REACT PROGRAMS

Prerequisite of React

For learning React first you have a clear understanding of HTML, CSS and JavaScript. As React is a JavaScript library and uses most of its concept so you really have to understands the major concepts of it.

HTML and CSS

JavaScript and ES6

JSX (Javascript XML) & Babel

Node+Npm

Git and CLI (Command Line Interface).

React Environment Setup:

npx create-react-app <<Application_Name>>

cd <<Application_Name>>

npm start

React JSX:

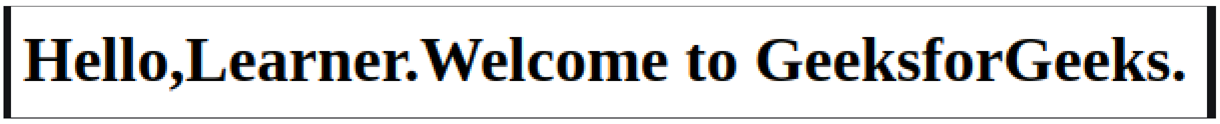
React JSX sample code:

const ele = <h1>This is sample JSX</h1>;

Example 1: This example wraps the JSX code in curly braces

```
// Filename - App.js

import React from "react";

const name = "Learner";

const element = (
    <h1>
        Hello,
        {name}.Welcome to GeeksforGeeks.
    </h1>
);

ReactDOM.render(element,
document.getElementById("root"));
```

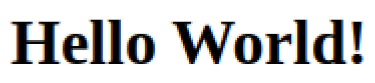
Output:


Hello,Learner.Welcome to GeeksforGeeks.

Example 2: In this example where conditional expression is embedded in JSX:

```
// Filename - App.js

import React from "react";

let i = 1;

const element = (
    <h1>{i == 1 ? "Hello World!" : "False!"} </h1>
);

ReactDOM.render(element,
document.getElementById("root"));
```

Output:


Hello World!

Comments in JSX :

```
// Filename - App.js

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

const element = (
    <div>
        <h1>Hello World !{/*This is a comment*/}</h1>
    </div>
);

ReactDOM.render(element, document.getElementById("root"));
```

Converting HTML to JSX

HTML

```
<!DOCTYPE html>
<html>

<head>
    <title>Basic Web Page</title>
</head>
<body>
    <h1>Welcome to GeeksforGeeks</h1>
    <p>A computer science portal for geeks</p>
</body>

</html>
```

The Converted JSX Code will look like:

Javascript

```
<>
  <title>Basic Web Page</title>
  <h1>Welcome to GeeksforGeeks</h1>
  <p>A computer science portal for geeks</p>
</>
```

React JS ReactDOM:

```
// Installing
npm i react-dom
```

```
// Importing
import ReactDOM from 'react-dom'
```

After installing react-dom it will appear in the dependenices in package.json file like:

```
"dependencies": {
   "react": "^18.2.0",
   "react-dom": "^18.2.0",
   "react-scripts": "5.0.1",
}
```

React Lists:

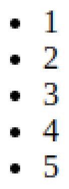
React List Examples

Example: This example implements a simple list in ReactJS.

JavaScript

```
// Filename -  index.js
import React from 'react';
import ReactDOM from 'react-dom';
const numbers = [1,2,3,4,5];
const updatedNums = numbers.map((number)=>{
   return <li>{number}</li>;
});
ReactDOM.render(
   <ul>
      {updatedNums}
   </ul>,
   document.getElementById('root')
);
```

Output:

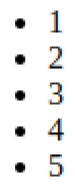


```
• 1
• 2
• 3
• 4
• 5
```

Using Keys in Lists:

```
import React from 'react';
import ReactDOM from 'react-dom';
// Component that will return an
// unordered list
function Navmenu(props)
{
   const list = props.menuitems;
const updatedList = list.map((listItems)=>{
        return(
                <li key={listItems.toString()}>
                    {listItems}
                </li>
            );
    });
  return(
        <ul>{updatedList}</ul>
    );
}
const menuItems = [1,2,3,4,5];
ReactDOM.render(
    <Navmenu menuitems = {menuItems} />,
    document.getElementById('root')
);
```

Output:



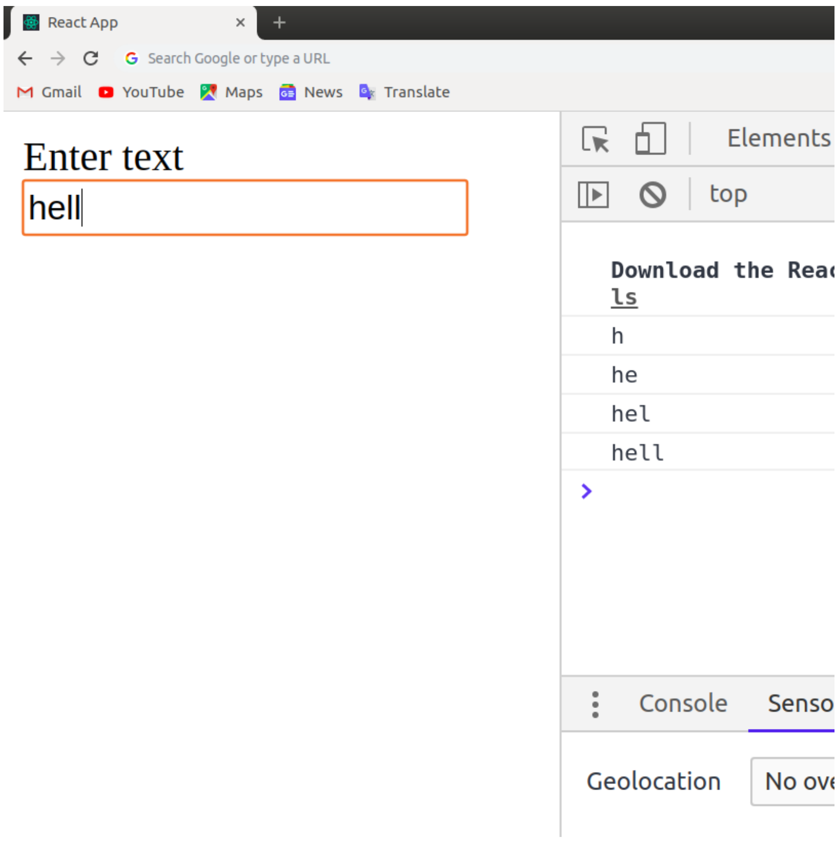
```
• 1
• 2
• 3
• 4
• 5
```

Adding Forms in React:

```
// Filename - src/index.js:
import React from 'react';
import ReactDOM from 'react-dom';
class App extends React.Component {
    onInputChange(event) {
        console.log(event.target.value);
    }
    render() {
```

```
return (
        <div>
           <form>
              <label>Enter text</label>
              <input type="text"
                 onChange={this.onInputChange}/>
           </form>
        </div>
     );
  }
}
ReactDOM.render(<App />,
        document.querySelector('#root'));
```

Output:



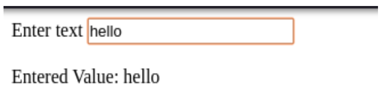
```
Enter text
hell
```

```
Download the Reac
ls
h
he
hel
hell
>
```

Handling React Forms

```
// Filename - index.js
import React from 'react';
import ReactDOM from 'react-dom';
class App extends React.Component {
   state = { inputValue: '' };
   render() {
      return (
         <div>
            <form>
               <label> Enter text </label>
               <input type="text"
                  value={this.state.inputValue}
                  onChange={(e) => this.setState(
                     { inputValue: e.target.value })} />
            </form>
            <br />
            <div>
               Entered Value: {this.state.inputValue}
            </div>
         </div>
      );
   }
}
ReactDOM.render(<App />,
            document.querySelector('#root'));
```

Output:


```
Enter text hello

Entered Value: hello
```

Submitting React Forms

```
// Filename - index.js

import React from "react";

import ReactDOM from "react-dom";

class App extends React.Component {

   state = { inputValue: "" };

   onFormSubmit = (event) => {

      event.preventDefault();

      this.setState({ inputValue: "Hello World!" });

   };

   render() {

      return (

         <div>

            <form onSubmit={this.onFormSubmit}>

               <label> Enter text </label>

               <input

                  type="text"

                  value={this.state.inputValue}

                  onChange={(e) =>

                     this.setState({

                        inputValue: e.target.value,

                     })

                  }
```

```
          />
        </form>
        <br />
        <div>
          Entered Value: {this.state.inputValue}
        </div>
      </div>
    );
  }
}
ReactDOM.render(<App />, document.querySelector("#root"));
```

Output:

```
Enter text rajkumar
Entered value: rajkumar
```

Multiple Input Fields

```
import React from "react";
// import ReactDOM from 'react-dom/client';
import "./index.css";
// import App from './App';
// import reportWebVitals from './reportWebVitals';
// Filename - index.js
import ReactDOM from "react-dom";
class App extends React.Component {
  state = { username: "", email: "" };
  onFormSubmit = (event) => {
    event.preventDefault();
    this.setState({
      username: "gfg123",
      email: "abc@gfg.org",
    });
  };
  render() {
    return (
      <div
        style={{
          margin: "auto",
          marginTop: "20px",
          textAlign: "center",
        }}
      >
        <form onSubmit={this.onFormSubmit}>
          <label> Enter username: </label>
          <input
            type="text"
            value={this.state.username}
            onChange={(e) =>
              this.setState((prev) => ({
                ...prev,
                username: e.target.value,
              }))
            }
          />
          <br />
          <br />
```

```
                    <label>Enter Email Id:</label>
                    <input
                        type="email"
                        value={this.state.email}
                        onChange={(e) =>
                            this.setState((prev) => ({
                                ...prev,
                                email: e.target.value,
                            }))
                        }
                    ></input>
                    <br />
                    <br />
                    <input type="submit" value={"Submit"} />
                </form>
                <br />
                <div>
                    Entered Value: {this.state.username}
                </div>
            </div>
        );
    }
}
const root = ReactDOM.createRoot(
    document.getElementById("root")
);
```

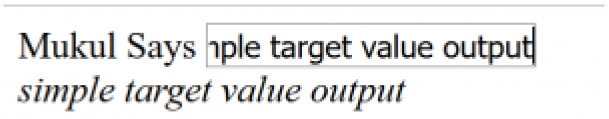
```
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

ReactJS Refs:

```
// Filename - App.js
// without refs
class App extends React.Component {
  constructor() {
    super();
    this.state = { sayings: "" };
  }
  update(e) {
    this.setState({ sayings: e.target.value });
  }
  render() {
    return (
      <div>
        Mukul Says{" "}
        <input
          type="text"
          onChange={this.update.bind(this)}
        />
        <br />
        <em>{this.state.sayings}</em>
```

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

Output:

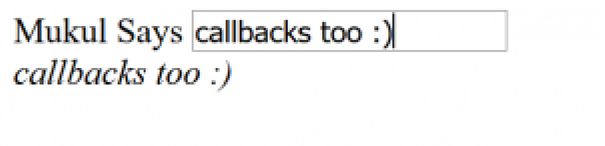


```
Mukul Says nple target value output
simple target value output
```

Example 2: In this example, we directly define callback function within ref.

```
// Filename - App.js
// callback used inside ref
class App extends React.Component {
    constructor() {
        super();
        this.state = { sayings: "" };
    }
    update(e) {
        this.setState({ sayings: this.a.value });
    }
    render() {
        return (
            <div>
                Mukul Says{" "}
                <input
                    type="text"
                    ref={(call_back) => {
                        this.a = call_back;
                    }}
                onChange={this.update.bind(this)}
              />
              <br />
              <em>{this.state.sayings}</em>
            </div>
          );
        }
      }
ReactDOM.render(<App />, document.getElementById("root"));
```

Output:



```
Mukul Says callbacks too :)
callbacks too :)
```

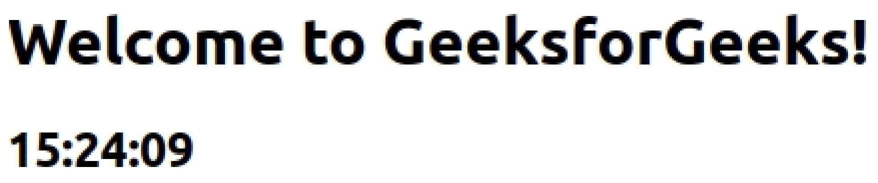
ReactJS Rendering Elements

```
<div id="root"></div>
```

```
import React from 'react';
import ReactDOM from 'react-dom';
function App() {
        const myElement = (
                <div>
                        <h1>Welcome to GeeksforGeeks!</h1>
                        <h2>{new Date().toLocaleTimeString()}</h2>
                </div>
        );
        ReactDOM.render(
                myElement,
                document.getElementById("root")
        );
}
setInterval(App, 1000);
```

```
export default App;
```

Output



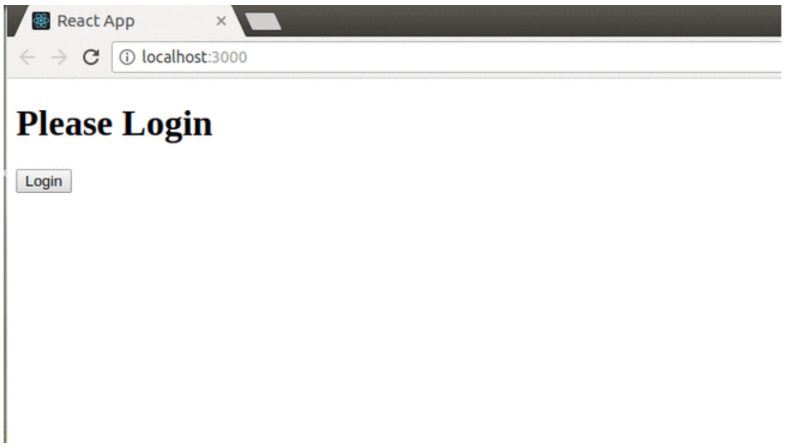
```
Welcome to GeeksforGeeks!
15:24:09
```

React Conditional Rendering:

if-else Statement Example:

```
import React from "react";
import ReactDOM from "react-dom";
// Message Component
// Message Component
function Message(props)
{
    if (props.isLoggedIn)
        return <h1>Welcome User</h1>;
    else
        return <h1>Please Login</h1>;
}
// Login Component
function Login(props) {
    return <button onClick={props.clickFunc}>Login</button>;
}
// Logout Component
function Logout(props) {
    return <button onClick={props.clickFunc}>Logout</button>;
}
// Parent Homepage Component
class Homepage extends React.Component {
  constructor(props) {
    super(props);
    this.state = { isLoggedIn: false };
    this.ifLoginClicked = this.ifLoginClicked.bind(this);
    this.ifLogoutClicked = this.ifLogoutClicked.bind(this);
  }
  ifLoginClicked() {
    this.setState({ isLoggedIn: true });
  }
  ifLogoutClicked() {
    this.setState({ isLoggedIn: false });
  }
  render() {
    return (
      <div>
        <Message isLoggedIn={this.state.isLoggedIn} />
        {this.state.isLoggedIn ? (
          <Logout clickFunc={this.ifLogoutClicked} />
        ) : (
          <Login clickFunc={this.ifLoginClicked} />
        )}
      </div>
    );
  }
}

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

Output:


React App

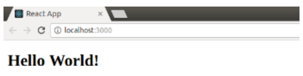
localhost:3000

Please Login

logical && operator Example:

```
import React from 'react';

import ReactDOM from 'react-dom';

// Example Component

function Example(){

        const counter = 5;

        return(<div>

                        {

                                (counter==5) &&

                                <h1>Hello World!</h1>

                        }

                </div>

                );

}

ReactDOM.render(

        <Example />,

        document.getElementById('root')

);
```

Output


React App

localhost:3000

Hello World!

Ternary Operator Example:

```
import React from "react";
import ReactDOM from "react-dom";
// Message Component
function Message(props) {
        return props.isLoggedIn ? <h1>Welcome User</h1> : <h1>Please Login</h1>;
}
// Login Component
function Login(props) {
        return <button onClick={props.clickFunc}>Login</button>;
}
// Logout Component
function Logout(props) {
        return <button onClick={props.clickFunc}>Logout</button>;
}
// Parent Homepage Component
class Homepage extends React.Component {
        constructor(props) {
                super(props);

                this.state = { isLoggedIn: false };

                this.ifLoginClicked = this.ifLoginClicked.bind(this);
                this.ifLogoutClicked = this.ifLogoutClicked.bind(this);
        }
        ifLoginClicked() {
                this.setState({ isLoggedIn: true });
        }
        ifLogoutClicked() {
                this.setState({ isLoggedIn: false });
        }
        render() {
                return (
                        <div>
                                <Message isLoggedIn={this.state.isLoggedIn} />
                                {this.state.isLoggedIn ? (
                                        <Logout clickFunc={this.ifLogoutClicked} />
                                ) : (
                                        <Login clickFunc={this.ifLoginClicked} />
                                )}
                        </div>
                );
        }
}
ReactDOM.render(<Homepage />, document.getElementById("root"));
```

ReactDOM.render():

```
// Filename - src/index.js:
import React from "react";
import ReactDOM from "react-dom";
// This is a functional component
const Welcome = () => {
  return <h1>Hello World!</h1>;
};
ReactDOM.render(
  <Welcome />,
  document.getElementById("root")
);
```

Props:

```
function Message(props) {
  return <h2>{props.text}</h2>;
}
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<Message text="Hello, world!" />);
```

Components in Components

```
// Filename - src/index.js:
import React from "react";
import ReactDOM from "react-dom";
const Greet = () => {
   return <h1>Hello Geek</h1>
}
// This is a functional component
const Welcome = () => {
    return <Greet />;
};
ReactDOM.render(
    <Welcome />,
    document.getElementById("root")
);
```

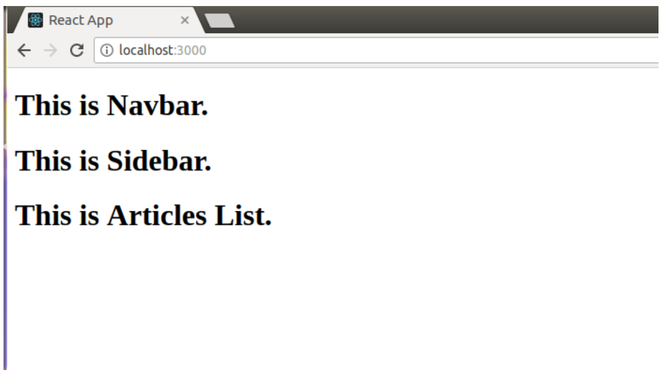
Composing Components:

Filename- App.js:

```
import React from 'react';
import ReactDOM from 'react-dom';
// Navbar Component
const Navbar=()=>
{
            return <h1>This is Navbar.< /h1>
}
```

```
// Sidebar Component
const Sidebar=()=> {
        return <h1>This is Sidebar.</h1>
}
// Article list Component
const ArticleList=()=>
{
        return <h1>This is Articles List.</h1>
}

// App Component
const App=()=>
{
        return(
                        <div>
                                <Navbar />
                                <Sidebar />
                                <ArticleList />
                        </div>
                );
}
ReactDOM.render(
        <App />,
        document.getElementById("root")
);
```

Output:

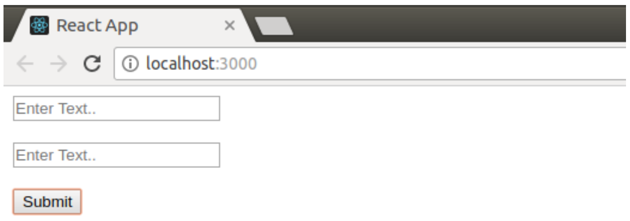


```
This is Navbar.
This is Sidebar.
This is Articles List.
```

Decomposing Components:

```
import React from 'react';

import ReactDOM from 'react-dom';

const Form=()=>

{

        return (

                <div>

                        <input type = "text" placeholder = "Enter Text.." />

                        <br />

                        <br />

                        <input type = "text" placeholder = "Enter Text.." />

                        <br />

                        <br />

                        <button type = "submit">Submit</button>

                </div>

        );

}

ReactDOM.render(

        <Form />,

        document.getElementById("root")

)
```

Output



Example 1: This example demonstrates the creation of functional components.

```
// Filename - index.js
import React from "react";
import ReactDOM from "react-dom";
import Demo from "./App";
const root = ReactDOM.createRoot(
    document.getElementById("root")
);
root.render(
    <React.StrictMode>
        <Demo />
    </React.StrictMode>
);
//Filename - App.js

import React from 'react';
import ReactDOM from 'react-dom';

const Demo=()=>{return <h1>Welcome to GeeksforGeeks</h1>};
export default Demo;
```

Output:

Welcome to Geeksforgeeks

Example 2: This example demonstrates the use of useState() hook in functional component.

```
/ Filename - index.js

import React from "react";
import ReactDOM from "react-dom";
import Example from "./App";
```

```
const root = ReactDOM.createRoot(
    document.getElementById("root")
);
root.render(
    <React.StrictMode>
        <Example />
    </React.StrictMode>
);
/ Filename - App.js

import React, { useState } from "react";

const Example = () => {
    const [change, setChange] = useState(true);
    return (
        <div>
            <button onClick={() => setChange(!change)}>
                Click Here!
            </button>
            {change ? (
                <h1>Welcome to GeeksforGeeks</h1>
            ) : (
                <h1>A Computer Science Portal for Geeks</h1>
            )}
        </div>
    );
};
export default Example;
```

Example 3: This example demonstrates the use of useEffect() hook.

```
// Filename - index.js

import React from "react";
import ReactDOM from "react-dom";
import Example from "./App";

const root = ReactDOM.createRoot(
    document.getElementById("root")
);
root.render(
    <React.StrictMode>
        <Example />
    </React.StrictMode>
);
// Filename - App.js
```

```
import React, { useEffect } from "react";

const Example = () => {
    useEffect(() => {
        console.log("Mounting...");
    });
    return <h1>Geeks....!</h1>;
};
export default Example;
```

Output:

Geeks....!

Example 4: This example demonstrates the use of props.

```
/ Filename - index.js

import React from "react";
import ReactDOM from "react-dom";
import PropsExample from "./App";

const root = ReactDOM.createRoot(
    document.getElementById("root")
);
root.render(
    <React.StrictMode>
        <PropsExample />
    </React.StrictMode>
);
// Filename - App.js

import React, { useState } from "react";

const Example = (props) => {
    return <h1>{props.data}</h1>;
};
const PropsExample = () => {
    // const [change, setChange] = useState(true);
    const [change, setChange] = useState(false);
    return (
        <div>
            <button onClick={() => setChange(!change)}>
                Click Here!
            </button>
            {change ? (
                <Example data="Welcome to GeeksforGeeks" />
            ) : (
                <Example data="A Computer Science Portal for Geeks" />
            )}
        </div>
    );
};
export default PropsExample;
```

constructor():

```
// Filename - src/index.js:

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

class Test extends React.Component {
    constructor(props) {
        super(props);
        this.state = { hello: "World!" };
    }

    render() {
        return (
            <div>
                <h1>
                    GeeksForGeeks.org, Hello
                    {this.state.hello}
                </h1>
            </div>
        );
    }
}

const root = ReactDOM.createRoot(
    document.getElementById("root")
);
root.render(<Test />);
```

static getDerivedStateFromProps:

```
// Filename - src/index.js:

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

class Test extends React.Component {
    constructor(props) {
        super(props);
        this.state = { hello: "World!" };
    }
    static getDerivedStateFromProps(props, state) {
        return { hello: props.greet };
    }


    render() {
        return (
            <div>
                <h1>
                    GeeksForGeeks.org, Hello
                    {this.state.hello}
                </h1>
            </div>
        );
    }
}

const root = ReactDOM.createRoot(
    document.getElementById("root")
);
root.render(<Test greet="Geeks!"/>);
```

render() Example:

```
// Filename - src/index.js:

import React from "react";

import ReactDOM from "react-dom/client";

class Test extends React.Component {

    render() {

        return (

            <div>

                <h1>

                    GeeksforGeeks

                </h1>

            </div>

        );

    }
}
const root = ReactDOM.createRoot(
    document.getElementById("root")
);
root.render(<Test />);
```

componentDidMount() Example:

```
// Filename - src/index.js:
import React from "react";
import ReactDOM from "react-dom/client";
class Test extends React.Component {
    constructor(props) {
        super(props);
        this.state = { hello: "World!" };
    }
    componentDidMount() {
        this.setState({hello:"Geeks!"})
    }
    render() {
        return (
            <div>
                <h1>
                    GeeksForGeeks.org, Hello
                    {this.state.hello}
                </h1>
            </div>
        );
    }
}
```

```
const root = ReactDOM.createRoot(
    document.getElementById("root")
);
root.render(<Test />);
```

getDerivedStateFromProps:

```
static getDerivedStateFromProps(props, state) {
        if(props.name !== state.name){
            //Change in props
            return{
                name: props.name
            };
        }
        return null; // No change to state
}
```

setState()

```
this.setState((prevState, props) => ({
      counter: prevState.count + props.diff
}));
```

setState Example:

```
// Filename - index.js
import React from "react";
import ReactDOM from "react-dom/client";
class App extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            count: 0,
        };
    }
    increment = () => {
        this.setState((prevState) => ({
            count: prevState.count + 1,
        }));
    };
    decrement = () => {
        this.setState((prevState) => ({
            count: prevState.count - 1,
        }));
    };
    render() {
        return (
            <div>
                <h1>
                    The current count is :{" "}
                    {this.state.count}
                </h1>
                <button onClick={this.increment}>
                    Increase
                </button>
                <button onClick={this.decrement}>
                    Decrease
                </button>
            </div>
        );
    }
}
const root = ReactDOM.createRoot(
    document.getElementById("root")
);
root.render(
    <React.StrictMode>
        <App />
    </React.StrictMode>
```

```
);
```

componentWillUnmount() Example:

```
import React from "react";

class ComponentOne extends React.Component {

    // Defining the componentWillUnmount method

    componentWillUnmount() {

        alert("The component is going to be
unmounted");

    }

    render() {

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

    }

}

class App extends React.Component {

    state = { display: true };

    delete = () => {

        this.setState({ display: false });

    };

    render() {

        let comp;

        if (this.state.display) {

            comp = <ComponentOne />;

        }

        return (

            <div>

                {comp}

                <button onClick={this.delete}>

                    Delete the component

                </button>
            </div>
        );
    }
}

export default App;
```

.

Implementing the Component Lifecycle methods

Let us now see one final example to finish the article while revising what's discussed above.

First, create a react app and edit your **index.js** file from the src folder.

```
// Filename - src/index.js:
import React from "react";
import ReactDOM from "react-dom/client";
class Test extends React.Component {
    constructor(props) {
        super(props);
        this.state = { hello: "World!" };
    }
    componentDidMount() {
        console.log("componentDidMount()");
    }
    changeState() {
        this.setState({ hello: "Geek!" });
    }
    render() {
        return (
            <div>
                <h1>
```

```
                    GeeksForGeeks.org, Hello
                    {this.state.hello}
                </h1>
                <h2>
                    <a
                        onClick={this.changeState.bind(
                            this
                        )}
                    >
                        Press Here!
                    </a>
                </h2>
            </div>
        );
    }
    shouldComponentUpdate(nextProps, nextState) {
        console.log("shouldComponentUpdate()");
        return true;
    }
    componentDidUpdate() {
        console.log("componentDidUpdate()");
    }
}
const root = ReactDOM.createRoot(
    document.getElementById("root")
);
root.render(<Test />);
```

ReactJS Methods as Props

```
import './App.css';
import React from 'react';
```

```
// imports component
import ParentComponent from './components/ParentComponent';
function App() {
    return (
        <div className="App">
            <h1>-----------METHODS AS PROPS-------------</h1>
            <ParentComponent />

        </div>
    );
}
export default App;


import React, { Component } from 'react';
import ChildComponent from './ChildComponent';

class ParentComponent extends Component {
    constructor(props) {
        super(props);

        this.state = {
            parentName:'Parent'
        }

        this.greetParent = this.greetParent.bind(this)
    }

    greetParent() {
        alert(`Hello ${this.state.parentName}`)
    }

    render() {
        return (
            <div>
                <ChildComponent greetHandler={this.greetParent}/>
            </div>
        )
    }
}

export default ParentComponent;
import React from 'react';

function ChildComponent(props) {
    return (
        <div>
            <button onClick={() => props.greetHandler()}>
```

```
                Greet Parent
            </button>
        </div>
    )
}

export default ChildComponent;
```

Passing parameters to parents in methods as props

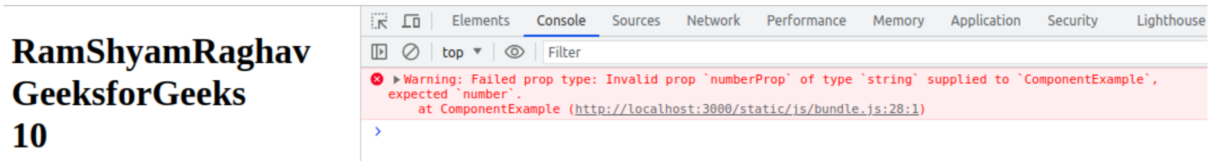
```
// App.js

import './App.css';
import React from 'react';

// imports component
import ParentComponent from './components/ParentComponent';

function App() {
    return (
        <div className="App">
            <h1>-----------METHODS AS PROPS-------------</h1>
            <ParentComponent />

        </div>
    );
}

export default App;
// ParentComponent.js

import React, { Component } from 'react';
import ChildComponent from './ChildComponent';

class ParentComponent extends Component {
    constructor(props) {
        super(props);

        this.greetParent = this.greetParent.bind(this)
    }

    greetParent(name) {
        alert(`Hello ${name}`)
    }

    render() {
        return (
            <div>
                <ChildComponent greetHandler={this.greetParent}/>
            </div>
        )
    }
}

export default ParentComponent;
/ ChildComponent.js

import React from 'react';

function ChildComponent(props) {
    return (
        <div>
            <button onClick={() => props.greetHandler("Child")}>
                Greet Parent from child
            </button>
        </div>
    )
}

export default ChildComponent;
```

Example: Write the following code in index.js file of your react application

```
import PropTypes from 'prop-types';
import React from 'react';
import ReactDOM from 'react-dom/client';

// Component
class ComponentExample extends React.Component{
    render(){
        return(
                <div>

                    {/* printing all props */}
                    <h1>
                        {this.props.arrayProp}
                        <br />

                        {this.props.stringProp}
                        <br />

                        {this.props.numberProp}
                        <br />

                        {this.props.boolProp}
                        <br />
                    </h1>
```

```
                 </div>
             );
     }
}

// Validating prop types
ComponentExample.propTypes = {
    arrayProp: PropTypes.array,
    stringProp: PropTypes.string,
    numberProp: PropTypes.number,
    boolProp: PropTypes.bool,
}

// Creating default props
ComponentExample.defaultProps = {

    arrayProp: ['Ram', 'Shyam', 'Raghav'],
    stringProp: "GeeksforGeeks",
    numberProp: "10",
    boolProp: true,
}

const root =
ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <React.StrictMode>
    <ComponentExample />
  </React.StrictMode>
);
```

Output:



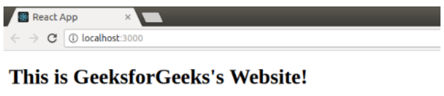
```
RamShyamRaghav
GeeksforGeeks
10
```

```
Warning: Failed prop type: Invalid prop `numberProp` of type `string` supplied to `ComponentExample`,
  expected `number`.
      at ComponentExample (http://localhost:3000/static/js/bundle.js:28:1)
```

defaultProps:

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

// Component
class ExampleClass extends React.Component {
      render() {
            return (
                  <div>
                        {/* using default prop - title */}
                        <h1>This is {this.props.title}'s Website!</h1>
                  </div>
            );
      }
}

// Creating default props for
// ExampleClass Component
ExampleClass.defaultProps = {
      title: "GeeksforGeeks"
}

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

Output:

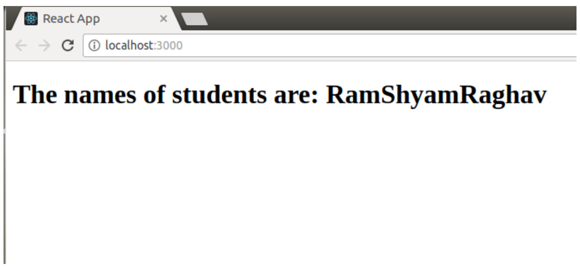


This is GeeksforGeeks's Website!

Open your react project directory and edit the **App.js** file from src folder:
src/App.js:

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

// Component
class ExampleClass extends React.Component {
      render() {
            return (
                  <div>
                        {/* accessing array prop directly */}
                        <h1>The names of students are:
{this.props.names}</h1>
                  </div>
            );
      }
}

// Passing an array as prop
ExampleClass.defaultProps = {
      names: ['Ram', 'Shyam', 'Raghav']
}

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

Output:



React App

localhost:3000

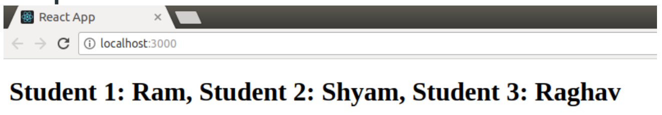
The names of students are: RamShyamRaghav

Open your react project directory and edit the **App.js** file from src folder:

src/App.js:

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

// Component
class ExampleClass extends React.Component {
      render() {
            return (
                  <div>
                        {/* iterating over array using map() */}
                        <h1>{this.props.names.map(
                              function namesIterator(item, i) {
                                    return (
                                          "Student " + (i + 1) + ": " +
                                          item +
                                          ((i != 2) ? ', ' : '\n')
                                    )
                              }

                        )}</h1>
                  </div>
            );
      }
}

// Passing an array as prop
ExampleClass.defaultProps = {
      names: ['Ram', 'Shyam', 'Raghav']
}

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

Output:



```
Student 1: Ram, Student 2: Shyam, Student 3: Raghav
```

Creating State Object

```
import React from 'react';

class MyComponent extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            brand: 'Ford', // Example property in the state
        };
    }

    render() {
        return (
            <div>
                <h1>My Car</h1>
                {/* Other component content */}
            </div>
        );
    }
}

export default MyComponent;
```

Example

This example demonstrates the use of React JS state creating a simple counter application.

```
// Filename - index.js



import React from "react";

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

```
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0,
    };
  }

  increment = () => {
    this.setState((prevState) => ({
      count: prevState.count + 1,
    }));
  };

  decrement = () => {
    this.setState((prevState) => ({
      count: prevState.count - 1,
    }));
  };

  render() {
    return (
      <div>
        <h1>
```

```
                The current count is :{" "}
                {this.state.count}
            </h1>
            <button onClick={this.increment}>
                Increase
            </button>
            <button onClick={this.decrement}>
                Decrease
            </button>
        </div>
    );
  }
}


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

Relation between Components and normal functions in JavaScript

```
// simple component

class FakeComponent extends React.component {

    render() {

        return <div>Hello World!</div>

    }

}

// simple javascript function

const FakeFunction = () => console.log('Hello World!');
```

Let us see an the example where we can properly understand the difference between state props

```
// Filename - index.js



import React, { Component } from "react"

import ReactDOM from 'react-dom';

import './index.css';



const Fruit = (props) => {



    return (

        <div className="fruit">

            <h1>List of Fruits</h1>

            <p>Name: {props.fruits.name}</p>

            <p>Color: {props.fruits.color}</p>

        </div>
```

```
  )
}

class Car extends Component {
  constructor() {
    super()
    this.state = {
      car: 'Ferrari'
    }
  }

  changeMessage() {
    this.setState({
      car: 'Jaguar'
    })
  }

  render() {
    return (
      <div className="App">
        <h1>{this.state.car}</h1>
        <button onClick={() => this.changeMessage()}>
          Change
        </button>
```

```
            </div>


        )
    }
}



function App() {


    const fruits =
    {
        name: "Mango",
        color: "Yellow"
    }


    return (
        <div className="App">
            <Fruit fruits={fruits} />
            <hr></hr>
            <Car />
        </div>
    );
}
```

```
const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>

);
```

Implementing State in React Components

```
import React, { Component } from "react";



class App extends React.Component {

    constructor() {

        super();

        this.state = {

            count: 0,

        };

        this.increase = this.increase.bind(this);

    }



    increase() {

        this.setState({ count: this.state.count + 1 });

    }



    render() {
        return (
            <div style={{ margin: "50px" }}>
                <h1>Welcome to Geeks for Geeks </h1>
                <h3>Counter App using Class Component : </h3>
                <h2> {this.state.count}</h2>
                <button onClick={this.increase}> Add</button>
            </div>
        );
    }
}

export default App;
```

Implement state using react hooks Example:

```
import React, { useState } from "react";


const App = () => {
    const [count, setCount] = useState(0);


    const increase = () => {
        setCount(count + 1);
    }


    return (
        <div style={{ margin: '50px' }}>
```

```
            <h1>Welcome to Geeks for Geeks </h1>

            <h3>Counter App using Functional Component : </h3>

            <h2>{count}</h2>

            <button onClick={increase}>Add</button>

        </div>

   )

}

export default App;
```

Using Hooks in React

```
// Filename - index.js



import React, { useState } from "react";

import ReactDOM from "react-dom/client";

function App() {

    const [click, setClick] = useState(0);



    // Using array destructuring here

    // to assign initial value 0

    // to click and a reference to the function

    // that updates click to setClick

    return (

        <div>

            <p>You clicked {click} times</p>

            <button onClick={() => setClick(click + 1)}>
```

```
                Click me
            </button>
        </div>
    );
}

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

React useState Hook

```
// Filename - App.js

import React, { useState } from 'react';

function App() {
    const click = useState('GeeksForGeeks');
    return (
        <h1>Welcome to {click}</h1>
    );
}
```

export default App;

Example 1: Updating React useState Hook State

```
// Filename - App.js


import React, { useState } from 'react';


function App() {
    const [click, setClick] = useState(0);
    // using array destructuring here
    // to assign initial value 0
    // to click and a reference to the function
    // that updates click to setClick
    return (
        <div>
            <p>You clicked {click} times</p>


            <button onClick={() => setClick(click + 1)}>
                Click me
            </button>
        </div>
    );
}
export default App;
```

Example 1: React useState Hook Arrays

```
// Filename - App.js


import React, { useState } from 'react';


function App() {
    const [click, setClick] = useState([]);
    const addNumber = () => {
        setClick([
            ...click,
            {
                id: click.length,
                value: Math.random() * 10
            }
        ]);
    };

    return (
        <div>
            <ul>
                {click.map(item => (
                    <li key={item.id}>{item.value}</li>
                ))}
            </ul>
            <button onClick={addNumber}>
                Click me
            </button>
        </div>
    );
}
export default App;
```

Example 2: React useState Hook Object

```
// Filename - App.js

import React, { useState } from 'react';

function App() {
```

```
const [data, setData] = useState({
    username: '',
    password: ''
});
const [form, setForm] = useState({
    username: '',
    password: ''
});
const [submit, submitted] = useState(false);

const printValues = e => {
    e.preventDefault();
    setForm({
        username: data.username,
        password: data.password
    });
    submitted(true);
};

const updateField = e => {
    setData({
        ...data,
        [e.target.name]: e.target.value
    });
};

return (
    <div>
        <form onSubmit={printValues}>
            <label>
                Username:
```

```
          <input
            value={data.username}
            name="username"
            onChange={updateField}
          />
        </label>
        <br />
        <label>
          Password:
          <input
            value={data.password}
            name="password"
            type="password"
            onChange={updateField}
          />
        </label>
        <br />
        <button>Submit</button>
      </form>

      <p>{submit ? form.username : null}</p>

      <p>{submit ? form.password : null}</p>
    </div>
  );
}
export default App;
```

React UseEffect Hook Example:

```
// File name - HookCounterOne.js
// useEffect is defined here

import { useState, useEffect } from "react";
```

```
function HookCounterOne() {
    const [count, setCount] = useState(0);

    useEffect(() => {
        document.title = `You clicked ${count} times`;
    }, [count]);

    return (
        <div>
            <button onClick={() => setCount((prevCount) => prevCount +
1)}>
                Click {count} times{" "}
            </button>
        </div>
    );
}
export default HookCounterOne;
// Filename - App.js
// Importing and using HookCounterOne

import React from "react";
import "./App.css";
import HookCounterOne from "./components/HookCounterOne";

function App() {
    return (
        <div className="App">
            <HookCounterOne />
        </div>
    );
}
export default App;
```

Context in React

1. MarksContext.tsx(typescript)

```
// MarksContext.tsx(typescript) File
import * as React from "react";

export interface MarksContext {
    name: string;
    marks: number;
}
const contextmarks = (React.createContext <
MarksContext) | (null > null);
```

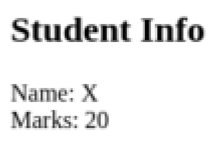
```
export const MarksContextProvider =
contextmarks.Provider;
export const MarksContextConsumer =
contextmarks.Consumer;
```

2. App.tsx(typescript)

```
import * as React from "react";
import { render } from "react-dom";
import { MarksContext, MarksContextProvider } from
"./MarksContext";
import { MarksContextConsumer } from
"./MarksContext";

const sample: MarksContext = {
    name: "X",
    marks: 20,
};

export const A = () => (
    <MarksContextProvider value={sample}>
        <B />
    </MarksContextProvider>
);

const B = () => (
    <div>
        <h2>Student Info</h2>
        <C />
    </div>
);

const C = () => (
    <MarksContextConsumer>
        {(appContext) =>
            appContext && (
                <div>
                    Name: {appContext.name} <br />
                    Marks: {appContext.marks} <br />
                </div>
            )
        }
    </MarksContextConsumer>
);
```

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

Output: .



Student Info

Name: X
Marks: 20

React Router

```
/* src/index.css */

body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
}

h2 {
    text-align: center;
    color: #333;
}

nav ul {
    display: flex;
    justify-content: center;
    list-style: none;
    padding: 0;
}

nav li {
    margin: 0 10px;
}

nav a {
    text-decoration: none;
    color: #333;
}

button {
    display: block;
    margin: 20px auto;
    padding: 10px 20px;
    background-color: #007BFF;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

button:hover {
    background-color: #0056b3;
}

// src/index.js

import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";

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

import React from "react";
import {
    BrowserRouter as Router,
    Routes,
    Route,
    Link,
    useNavigate,
    Outlet,
} from "react-router-dom";

const Home = () => {
    const navigate = useNavigate();

    return (
        <div>
            <h2>Home Page</h2>
            <button onClick={() =>
                 navigate("/contact")}>Go to Contact</button>
        </div>
    );
};

const About = () => (
    <div>
        <h2>About Page</h2>
        <nav>
            <ul>
                <li>
                    <Link to="team">Our Team</Link>
                </li>
                <li>
                    <Link to="company">Our Company</Link>
                </li>
            </ul>
        </nav>
        <Outlet />
    </div>
);

const Contact = () => <h2>Contact Page</h2>;
const Team = () => <h2>Team Page</h2>;
const Company = () => <h2>Company Page</h2>;

function App() {
    return (
        <Router>
            <nav>
                <ul>
                    <li>
                        <Link to="/">Home</Link>
                    </li>
                    <li>
                        <Link to="/about">About</Link>
                    </li>
                    <li>
                        <Link to="/contact">Contact</Link>
                    </li>
                </ul>
            </nav>
            <Routes>
                <Route path="/" element={<Home />} />
                <Route path="/about" element={<About />}>
                    <Route path="team" element={<Team />} />
                    <Route path="company" element={<Company />} />
                </Route>
                <Route path="/contact" element={<Contact />} />
            </Routes>
        </Router>
    );
}

export default App;
```

React JS Types of Routers

Example: This example demonstrates the use of MemoryRouter.

```
// Filename - App.js

import React, { Component } from "react";
import {
    MemoryRouter as Router,
    Route,
    Link,
    Switch,
} from "react-router-dom";
import Home from "./component/home";
import About from "./component/about";
import Contact from "./component/contact";
import "./App.css";

class App extends Component {
    render() {
        return (
            <Router>
                <div className="App">
                    <ul className="App-header">
                        <li>
                            <Link to="/">Home</Link>
                        </li>
                        <li>
                            <Link to="/about">
                                About Us
                            </Link>
                        </li>
                        <li>
                            <Link to="/contact">
                                Contact Us
                            </Link>
                        </li>
                    </ul>
                    <Switch>
                        <Route
                            exact
                            path="/"
                            component={Home}
                        ></Route>
                        <Route
                            exact
                            path="/about"
                            component={About}
                        ></Route>
                        <Route
                            exact
                            path="/contact"
                            component={Contact}
                        ></Route>
                    </Switch>
                </div>
            </Router>
        );
    }
}

export default App;
```

Example: This example demonstrates the use of BrowserRouter.

```
// Filename - App.js

import React, { Component } from "react";
import {
    BrowserRouter as Router,
    Route,
    Link,
    Switch,
} from "react-router-dom";

import Home from "./component/home";
import About from "./component/about";
import Contact from "./component/contact";
import "./App.css";

class App extends Component {
    render() {
        return (
            <Router>
                <div className="App">
                    <ul className="App-header">
                        <li>
                            <Link to="/">Home</Link>
                        </li>
                        <li>
                            <Link to="/about">
                                About Us
                            </Link>
                        </li>
```

```
                         <li>
                             <Link to="/contact">
                                 Contact Us
                             </Link>
                         </li>
                     </ul>
                     <Switch>
                         <Route
                             exact
                             path="/"
                             component={Home}
                         ></Route>
                         <Route
                             exact
                             path="/about"
                             component={About}
                         ></Route>
                         <Route
                             exact
                             path="/contact"
                             component={Contact}
                         ></Route>
                     </Switch>
                 </div>
             </Router>
         );
     }
}

export default App;
```

Hash Router:

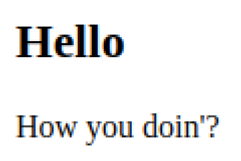
```
// Filename - App.js

import React, { Component } from "react";
import {
    HashRouter as Router,
    Route,
    Link,
    Switch,
} from "react-router-dom";
import Home from "./component/home";
import About from "./component/about";
import Contact from "./component/contact";
import "./App.css";

class App extends Component {
    render() {
        return (
            <Router>
                <div className="App">
                    <ul className="App-header">
                        <li>
                            <Link to="/">Home</Link>
                        </li>
                        <li>
                            <Link to="/about">
                                About Us
                            </Link>
                        </li>
                        <li>
                            <Link to="/contact">
                                Contact Us
                            </Link>
                        </li>
                    </ul>
                    <Switch>
                        <Route
                            exact
                            path="/"
                            component={Home}
                        ></Route>
                        <Route
                            exact
                            path="/about"
                            component={About}
                        ></Route>
                        <Route
                            exact
                            path="/contact"
                            component={Contact}
                        ></Route>
                    </Switch>
                </div>
            </Router>
        );
    }
}

export default App;
```

ReactJS Fragments

import React from "react";

```
// Simple rendering with div
class App extends React.Component {
        render() {
                return (
                        // Extraneous div element
                        <div>
                                <h2>Hello</h2>

                                <p>How you doin'?</p>
                        </div>
                );
        }
}
export default App;
```

Output:



Hello

How you doin'?

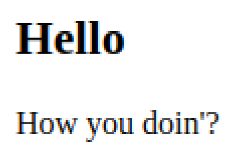
Example: Open **App.js** and replace the code with the below code.

```
import React from "react";

// Simple rendering with fragment syntax
class App extends React.Component {
    render() {
        return (
            <React.Fragment>
                <h2>Hello</h2>

                <p>How you doin'?</p>
            </React.Fragment>
        );
    }
}

export default App;
```

Example: Open **App.js** and replace the code with the below code.

```
import React from "react";

// Simple rendering with short syntax
class App extends React.Component {
    render() {
        return (
            <>
                <h2>Hello</h2>

                <p>How you doin'?</p>
            </>
        );
    }
}

export default App;
```

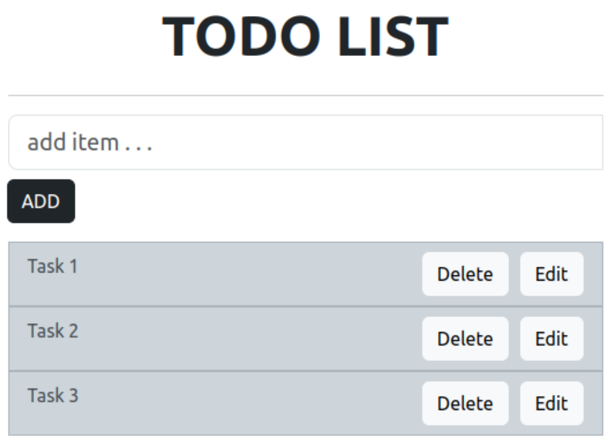
Output:



Hello

How you doin'?

Create ToDo App using ReactJS



```
npx create-react-app todo-react
cd todo-react
npm install bootstrap
npm install react-bootstrap
```

The dependencies in **package.json file** will look like:

```
"dependencies": {
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^18.2.0",
    "bootstrap": "^5.3.0",
    "react-bootstrap": "^2.7.4",
    "react-dom": "^18.2.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
}
```

Example: Write the below code in App.js file in the src directory

```
// App.js File
import React, { Component } from "react";
import "bootstrap/dist/css/bootstrap.css";
import Container from "react-bootstrap/Container";
import Row from "react-bootstrap/Row";
import Col from "react-bootstrap/Col";
import Button from "react-bootstrap/Button";
import InputGroup from "react-bootstrap/InputGroup";
import FormControl from "react-bootstrap/FormControl";
import ListGroup from "react-bootstrap/ListGroup";

class App extends Component {
    constructor(props) {
        super(props);

        // Setting up state
        this.state = {
            userInput: "",
            list: [],
        };
    }

    // Set a user input value
    updateInput(value) {
        this.setState({
            userInput: value,
        });
    }
```

```
// Add item if user input in not empty
addItem() {
    if (this.state.userInput !== "") {
        const userInput = {
            // Add a random id which is used to delete
            id: Math.random(),

            // Add a user value to list
            value: this.state.userInput,
        };

        // Update list
        const list = [...this.state.list];
        list.push(userInput);

        // reset state
        this.setState({
            list,
            userInput: "",
        });
    }
}

// Function to delete item from list use id to delete
deleteItem(key) {
    const list = [...this.state.list];

    // Filter values and leave value which we need to delete
    const updateList = list.filter((item) => item.id !== key);

    // Update list in state
    this.setState({
        list: updateList,
    });
}

editItem = (index) => {
  const todos = [...this.state.list];
  const editedTodo = prompt('Edit the todo:');
  if (editedTodo !== null && editedTodo.trim() !== '') {
    let updatedTodos = [...todos]
    updatedTodos[index].value= editedTodo
    this.setState({
      list: updatedTodos,
  });
  }
```

```
    }

    render() {
        return (
            <Container>
                <Row
                    style={{
                        display: "flex",
                        justifyContent: "center",
                        alignItems: "center",
                        fontSize: "3rem",
                        fontWeight: "bolder",
                    }}
                >
                    TODO LIST
                </Row>

                <hr />
                <Row>
                    <Col md={{ span: 5, offset: 4 }}>
                        <InputGroup className="mb-3">
                            <FormControl
                                placeholder="add item . . . "
                                size="lg"
                                value={this.state.userInput}
                                onChange={(item) =>
                                    this.updateInput(item.target.value)
                                }
                                aria-label="add something"
                                aria-describedby="basic-addon2"
                            />
                            <InputGroup>
                                <Button
                                    variant="dark"
                                    className="mt-2"
                                    onClick={() => this.addItem()}
                                >
                                    ADD
                                </Button>
                            </InputGroup>
                        </InputGroup>
                    </Col>
                </Row>
                <Row>
                    <Col md={{ span: 5, offset: 4 }}>
                        <ListGroup>
```

```
                              {/* map over and print items */}
                              {this.state.list.map((item, index) => {
                                  return (
                                    <div key = {index} >
                                      <ListGroup.Item
                                          variant="dark"
                                          action
                                          style={{display:"flex",
                                                  justifyContent:'space-
between'
                                        }}
                                      >
                                          {item.value}
                                          <span>
                                          <Button
style={{marginRight:"10px"}}
                                          variant = "light"
                                          onClick={() =>
this.deleteItem(item.id)}>
                                            Delete
                                          </Button>
                                          <Button variant = "light"
                                          onClick={() =>
this.editItem(index)}>
                                            Edit
                                          </Button>
                                          </span>
                                      </ListGroup.Item>
                                    </div>
                                  );
                              })}
                          </ListGroup>
                      </Col>
                  </Row>
              </Container>
          );
      }
}

export default App;
```

Steps to run the Application:

- Type the following command in the terminal:

```
npm start
```

- Type the following URL in the browser:

```
 http://localhost:3000/
```