

key attribute. You need a key for mapping a collection to an array of fragments such as to create a description list.

```
Function = (props) {  
    return (  
        <Fragment>  
        { props.item.data.map(item => (  
            <React.Fragment key={item.id}>  
                <h2>{item.name}</h2>  
                <p>{item.url}</p>  
                <p>{item.description}</p>  
            </React.Fragment>  
        ))}  
        </Fragment>  
    )  
}
```

## React Router :-

Routing is a process in which a user is directed to different pages based on their action or request. React.js Router mainly used for developing Single Page application. React Router is used to define multiple routes in the application.

React Route is a standard library system built on the top of the React and used to create routing in the React application using React Router package. It provides the synchronous URL on the browser with data that will be displayed on the web page.

React contain 3 different package of routing

① react-router :- It provides the core routing components and function for the React Router application.

② react-router-native :-

It is used for mobile application.

③ react-router-dom :-

It is used for web application design.

\$ npm install react-router-dom --save

Components in React Router.

<BrowserRouter>

- It is used for handling URL

<HashRouter>

- It is used to handle static request.

Benefits of React Router:-

① It is not necessary to set the browser history manually.

② Link uses a navigate the internal links in the application. It is similar to anchor tag.

③ It uses switch feature for rendering

④ If the router needs only a single child element.

## React CSS

CSS in React is used to style the React App or component. The style attribute is the most common attribute for styling in React applications, which adds dynamically-computed styles at render time. It accepts a javascript object in camelcased properties rather than CSS string.

### ① Inline Styling

The inline styles are specified with a javascript object in camelcase version of the style name. Its value is the styles value which we usually take in a string.

App.js

```
import React from 'react';
import ReactDOM from 'react-dom';
class App extends React.Component {
  render() {
    return (
      <div>
        <h1 style={{color:"Green"}}>Hello JavaT&lt;/h1>
        <p>Here you can find all CS Tutorials</p>
      </div>
    );
  }
}
export default App;
```

## ② CSS Stylesheet :-

You can write styling in a separate file for your React application, and save the file with a CSS extension. You can import the file in your application.

App.js.

```
import React from 'react';
import ReactDOM from 'react-dom';
import './App.css';
class App extends React.Component {
  render() {
    <div>
      <h1> Hello JavaTpoint </h1>
      <p> You can find all cs tutorials </p>
    </div>
  }
}
export default App;
```

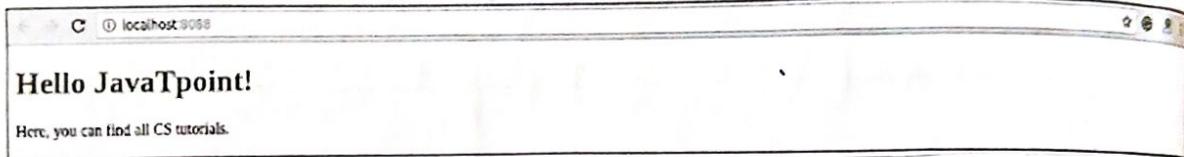
App.css

```
body {
  background-color: #008080;
  color: yellow;
  padding: 40px;
  font-family: Arial;
  text-align: center;
}
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
<meta charset = "Utf-8"/>
<meta name = "viewport"
      content = "width-device-width,
                  initial-scale=1"/>
<title> React App </title>
</head>
<body>
  <div id = "app"></div>
</body>
</html>
```



### ③ CSS Module

CSS Module is another way of adding styles to your application. In a css file where all class names and animation names are scoped locally by default. It is available only for the component which imports it means any styling you add can never be applied to other components without your permission, and you never need to worry about name conflicts.

#### App.js

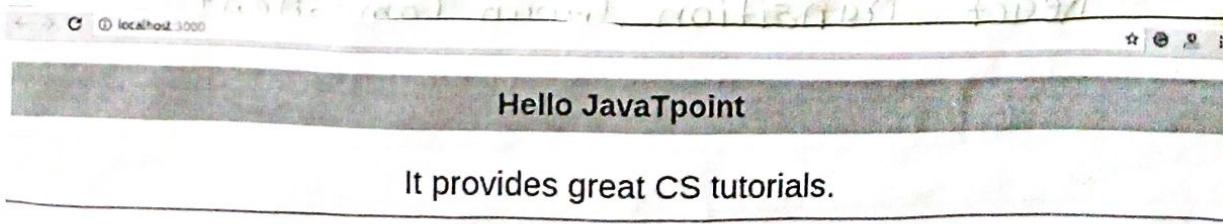
```
import React from 'react';
import ReactDOM from 'react-dom';
import styles from './myStyles.module.css'
```

```
class App extends React.Component {  
  render() {  
    return (  
      <div>  
        <h1 className={Style.myStyle}>Hello JavaT</h1>  
        <p className={Style.parastyle}> It provides great  
          CS tutorials </p>  
      </div>  
    );  
  }  
  export default App;
```

myStyles.module.css

```
• myStyle {  
  background-color: #cdcdc0;  
  color: red;  
  padding: 10px;  
  font-family: Arial;  
  text-align: center;  
}  
• parastyle {  
  color: green;  
  font-family: Arial;  
  font-size: 35px;  
  text-align: center;  
}
```

}



It provides great CS tutorials.

#### 4. Styled Component:-

Styled component is a library for React. It uses enhanced CSS for styling React components in your application which is written with a mixture of JavaScript and CSS.

The styled component provides.

- Automatic Critical CSS.
- No class name bugs.
- Easier deletion of CSS.
- Simple dynamic styling.
- Painless maintenance.

#### React Animation.

The animation is a technique in which images are manipulated to appear as moving images. It is one of the most used techniques to make interactive web applications.

React Transition group has mainly 2 API Create Transition.

##### ① React Transition Group:-

It uses as low-level API for animation.

##### ② React CSS Transition Group:-

It uses as a high-level API for implementing basic CSS transitions and animations.

React Transition Group Component.

Provides 3 main components.

① Transition

② CSS Transition

## Transition Group.

Transition:- It has a simple component API to declare a transition from one component state to another over time. It is mainly used to animate the mounting and unmounting of a component. It can also be used for in-place transition state as well.

- ① entering
- ② entered.
- ③ existing
- ④ exited.

## CSS Transition:-

The CSS transition component uses stylesheet classes to write the transition and create animations. It is inspired by the ng-animate library. It can also inherit all the props of the transition component.

- o Appear
- o Enter
- o Exist.

## Transition Group:-

This component is used to manage a set of transition component in a list. It is a state machine that controls the mounting and unmounting components overtime. The transition component does not define any animation directly. 'Transition Group' component can have different animation with a component.

## App.js

```
import React, {Component} from 'react';
import {CSSTransitionGroup} from 'react-transition-group';

class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {items: ['Blockchain', 'ReactJS',
      'TypeScript', 'JavaTpoint']};
    this.handleAdd = this.handleAdd.bind(this);
  }

  handleAdd() {
    const newItems = this.state.items.concat([
      prompt('Enter Item Name')
    ]);
    this.setState({items: newItems});
  }

  handleRemove(i) {
    let newItems = this.state.items.slice();
    newItems.splice(i, 1);
    this.setState({items: newItems});
  }

  render() {
    const items = this.state.items.map((item, i) =>
      <div key={item} onClick={()=>this.handleRemove(i)}>
        {item}
      </div>
    );
    return (
      <div>
        <h1> Animation Examples </h1>
        {items}
      </div>
    );
  }
}
```

```
<button onClick={this.handleAdd}> Insert them
    </button>
<CSSTransitionGroup
    transitionName="example"
    transitionEnterTimeout={800}
    transitionLeaveTimeout={600}>
    {items}
    </CSSTransitionGroup>
</div>
);
}
}

export default App;
```

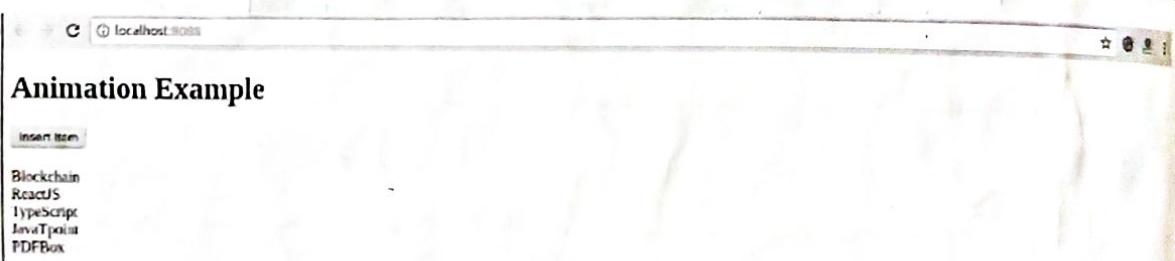
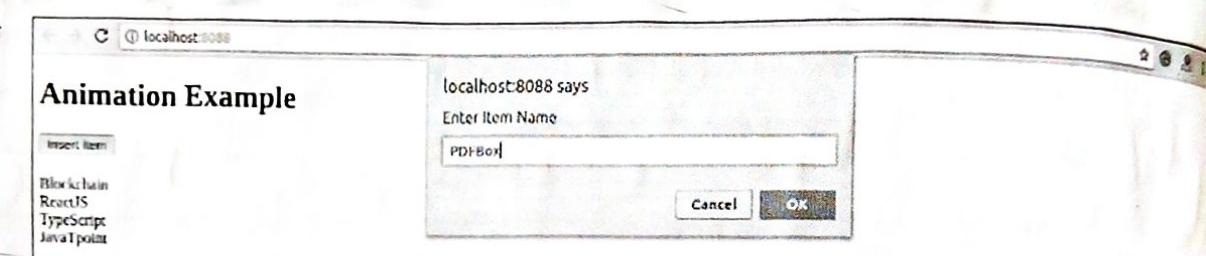
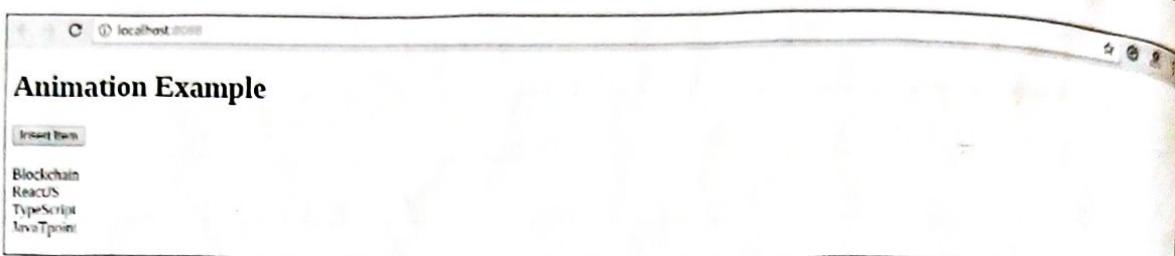
### Main.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App.js';
ReactDOM.render(<App />, document.getElementById
('app'));
```

### Style.css

```
.example-enter {
    opacity: 0.01;
}
.example-enter.example-enter-active {
    opacity: 1;
    transition: opacity 500ms ease-in;
}
.example-leave {
    opacity: 1;
}
```

```
example-leave, example-leave-active {  
    opacity: 0.01;  
    transition: opacity 300ms ease-in;  
}
```



## React Map :-

A map is a data collection type where data is stored in the form of pairs. It contains a unique key. The value stored in the map must be mapped to the key. We cannot store duplicate pair in the map. It is because of the uniqueness of each stored key. It is mainly used for fast searching and looking up data.

In React, the map() method used for:

① Traversing the list element.

```
import React from 'react';
```

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

```
function NameList(props) {
```

```
  const myList = (props) {
```

```
    const listItems = myList.map((myList) =>
```

```
      <li> {myList} </li>
```

```
    );
```

```
    return (
```

```
      <div>
```

```
        <h2> React Map example </h2>
```

```
        <ul> {listItems} </ul>
```

```
      </div>
```

```
    );
```

```
  const myLists = ['A', 'B', 'C', 'D'];
```

```
  ReactDOM.render(
```

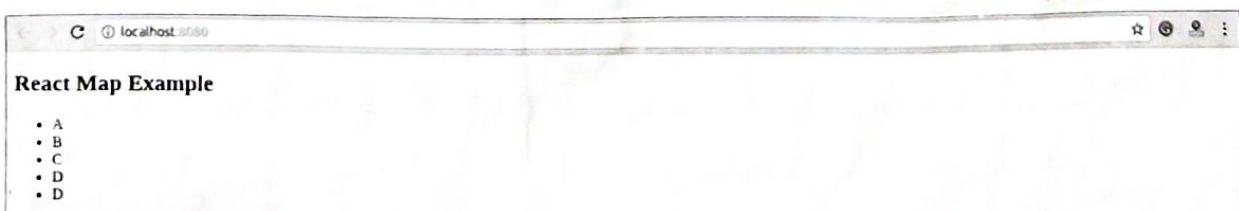
```
    <NameList myList = {myLists} />
```

```
    document.getElementById('app')
```

```
  );
```

```
export default App;
```

Output -



## React Table -

A table is an arrangement which organizes information into rows and columns. It is used to store and display data in a structured format.

The react-table is lightweight, fast, fully customizable and extendable Datagrid built for React. It is fully controllable via optional props and callbacks.

### Features:-

- ① It is lightweight at 11 kb.
- ② It is fully customizable.
- ③ It is fully controllable via optional props and callbacks.
- ④ It has client-side & server-side pagination
- ⑤ It has filters
- ⑥ Pivoting & Aggregation
- ⑦ Minimal design & easily themeable

```
import React,{Component} from 'react';
import ReactTable from "react-table";
import "react-table/react-table.css";
class App extends Component {
  render() {
    const data = [
      {name : "Ityaan", age : 26},
      {name : "Ahana", age : 22}
    ];
  }
}
```

```
name: 'Peter',
```

```
age: 40
```

```
}, {
```

```
name: 'Dhoni',
```

```
age: 37
```

```
}]
```

```
const columns = [{
```

```
header: 'Name',
```

```
accessor: 'name'
```

```
}, {
```

```
header: 'Age',
```

```
accessor: 'age'
```

```
}]
```

```
return (
```

```
<div>
```

```
<ReactTable
```

```
data = {data}
```

```
columns = {columns}
```

```
defaultPageSize = {2}
```

```
pageSizeOptions = {[2, 4, 6]}
```

```
/>
```

```
</div>
```

```
) { }
```

```
export default App;
```

## React Higher-Order Component

It also known as HOC. In React, HOC is an advanced technique for reusing component logic. It is a function that takes a component and returns a new component.

```
const NewComponent = higherOrderComponent  
(wrappedComponent)
```

## HOC.JS

```
import React, {Component} from 'react';
export default function HOC(HOCComponent){
    return class extends Component {
        render(){
            return (
                <div>
                    <HOCComponent>
                </div>
            );
        }
    }
}
```

## App.JS

```
import React {Component} from 'react';
import HOC from './HOC';
class App extends Component {
    render(){
        <div>
            <h2> HOC Example </h2>
            JavaTPoint provides best CS Tutorial
        </div>
    }
}
App = HOC(App);
export default App;
```

## Output -



HOC Example

JavaTPoint provides best CS tutorials.