

React JS

Source: Chai Aur Code

Instructor: Hitesh Choudhary

Notes: Araiz Naqvi

For my connections and me :)

Part 4



Creating a Miniature React Library Functionality

All React Apps will include an **HTML file** with the **script files**. Hence we tried to re-create one such as follows:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>Custom React App</title>
</head>
<body>
  <div class="root"></div>
  <script src="./customReact.js"></script>
</body>
</html>
```

In the above **index.html**, we included a **div** with class **root** and a **script** file called **customReact.js**. Moving forward let's create the **customReact.js**.

customReact.js

Start by **query selecting** the **root class**.

Next, let's define a React element like React would do with its **React.createElement method** and define what exactly the element would do.

Let's start with an **'a' element** with properties such as **href** and **target**. It shall have **children** that hold its **inner HTML**.

As follows:

```
let reactElement = {  
  type: 'a',  
  props: {  
    href: 'https://google.com',  
    target: '_blank'  
  },  
  children: 'Click me to visit Google.'  
}  
let Root = document.querySelector('.root');
```

Next, we include the **customRender** function that will render our root **'class'** content on the web page.

```
function customRender(reactElement, container)
{

    let domElement =
document.createElement(reactElement.type);
    domElement.innerHTML =
reactElement.children;

    for(let prop in reactElement.props){
        if(prop === 'children') continue;
        domElement.setAttribute(prop,
reactElement.props[prop]);
    }
    container.appendChild(domElement);
}
```

However, the following can't be redone in a React App since the `reactElement` and its properties are something we know about not React. Hence, we use an inbuilt method called `createElement` as follows:

```

import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App.jsx'

let reactElement = React.createElement(
  'a',
  {href:"https://github.com", target: '_blank'},
  'Click me to visit Github'
)

function MyFunc() {
  return(
    <h1>Araiz Here</h1>
  )
}

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    /* reactElement */
    <App />
    < MyFunc />
  </React.StrictMode>,
)

```

The above is how a new React element can be created.

Extras

When creating a function in main.jsx or anywhere else we can also use it traditionally as a function(). However, it is recommended to use it in the `<function />` manner.

Thanks