From 65 – react started

React is a JavaScript library for building user interfaces.

React is used to build single-page applications.

React allows us to create reusable UI components.

[Start learning React now ❯](https://www.w3schools.com/REACT/react_intro.asp)

Day 65 to day 68 – total theory

To get an overview of what React is, you can write React code directly in HTML.

But in order to use React in production, you need npm and [Node.js](https://nodejs.org/) installed.

Start by including three scripts, the first two let us write React code in our JavaScripts, and the third, Babel, allows us to write JSX syntax and ES6 in older browsers.

Creating react code using plain html and js:

<!DOCTYPE html>

<html>

<head>

<script src="[https://unpkg.com/react@18/umd/react.development.js](https://unpkg.com/react@17/umd/react.development.js)" crossorigin></script>

<script src="[https://unpkg.com/react-dom@18/umd/react-dom.development.js](https://unpkg.com/react-dom@17/umd/react-dom.development.js)" crossorigin></script>

<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>

</head>

<body>

<div id="mydiv"></div>

<script type="text/babel">

function Hello() {

return <h1>Hello World!</h1>;

}

const container = document.getElementById('mydiv');

const root = ReactDOM.createRoot(container);

root.render(<Hello />)

</script>

</body>

</html>Try it Yourself

The above code refer - Executed in Reactintro folder

**Day 69 – creating react app**

And folder structure

Setting up a React Environment

If you have npx and Node.js installed, you can create a React application by using create-react-app.

If you've previously installed create-react-app globally, it is recommended that you uninstall the package to ensure npx always uses the latest version of create-react-app.

To uninstall, run this command: npm uninstall -g create-react-app.

Run this command to create a React application named my-react-app:

npx create-react-app my-react-app

The create-react-app will set up everything you need to run a React application.

Run the React Application

Now you are ready to run your first *real* React application!

Run this command to move to the my-react-app directory:

cd my-react-app

Run this command to run the React application my-react-app:

npm start

A new browser window will pop up with your newly created React App! If not, open your browser and type localhost:3000 in the address bar.

**React.createElement(): creating element without using jsx**

By using simple javascript – document.getelementbyid and createElement

createElement lets you create a React element. It serves as an alternative to writing [JSX.](https://react.dev/learn/writing-markup-with-jsx)

Syntax:

Const element = createElement(type, props, ...children)

**App.jsx** – replace with below code

import { createElement } from 'react';

function App({ name }) {

  return createElement(

    'h1',

    { className: 'greeting' },

    'Hello'

  );

}

export default App;

the above code is equivalent to

return (

<h1 classname:”greeting” >Hello</h1>

)

Second argument- props can be null , or any attribute

**Creating nested elements/tags:**

import { createElement } from 'react';

// function App2({ name }) {

//   return createElement(

//     'h1',

//     { className: 'greeting' },

//     'heading1 ',

//     createElement('i', null, name),

//     '. Welcome!'

//   );

// }

// export default  App2

// . For comparison, here is the same example rewritten with JSX:

function App2() {

    return (

      <h1 className="greeting">

        Hello <i>name</i>. Welcome!

      </h1>

    );

  }

  export default App2

**Note**: Instead of old syntax – ReactDOM.render(<App/>,document.getElementID(‘root’) , use ReactDOM.createroot(document.getElementByID).render(<App/>)

**New syntax:**

ReactDOM.createRoot(document.getElementById('root')).render(<App/>)

//creating and rendering  in same component- main.jsx/index.js only

const title1=React.createElement('h1',null,'heading1');

const title2=React.createElement('h1',null,'heading2')

//more than one element to render- use array like this

ReactDOM.createRoot(document.getElementById('root')).render([title1,title2])