React JS

Learning ReactJS

Praveen Nair

What is React?

ReactJS is a popular JavaScript library

It is used for building user interfaces for web applications.

It was developed and maintained by Facebook

It is great for SPA which load once and later everything happens thru JavaScript without reloading the page.

React was release on 29th May 2013.

Similar products available in market are Angular and Veujs.

Installing React 18.0

Download and Install Node Js

Node –version

Npm – version (node package manager)

React Developer Tools – chrome extension

Npx create-react-app appname (if nodem error try **npm install -g npm**)

Npm start --- to run the app

Vscode extention vscode es7 react/reduc/react-native/js snippets

React JSX (JavaScriptXML)

- JSX allows us to write HTML elements in JavaScript and place them in the DOM
- export default function App() {
   return (<h1>Hello World</h1>);
  }
  ………………………………………
  Tags must be closed

JS vs React App Function

- function App(){
- let str=`<h1>Hello World</h1>`
- return str
- }
- root.innerHTML = App()
- ....
- function App(){
- return (<h1>Hello World</h1>)
- }
- root.render(<App/>)

One top Element / Fragment

-
  ```
  <div>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>
  </div>
  ```

-
  ```
  <>
    <p>Paragraph 1</p>
    <p>Paragraph 2</p>
  </>
  ```

React Components (jsx)

-
  ```
  export default function Header() {
    return <h1>Company Name</h1>;
  }
  ```

-
  ```
  export default function Content() {
    return <h1>Display content here</h1>;
  }
  ```

-
  ```
  export default function Footer() {
    return <h1>Footer Section</h1>;
  }
  ```

Component Props & Expression

- ….User.js…..

-
  ```
  export default function User(props) {
    return (
      <div>Hello {props.name}</div>
    )
  }
  ```
- .... App.js

-
  ```
  <div>
  <User name="John"/>
      </div>
  ```

If Statement with props

```
• function Result(props) {
•   const r = props.r;
•   if (r) {
•     return <Pass/>;
•   }
•   return <Fail/>;
• }
```

Ternary Operator

- ……………………………………….

  ```
  function Result(props) {
    const r = props.r;
    return (
      <>
        { r ? <Pass/> : <Fail/> }
      </>
    );
  }
  ```

Logical Operator

- const cart = ['rice', 'wheat', 'sugar’];
-
  ```
  {
  cart.length > 0 && //used instead of if
          <h2> You have {cart.length} items in your cart.</h2>
  }
  ```

Props & Expression (Numbers)

- ...Calc.js

-
  ```
  export default function Calc(props) {
    let x = props.a;
    let y = props.b;
    let z = x + y;
    return <div>Result is {z}</div>;
  ```

-
  ```
  }
  ```
- .... App.js

-
  ```
  <div>
       <Calc a={3} b={5}/>
      </div>
  ```

Props & Expression (Array)

- ...Add.js..

  ```
  export default function Add(props) {
    let result = props.arr.reduce((total, value) => {
      return total + value;
    });
  ```

- …..App.js

  ```
  <Add arr={[5,6,8]} />
  ```

Props & Expression (Object)

- ...Students.js

```
export default function Students(props) {
  return (
    <div>{props.score['John']}</div>
  )
}
```

……App.js

```
<Students score={{'John':40,'Cathy':50}}/>
```

Props & Expression (callback)

- ….Sqr.js

```
export default function Sqr(props) {
    let result = props.fn(props.a)
  return (
    <div>{result}</div>
  )
}
```

…… App.js

- <Sqr a={5} fn={sqrfunction}/>

Map function to display array

-
  ```
  function Cart() {
    const arr = ["Rice", "Wheat", "Sugar"];
    return (
        <ul>
          {arr.map((item) => (
            <li>{item}</li>
          ))}
        </ul>
    );
  }
  ```

Deploy reactapp to https://pages.github.com/

- 1. Create a new react app - ecomm-react

```
npx create-react-app ecomm-react
```

2. open ecomm-react folder in vscode

3. install gh pages

```
npm install gh-pages --save-dev
```

4. update package.json

```
    "homepage": "https://<Username>.github.io/<Repository-name>“
    "scripts":{
            "predeploy": "npm run build",
            "deploy": "gh-pages -d build"
            }
```

5. publish to github

6. npm run deploy

7. go to ecomm-react repo > settings > pages

Events in React

-
  ```
  function Customer() {
    const greet = () => {
      alert("Hello!");
    }
    return (
      <button onClick={greet}>Greet</button>
    );
  }
  ```

Events (Passing Arguments)

-
  ```
  function Customer() {
    const greet = (a) => {
      alert(a);
    }
    return (
      <button onClick={() => greet("Hi Ajay")}>Greet</button>
    );
  }
  ```

Props (default)

-
  ```
  import ReactDOM from "react-dom/client";
  ```

-
  ```
  function Student(props){
  return props.age
  }
  ```

-
  ```
  Student.defaultProps = {  //outside the component
    name:'John’,
    age:20
  }
  ```

-
  ```
  const r = ReactDOM.createRoot(document.getElementById("root"));
  r.render(
  <Student name='John' age="20" />
  );
  ```

React Styling – Inline Styling

```
const Header = () => {
  return (
    <>
      <h1 style={{backgroundColor: “blue"}}>Hello World!</h1>
      <p>Add a little style!</p>
    </>
  );
}
```

………………………………….

camelCase property name – used backgroundColor instead of background-color

Spread Operator Arrays (…)

- export default function App6() {
- const arr1 = [1,4,5];
- const arr2 = [6,8,1];
   //const newArr = arr1.concat(arr2);
- const newArr = [...arr1, ...arr2];
- console.log(newArr);
- }

Spread Operator Objects (…)

-
  ```
  export default function App6() {
    const obj1 = { name: "John" };
    const obj2 = { age: 30 };
    const student = {...obj1, ...obj2};
    console.log(student);
  }
  ……………………………………………
  export default function App6() {
    const obj1 = { name: "John",age:20 };
    const obj2 = { age: 30 };
    const student = {...obj1, ...obj2};  //age will be 30
    console.log(student);
  }
  ```

Modify Object (…)

-
  ```
  export default function App6() {
    const obj1 = { name: "John",age:20 };
    const student = {...obj1, age:30};
    console.log(student);
  }
  …………………………..
  export default function App6() {
    const obj1 = { name: "John",age:20 };
    const student = {...obj1, city:'NYC’};
    console.log(student);
  }
  ```

State Management (useState hook)

- The data is stored in the React Component's state object.
- When the state object changes, the component will re-render itself.
- export default function App6() {

```
export default function App6() {
  const [count, setCount] = useState(1); //cannot be inside loops, if or functions
  console.log(Date())  //runs everytime count changes
  return (
   <>
     <button onClick={() => setCount((prevState) => prevState - 1)}>-</button>
     <span>{count}</span>
     <button onClick={() => setCount((prevState) => prevState + 1)}>+</button>
   </>
  );
}
```

Side Effects (useEffect hook)

```
export default function App6() {
  const [runs, setRuns] = useState(0);
  const [wickets, setWickets] = useState(0);
  //   console.log("Component loaded");
  useEffect(() => {
    if (wickets > 0) console.log("Better Luck Next Time!");
  }, [wickets]);
  useEffect(() => {
    if (runs > 0) console.log("Good Job!");
  }, [runs]);
  return (
    <>
      <button onClick={() => setRuns((prevState) => prevState + 1)}>
        Runs({runs})
      </button>
      <button onClick={() => setWickets((prevState) => prevState + 1)}>
        Wickets({wickets})
      </button>
    </>
  );
}
```

State Management (objects)

-
  ```
  export default function App6() {
    const [student, setStudent] = useState({ name: "John", age: 34 });
  return (
      <>
        <p> <input   type="text“  onChange={(e) =>  setStudent((prevState) => ({...prevState, ...{ name: e.target.value }, }))}
            placeholder="Enter Name“></input> </p>
        <p> <input type="text“ onChange={(e) => setStudent((prevState) => ({ ...prevState, ...{ age: e.target.value }, }))}
            placeholder="Enter Age“ ></input></p>
        <span>
          {student.name}-{student.age}
        </span>
      </>
    );
  }
  ```

State Management (arrays-ex1)

- // React requires state to be immutable. If we want to change a state, we make a copy of it and replace the old state with the new copy. In React, it isn't allowed to change the original state array, so we can't use the push() method. We used the spread syntax (...) to unpack the elements of the existing array into a new array and added another value at the end.

```
import { useState } from "react";
export default function App6() {
 const [student, setStudent] = useState({ name: "John", age: 3 });
 const [students, setStudents] = useState([]);
 const addStudent = () => {
  setStudents((prevStudents) => [...prevStudents, student]); };
 return (
  <>
   <p> <input type="text" onChange={(e) => setStudent((prevStudent) => ({...prevStudent,name: e.target.value,}))}
     placeholder="Enter Name"></input></p>
   <p><input type="text" onChange={(e) => setStudent((prevStudent) => ({...prevStudent,age: e.target.value,}))}
     placeholder="Enter Age" ></input></p>
   <p><button onClick={addStudent}>Add Student</button></p>
   <div>
    {student.name}-{student.age}
    {students &&
     students.map((value, index) => (
      <div key={index}>
       {value.name}-{value.age}
      </div>
     ))}
   </div>
  </>
 );
}
```

State Management (delete array)

```
export default function App6() {
  const [student, setStudent] = useState({});
  const [students, setStudents] = useState([]);
  const addStudent = () => {
    setStudents((prevStudents) => [...prevStudents, student]);
  };
  const deleteStudent = (name) => {
    setStudents(students.filter((e) => e.name !== name));
  };
  return (
    <>
      <p> <input type="text" onChange={(e) => setStudent((prevStudent) => ({...prevStudent,...{ name: e.target.value },}))}
          placeholder="Enter Name" ></input></p>
      <p> <input type="text" onChange={(e) => setStudent((prevStudent) => ({...prevStudent,...{ age: e.target.value },}))}
          placeholder="Enter Age"></input></p>
      <p> <button onClick={addStudent}>Add Student</button></p>
      <div>
        {students &&
          students.map((value, index) => (
            <div key={index}>
              {value.name}-{value.age} -{" "}
              <button onClick={() => deleteStudent(value.name)}>Delete</button>
            </div>
          ))}
      </div>
    </>
  );
}
```

State Management (edit array)

-
```
export default function App6() {
  const [student, setStudent] = useState({ name: "", age: "" });
  const [students, setStudents] = useState([]);
  const addStudent = () => {
    setStudents((prevStudents) => [...prevStudents, student]);
  };
  const deleteStudent = (name) => {
    setStudents(students.filter((e) => e.name !== name));
  };
  const editStudent = (value) => {
    setStudent(value);
    deleteStudent(value.name);
  };
  return (
    <>
      <p><input type="text" value={student.name} onChange={(e) => setStudent((prevStudent) => ({...prevStudent,...{ name: e.target.value },}))}
          placeholder="Enter Name" ></input> </p>
      <p> <input type="text" value={student.age} onChange={(e) => setStudent((prevStudent) => ({...prevStudent,...{ age: e.target.value },}))}
          placeholder="Enter Age"></input></p>
      <p><button onClick={addStudent}>Add Student</button></p>
      <div>
        {students &&
          students.map((value, index) => (
            <div key={index}>
              {value.name}-{value.age} -{" "}
              <button onClick={() => deleteStudent(value.name)}>Delete</button>
              <button onClick={() => editStudent(value)}>Edit</button>
            </div>
          ))}
      </div>
    </>
  );
}
```

React Router

- npm i -D react-router-dom

-
  ```
  <Router>   //Browserrouter as router
    <p><Link to="/add">Add</Link> |
        <Link to="/view">View</Link></p>
    <Routes>
     <Route index element={<Add />} />
     <Route path="add" element={<Add />} />
     <Route path="view" element={<View />} />
    </Routes>
  </ Router >
  ```

React-icons

- npm install react-icons

```
……………………………………………………………
import React from 'react’
import { FaHome, FaBookmark } from "react-icons/fa";
export default function App14() {
  return (
    <div>
        <p><FaHome/>Home</p>
        <p><FaBookmark />Feeds</p>
        <p>Post</p>
        <p>Album</p>
    </div>
  )
}
```

Ecomm shopping cart

- Add to cart
- Increase and decrease quantity