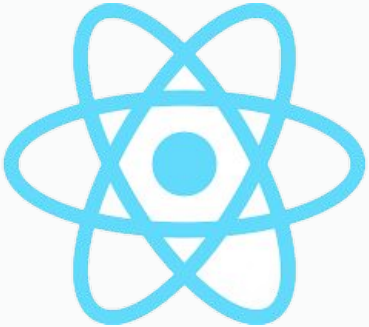


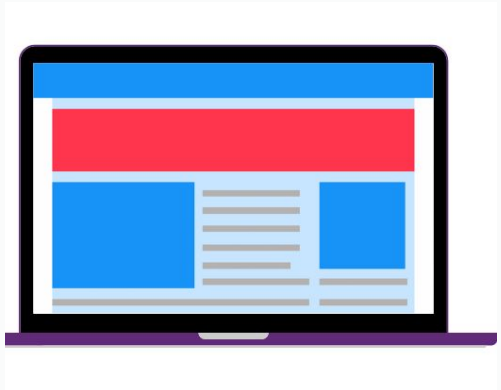
Functional Components

React Functional Components

Newton School of Technology

React functional components are JavaScript functions that return JSX to define and render UI elements in a React application.

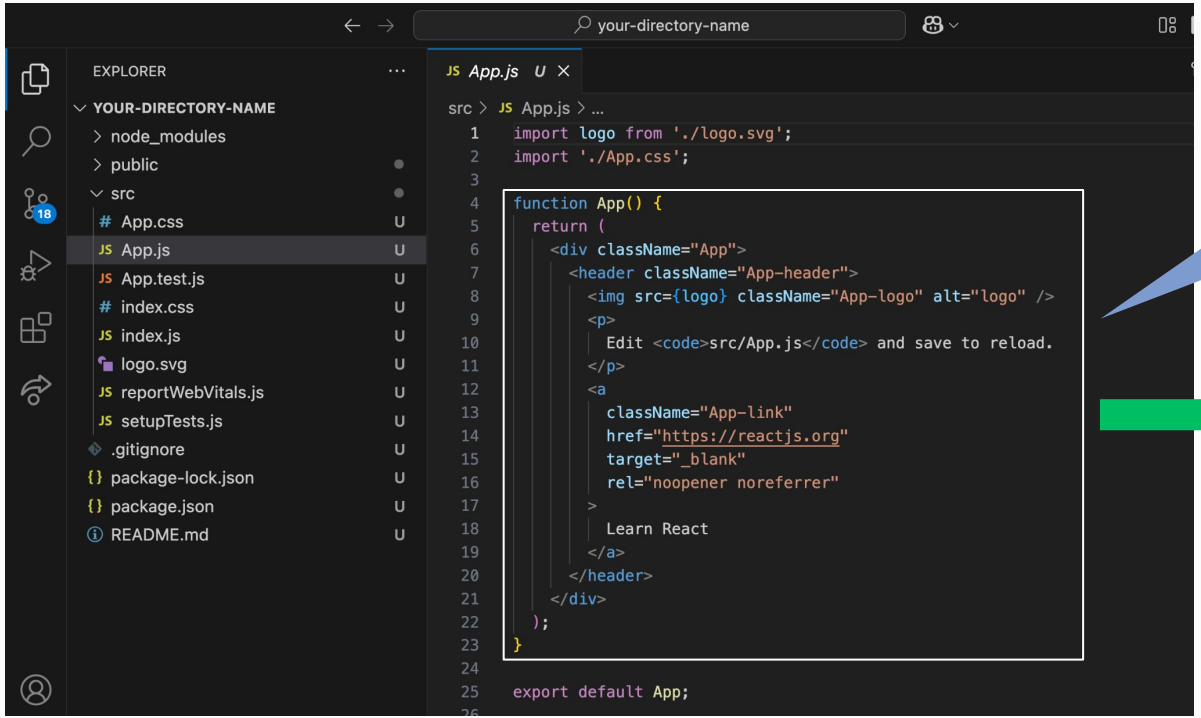




Just like the math functions, React functional components execute logic and return UI elements

Creating Functional Components

The default code inside *App.js* is actually a functional component:-

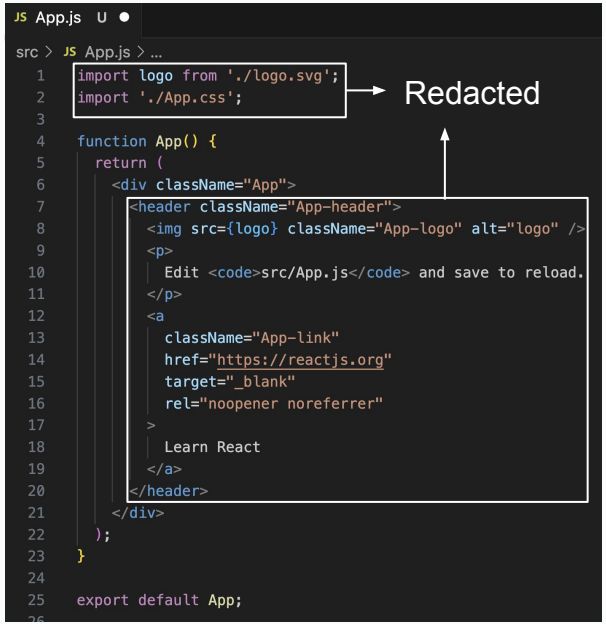


We need to clean up a bit to include our custom code.

Functional component

Cleaning up Template Code

We would like to create our own functional components so remove template code and start including our own code instead of react template provided by



Let's redact our template code

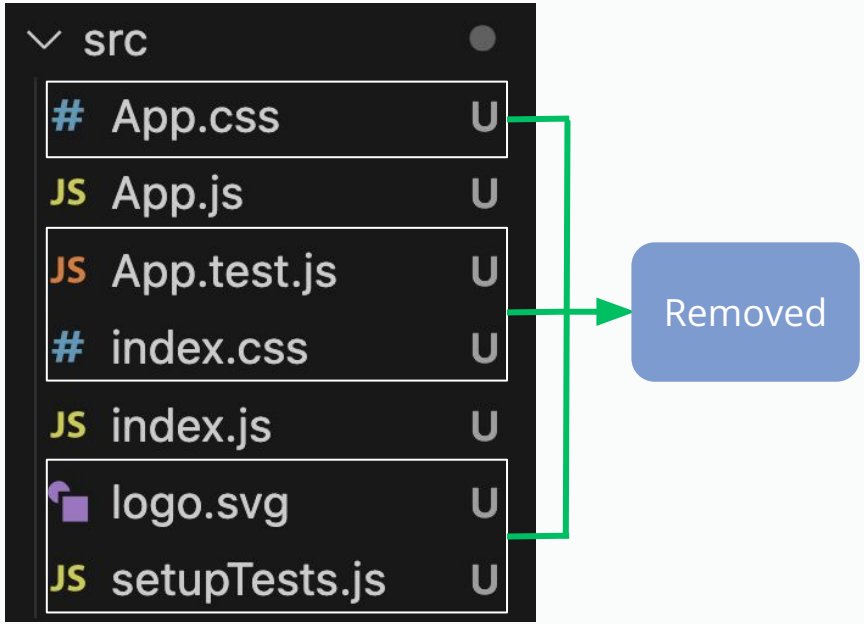
And add *Hello, World!*

```
function App() {
  return (
    <div>
      Hello, World!
    </div>
  );
}

export default App;
```

Hello, World!

Cleaning Up template Code

Let’s also delete unnecessary files from the src folder:-



Also remove index.css from index.js file as we have deleted this file

src > JS index.js > ...

```
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5
6  const root = ReactDOM.createRoot(document.getElementById('root'));
7  root.render(
8    <React.StrictMode>
9      <App />
10   </React.StrictMode>
11 );
```

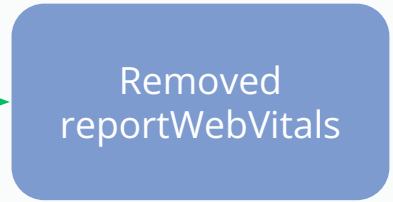
index.js file

Cleaning Up Template Code

Newton School of Technology

Inside *index.js reportWebVitals.js* file was throwing error, so we removed it from import statement as well as at the bottom of the file where it was being invoked:-

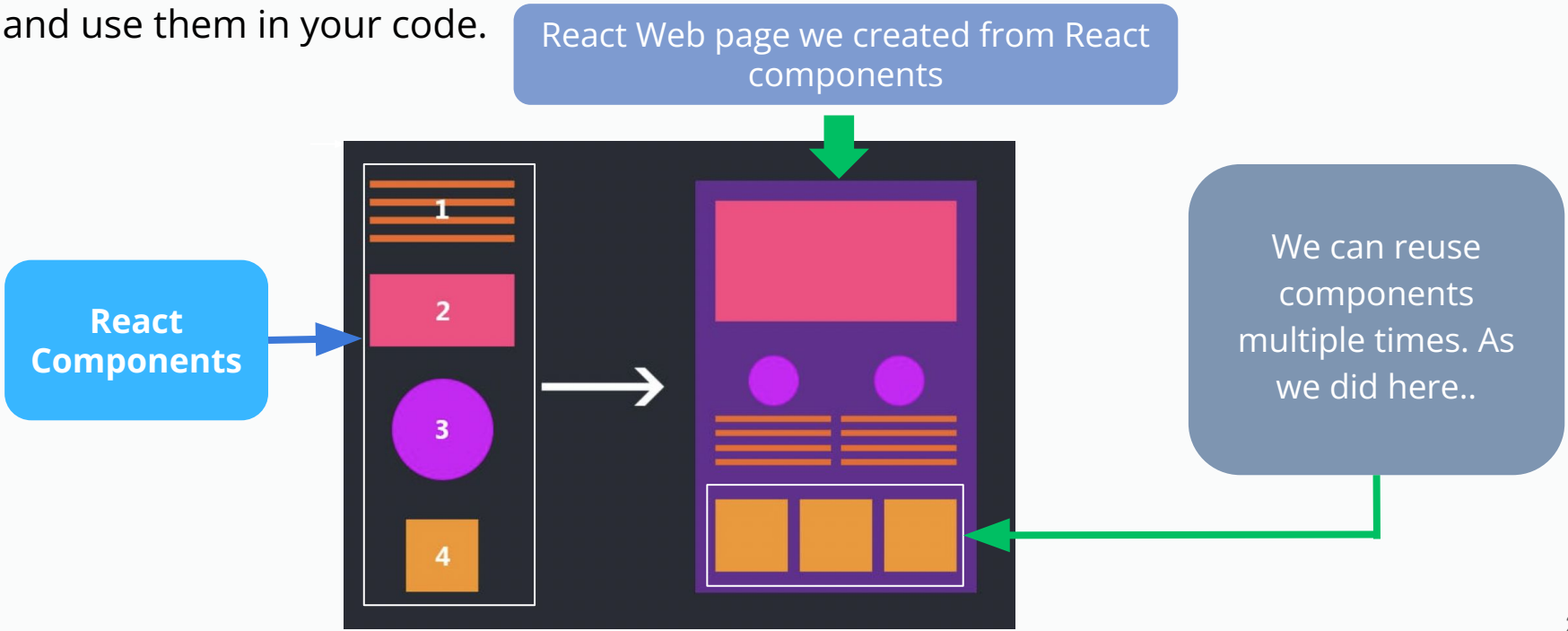
```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```



Component based structure in React

Newton School of Technology

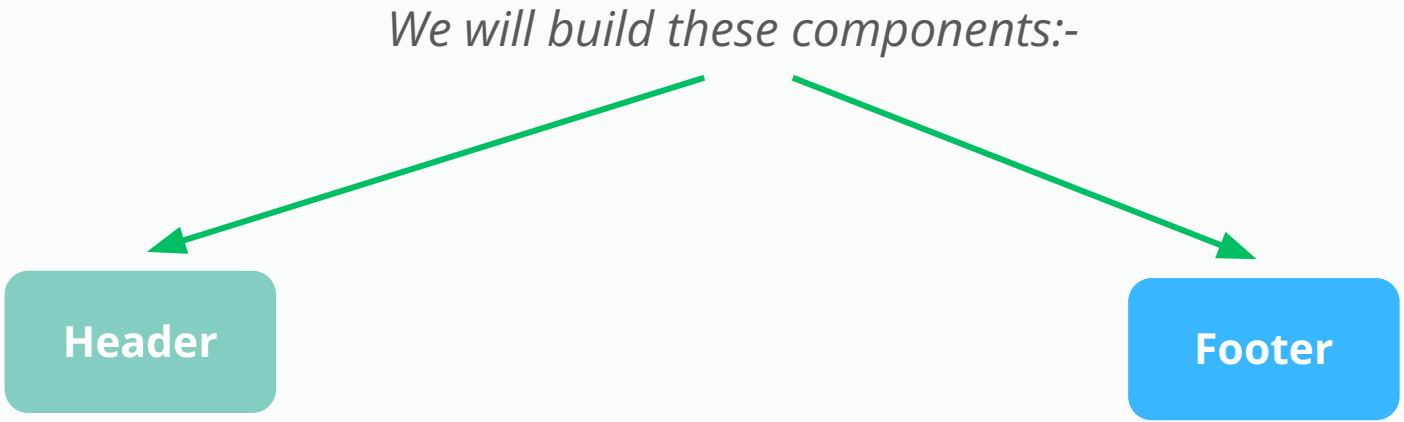
React follows a component based structure, where you built the components once and use them in your code.



Let’s create some new components

Newton School of Technology

Just like App.js which is the root component, we can build other components. For now we will build functional components.



Creating Component: Header.jsx

Newton School of Technology

For now we will create components in the same file i.e. App.js:-

```
const Header = () => {
  return <h1>Welcome to My React App</h1>;
};
```

Header Component

Here we have created a simple header component with a simple message.

Including Header in App.js

Newton School of Technology

We can simply use Header component previously written in App.js file and use it like this:-

```
4
5   const App = () => {
6     return (
7       <div>
8         <Header />
9         <p>This is the main content of the app.</p>
10      </div>
11    );
12  };
13
14  export default App;
```

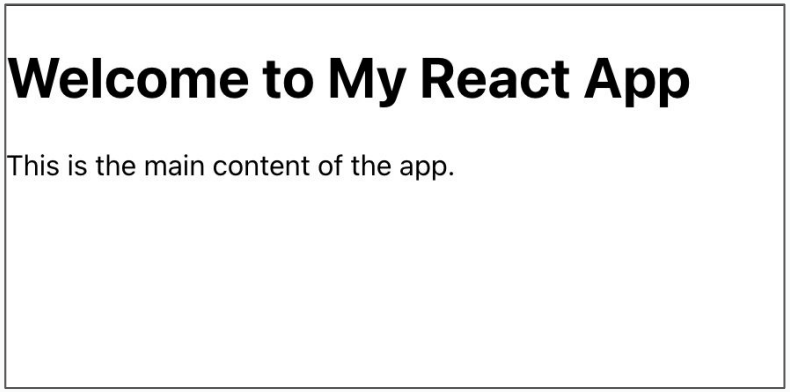
We can include our components using custom JSX tags, similar to HTML elements.

Let's see how it appears in the browser

Newton School of Technology

Let’s have a look at side browser:-

```
4
5   const App = () => {
6     return (
7       <div>
8         <Header />
9         <p>This is the main content of the app.</p>
10      </div>
11    );
12  };
13
14  export default App;
```



Creating Footer Component

Newton School of Technology

Similarly we can create Footer component and use it in inside App like this:-

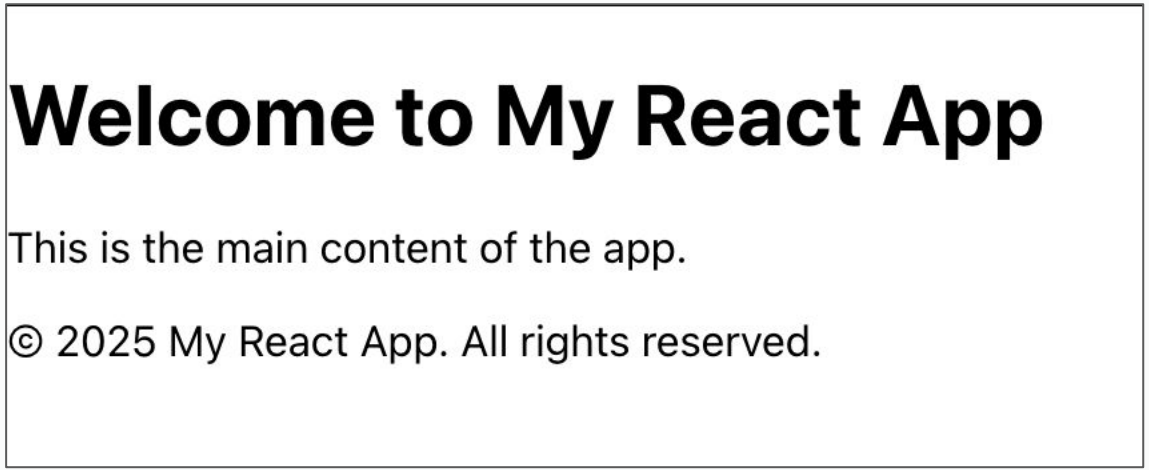
```
const Footer = () => {
  return (
    <p>
        © 2025 My React App.
        All rights reserved.
    </p>
  );
};
```

```
6   const App = () => {
7     return (
8       <div>
9         <Header />
10        <p>This is the main content of the app.</p>
11        <Footer />
12      </div>
13    );
14  };
15
16  export default App;
```

Let’s have a look of it in the browser

Newton School of Technology

Let’s open the browser and see how it looks:-

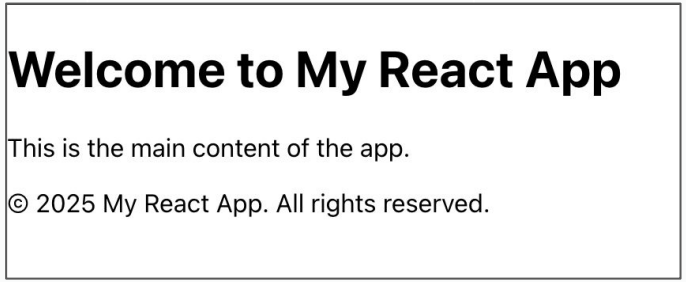


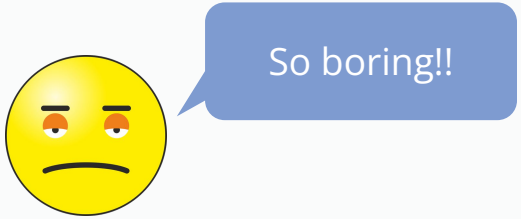
Similarly we can create as many components needed and use them throughout the project

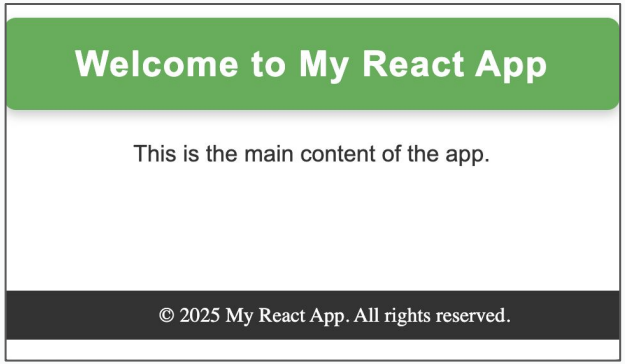
Styling our React Components

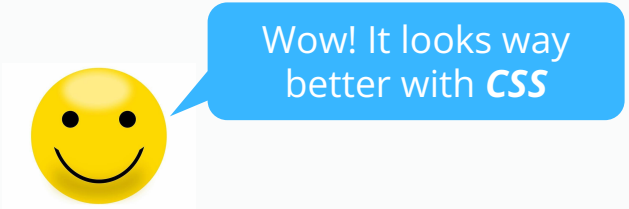
Newton School of Technology

We only have introduced how to write plain JSX, but it looks a bit off. To make it more appealing we can style it using CSS.









How to Style our Component - Inline?

Newton School of Technology

To write inline styles we use style attribute and write our style inside two curly braces:-

```
3
4   const Header = () => {
5     return (
6       <h1 style={{  }}>
7       Welcome to My React App</h1>
8     );
9   };
10
11  export default Header;
```

Style tag is used to include our styles and unlike html, we include styles inside double curly braces {{ }}

Why two curly braces {{...}}

Styling Component: { } vs {{ }}

Use single curly braces when you are using javascript variable to feed styles and double when you are writing styles directly.

```
const Header = () => {
    const headerStyle = {
        backgroundColor: "#4CAF50",
        color: "white",
        textAlign: "center",
        padding: "20px",
        fontSize: "28px",
        fontWeight: "bold",
        borderRadius: "10px"
    }
    return (
        <h1 style={headerStyle}>
        Welcome to My React App
        </h1>
    );
};
```

Just like any JSX variable, we enclose headerStyle in curly braces {}

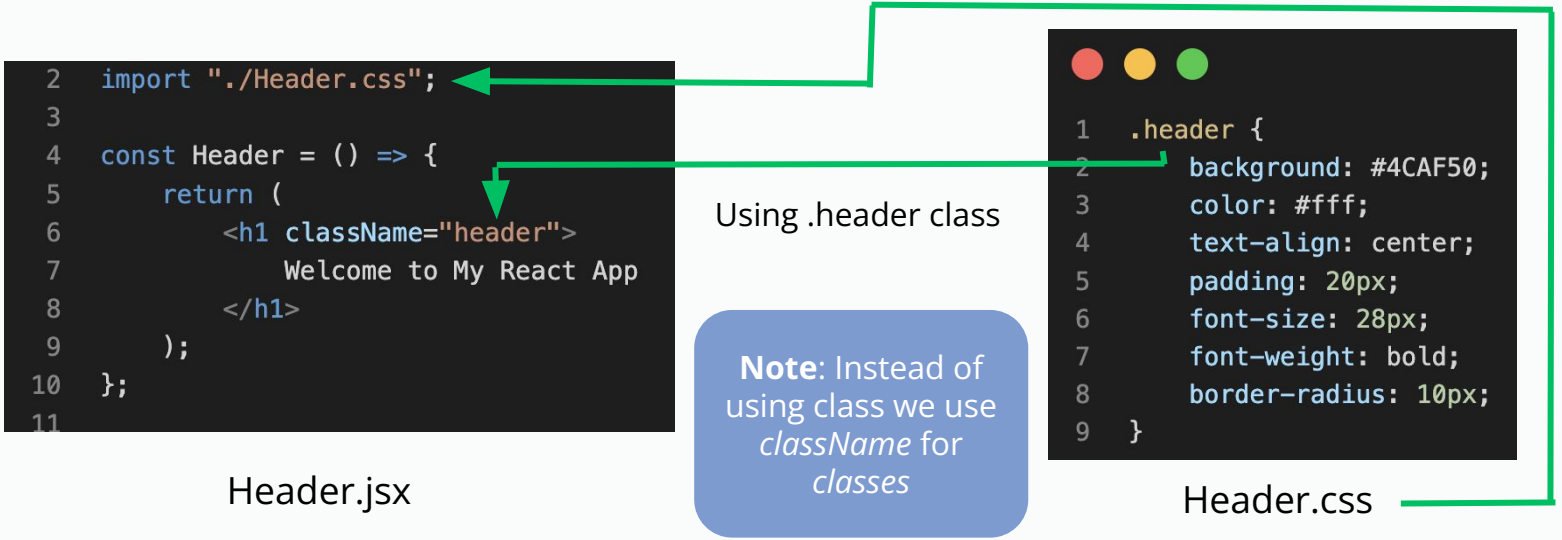
```
const Header = () => {
    return (
        <h1 style={{
            background: "#4CAF50",
            color: "#fff",
            textAlign: "center",
            padding: "20px",
            fontSize: "28px",
            fontWeight: "bold",
            borderRadius: "10px",
        }}>
            Welcome to My React App
        </h1>
    );
};

export default Header;
```

We first create a style object {} to store our styles. Then, in JSX, we wrap it inside {} again to apply it, just like using any JavaScript variable.

How to Style our Component - External?

Newton School of Technology

Using inline styles everywhere can clutter the code, so we often use an external CSS file. We define styles with class names, import the file, and apply the classes in components using className.



Note: Instead of using class we use *className* for *classes*

Have you observed Camel Casing?

Newton School of Technology

In our component which is JSX, style properties use **camelCase** instead of kebab-case.

```
const headerStyle = {
    backgroundColor: "#4CAF50",
    color: "white",
    textAlign: "center",
    padding: "20px",
    fontSize: "28px",
    fontWeight: "bold",
    borderRadius: "10px"
}
```

uses

camelCase	kebab-case
backgroundColor	background-color
textAlign	text-align
fontSize	font-size
borderRadius	border-radius

Don’t worry, We will learn JSX in next lecture.

Kebab-Style works outside Components

Newton School of Technology

However you can use kebab-style outside the component files, like .css files.

```css
.header {
    background: #4CAF50;
    color: #fff;
    text-align: center;
    padding: 20px;
    font-size: 28px;
    font-weight: bold;
    border-radius: 10px;
}
```

Kebab-Style
text-align
font-size
font-weight
border-radius

Why it is called camelCase?

Newton School of Technology

It's called **camelCase** because the capital letters in the middle create "humps" like a camel! 🐫

Look closely... see any humps? 👀

Were you able to guess?

😂

In Class Questions

References

1. **React Docs:** https://react.dev
2. **MDN React Guide**: Comprehensive and beginner-friendly documentation for React https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started
3. **Websites and Tutorials**:
   a. 📌 w3schools: https://www.w3schools.com/REACT/DEFAULT.ASP
   b. 📌codecademy: https://www.codecademy.com/learn/react-101
4. **Other Useful Resources**
   a. 📌 React GitHub Repository: https://github.com/facebook/react
   b. 📌 React Patterns & Best Practices: https://reactpatterns.com/

Newton School of Technology

Thanks for watching!