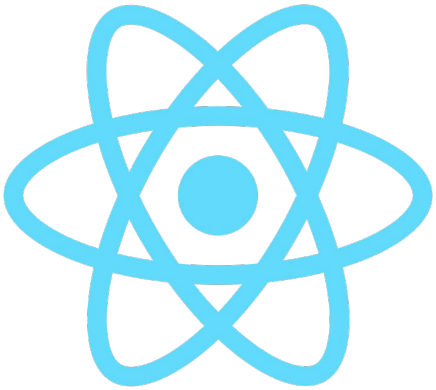
Introduction to React

Relevel
by Unacademy

Topics Covered

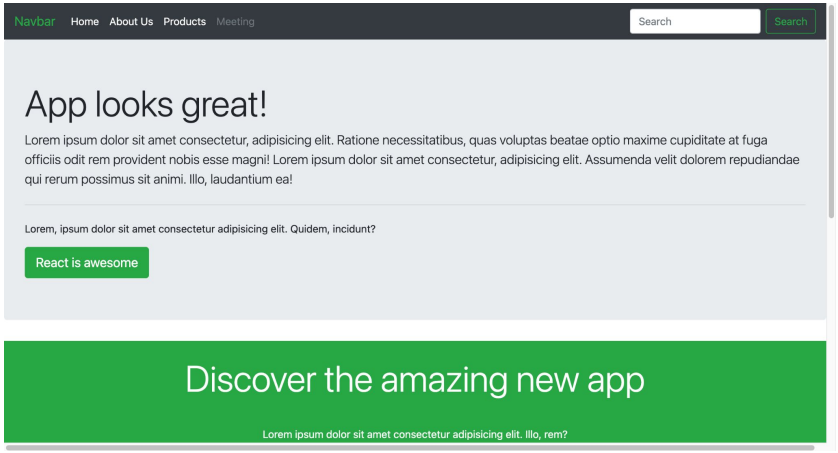
- What is react and myths
- ReactJS environment setup
- JSX in depth
- Function and Class components
- Practice Questions



App Introduction

We will be creating the following application(s):

- We will create our project first by using HTML only which includes different sections like NavBar, App section, Text section, Card section and Footer
- Then we will convert the same project into the React using the different components -
  - AppSection
  - HeroSection
  - Card
  - CardSection
  - Footer
  - Navbar



What is React

In traditional web application programming, for even a small change in the webpage, the whole page is reloaded. This makes the web pages slower than they should be.

How ReactJS solves this problem -
React only updates what's necessary

What is React?

- React or ReactJS is a JavaScript library.
- React is not a framework. (unlike Angular, the difference b/w framework and library can be explained here).
- React is an open-source project created by Facebook.
- React is used to build user interfaces (UI) on the front end.

React Myths

- React is a framework - This is the common misconception; react is not a fully-fledged framework, but it's just a library
- React is complicated to set up - Noooo! Earlier, it used to be complicated, but now we have a single command create-react-app which can be used for the setup.

Why is React so popular?

1. React is easy to maintain because react components can be reused
2. React is Fast and Dynamic; we don't need to refresh the entire page to update the specific section as it can be extracted to the separate component
3. React can be used for both Desktop web and mobile app

ReactJS environment setup

1. Use an online code playground. Try CodeSandbox.
    - how to use it - Navigate to the link; this will create the React project from scratch. In the center window, start making the changes in App.js. The changes will immediately reflect in the right window.

2. “Create React App” via npm.

Step1 - Go-to nodejs Download page - https://nodejs.org/en/download/ and download the LTS version for your operating system

Step2 - Install the downloaded file, this will install the node and npm in the system

ReactJS environment setup

Step3 - Open the command prompt and check node -v, and npm -v to check if packages are downloaded correctly

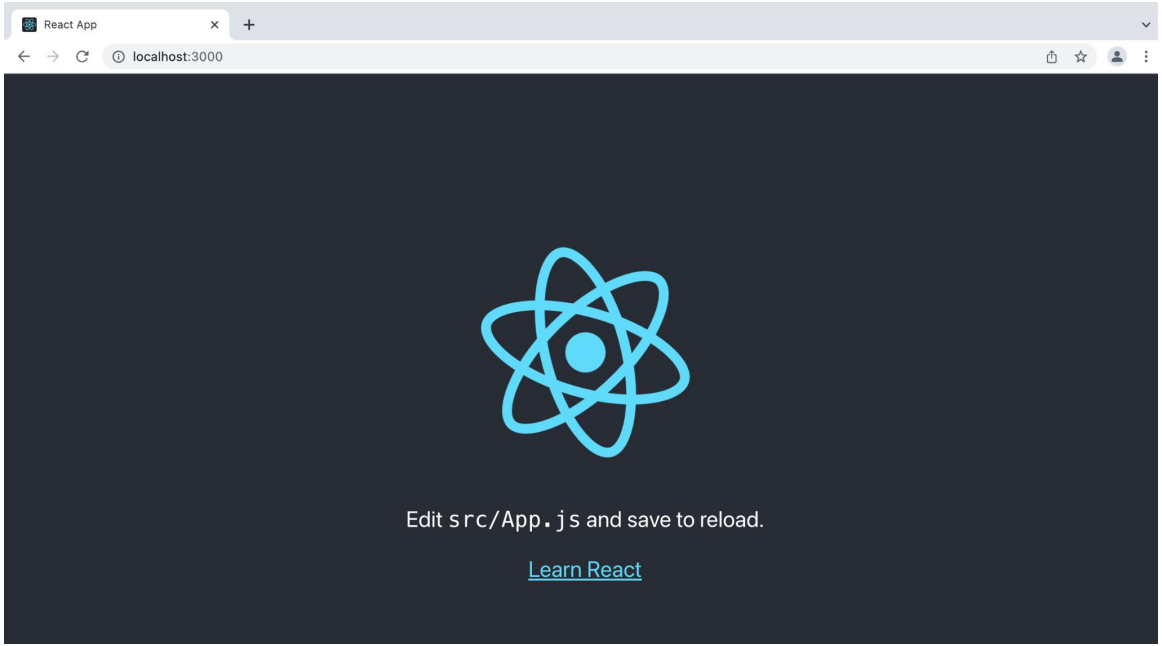
Step4 - run the command - `npm install -g create-react-app,` this will install the create-react-app command used for creating the react projects

Step5 - navigate to the folder where you want to create the react project and use the command `create-react-app relevel`

Step6 - navigate to the relevel project you created and use the command - `npm start`

ReactJS environment setup

Server will start at 3000 port and default page will show

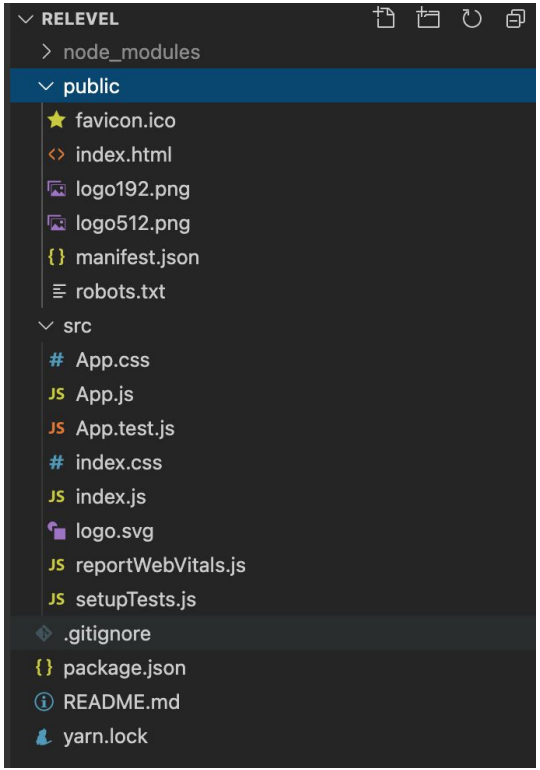


ReactJS environment setup

If you look into the project structure, you'll see a /public and /src directory, along with the regular node_modules, .gitignore, README.md, and package.json.

In /public, our important file is index.html, which is very similar to the static index.html file we made earlier — just a root div. This time, no libraries or scripts are being loaded in. The /src directory will contain all our React code.

To see how the environment automatically compiles and updates your React code, find the line that looks like this in /src/App.js:



Get Started with the code changes

Go ahead and delete all the files out of the /src directory. Just keep index.js.

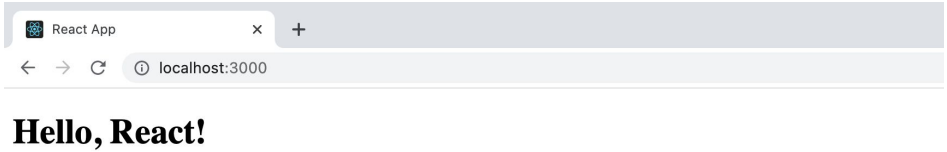
Now in index.js, we're importing React, ReactDOM and add a React component — in its simplest form — is a plain-old JavaScript function like this -

```
import React from "react";
import ReactDOM from "react-dom";

const App = () => {
  return (
    <div>
      <h1>Hello, React!</h1>
    </div>
  );
};

ReactDOM.render(<App />, document.getElementById("root"));
```

Get started with the code changes

Note how we wrote what looks like HTML in the returned output of the App function above. This is neither HTML nor JavaScript and it is not even React. This is **JSX**. It's an **extension** to JavaScript that allows us to write function calls in an HTML-like syntax.

Save this and check the browser, it will show something like this -



What is JSX

JSX stands for JavaScript XML

Consider this variable declaration -

```
const element = <h1>Hello, world!</h1>;
```

This syntax is neither a string nor HTML, It is called JSX

JSX allows us to write HTML elements in JavaScript and place them in the DOM without any createElement().
JSX converts HTML tags into react elements

What is JSX

You are not required to use JSX, but JSX makes it easier to write React applications.

Here are two examples. The first uses JSX and the second does not:

1. `const myelement = <h1> Using JSX here!</h1>;`
2. `const myelement = React.createElement('h1', {}, 'Not using JSX here!');`

JSX is basically a **compromise**. Instead of writing React components using the React.createElement syntax, we use a syntax very similar to HTML and then use a compiler to translate it into React.createElement calls.
JSX is an extension of the JavaScript language based on ES6, and is translated into regular JavaScript at runtime.

Function vs Class based components

Let's create our App component again. Before, we just created the plain javascript function, now we will define a class extends Component as it includes the render method. Now inside the render method return the same JSX to render. Sample code -

```
import React from "react";
import ReactDOM from "react-dom";

class App extends Component {
  render() {
    return (
      <div>
        <h1>Hello, React!</h1>
      </div>
    );
  }
}

ReactDOM.render(<App />, document.getElementById("root"));
```

Check the output in the browser, it should be the same

Function vs Class based components

Functional Component	Class Component
A functional component is just a plain JavaScript function that accepts props as an argument and returns a React element.	A class component requires you to extend from React. Component and create a render function which returns a React element.
There is no render method used in functional components.	It must have the render() method returning HTML

Convert HTML to React

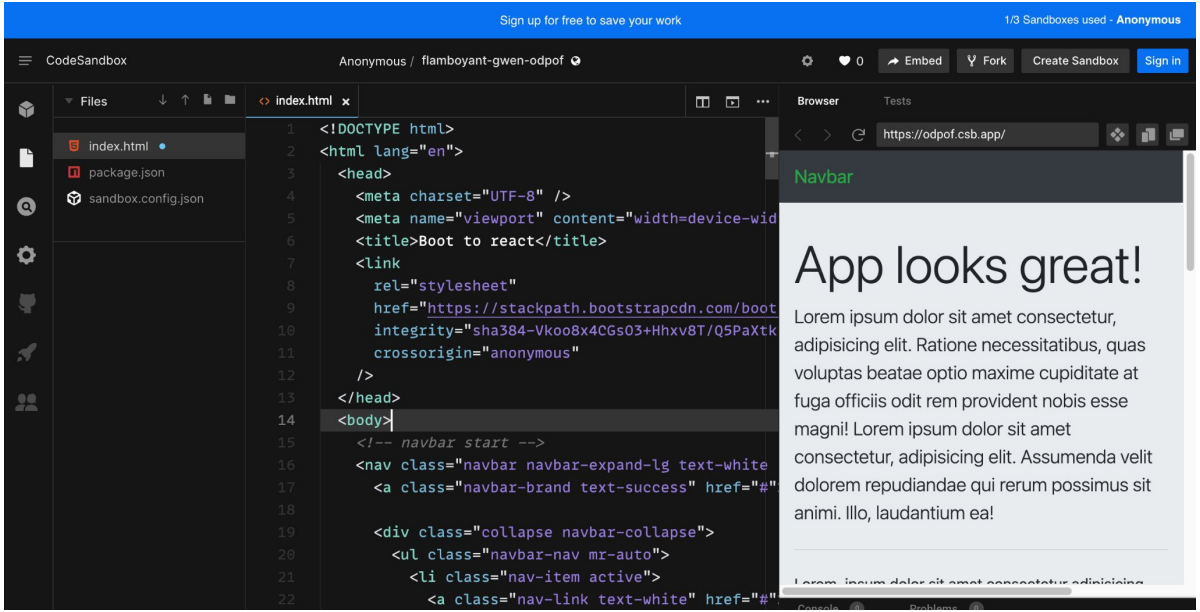
We will create the project shown above using HTML only first

Check the full code here - https://codesandbox.io/s/flamboyant-gwen-odpof?file=/index.html

We will use bootstrap here for the styling.
Explain each section created using simple HTML and bootstrap - <nav> section, App description section, App discover section includes play store and app store links, Cards section and Footer

CodeSand box will show the code and rendered section in the right -

Convert HTML to React



Convert HTML to React

Let's convert the project to React -

Use this link to create the blank project - https://codesandbox.io/s/react-new
In react, a component represents a part of the user interface. Now let’s create different components for each section

Convert HTML to React

How to Create and Export a Component?

Our application has five components:

- AppSection
- HeroSection
- Card
- CardSection
- Footer
- Navbar

And these components are imported into the App.js file.
Components can also contain some other Components like App.js is also a component that contains all the above components.
App.js component is further imported into the Index.js file.

Convert HTML to React

To import the component in some other file, we need to first export it in its own JS file.

Given below is the example of the **AppSection Component**:

```
1  import React from "react";
2
3  const AppSection = () => (
4    <div className="bg-success text-center">
5      <p className="display-4 text-white p-4">Discover the amazing new app</p>
6      <p className="text-white">
7        Lorem ipsum dolor sit amet consectetur adipisicing elit. Illo, rem?
8      </p>
9      <div className="row  p-4">
10       <div className="col-6 text-right">
11         <button className="btn btn-warning btn-lg">Play Store</button>
12       </div>
13       <button className="btn btn-warning btn-lg">App Store</button>
14       <div className="col-6 text-center"></div>
15     </div>
16   </div>
17 );
18
19 export default AppSection;
```

We use the syntax : **export default <component name>** to export the component for it to be further imported in some other JS file.
Here, we've used **export default AppSection** (Line no. 19) to export the AppSection Component.

Convert HTML to React

How to Import and Use a Component?

Syntax to import a component : **import <component name> from <file path>**

Given below is the **App.js** file:

```
1  import React from "react";
2
3  import NavBar from "./Component/Navbar";
4  import HeroSection from "./Component/HeroSection";
5  import AppSection from "./Component/AppSection";
6  import CardSection from "./Component/CardSection";
7  import Footer from "./Component/Footer";
8
9  const App = () => {
10   return (
11     <div>
12       <NavBar />
13       <HeroSection />
14
15       <CardSection />
16       <AppSection />
17       <Footer />
18     </div>
19   );
20 };
21
22 export default App;
```

All the components are imported using the aforementioned syntax (line no. 3-7).

And all the imported components are displayed using the syntax: **<Component Name/>** (Line no. 12-17)

Convert HTML to React

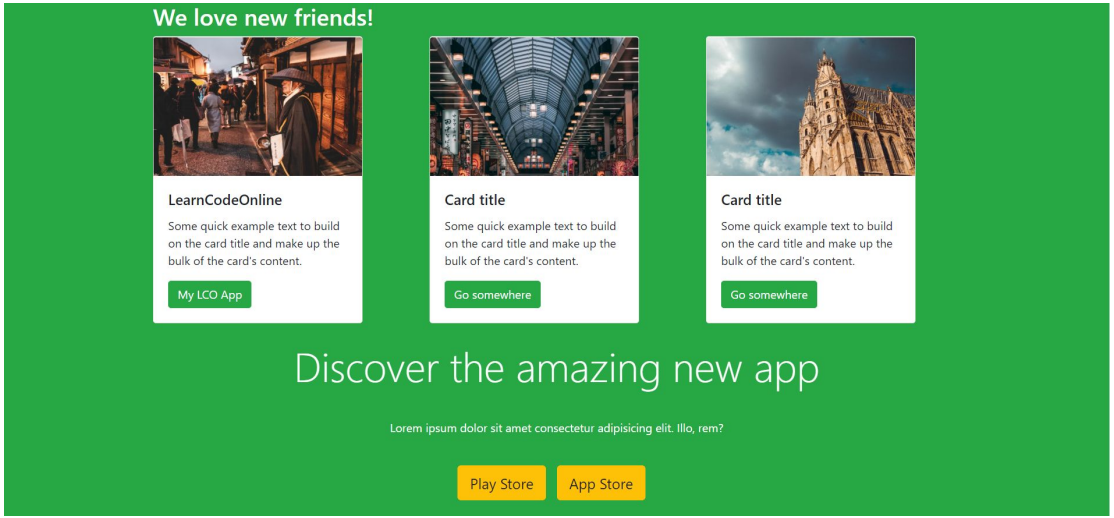
Given below is the Structure of the WebPage after importing and displaying all the components

NavBar and HeroSection

Convert HTML to React

Given below is the Structure of the WebPage after importing and displaying all the components

CardSection and AppSection -



Convert HTML to React

Given below is the Structure of the WebPage after importing and displaying all the components

Footer -

Code link -

Plain HTML project - https://codesandbox.io/s/priceless-fire-k4vj3?file=/index.html
- This sandbox project includes the code for creating the simple HTML project created above

React project - https://codesandbox.io/s/goofy-feynman-t786d?file=/src/App.js
- This sandbox project includes the code for converting the simple HTML project into the React project as explained above

Practice / HW Ques

- Create a blog website in react.

Upcoming Topics

- Intro to components and props
- Stateful vs stateless components
- Reusable components
- Merge components to create one page
- Conditional rendering of components

Thank-you