

# React JS Training

(Intermediate to Advanced)



*Narasimha*

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

# Schedule for React JS Training

---

Day#	Date	Topic
Day-1	16-Apr-2024	State Management : Class Components
Day-2	17-Apr-2024	State Management : Functional Components, Hooks
Day-3	18-Apr-2024	Http Client Programming – AJAX Calls to APIs (Node JS)
Day-4	19-Apr-2024	Working with Forms, Validations, Services
Day-5	22-Apr-2024	Redux – State Management Library
Day-6	23-Apr-2024	Unit Testing React Application
Day-7	24-Apr-2024	Routing – SPA in React JS

Day-2

# State Management in React JS



*Narasimha*

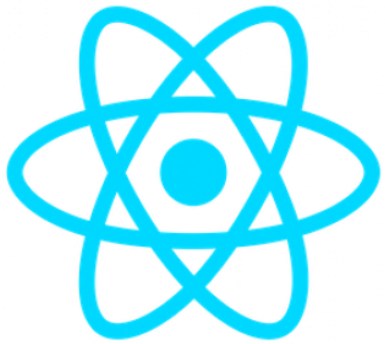
Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

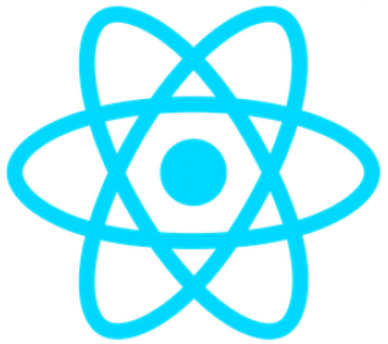
# Index – Day2

---

- 1. Functional Components**
- 2. State management using React Hooks**
- 3. useState() hook & useEffect() hook**
- 4. Adding Styles to React Components**



# Class Component VS Functional Components



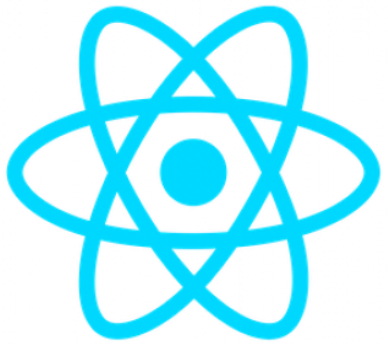
# Functional Components

# Working with State in React

---

- By default component data changes in events does not reflect on the UI.
- In order to preserve the changes to UI, react uses a concept called “**State**”.
- State is a concept of tracking the component data changes and re-render the UI.





# State management in Functional Components using React Hooks



# React Hooks

---

- Hooks are a new addition in React 16.8.
- They let you use state and other React features without writing a class (using functional components).
- Hooks generally replace class components
- Hooks allow us to "hook" into React features such as state and lifecycle methods.
- You must import Hooks from react.



# React Hooks -- Rules

---

- Hooks are JavaScript functions, but they impose two additional rules:
  1. Only call Hooks at the top level. Don't call Hooks inside loops, conditions, or nested functions.
  2. Only call Hooks from React function components. Don't call Hooks from regular JavaScript functions.

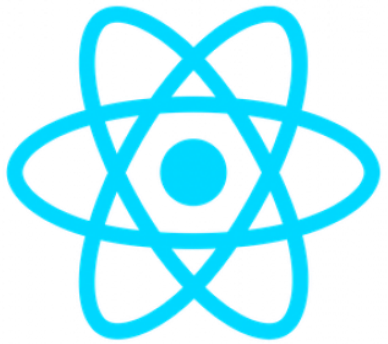


# React Hooks – Important Hooks

---

1. `useState`
2. `useEffect`
3. `useParams`
4. `useNavigate`
5. `useRef`





# useState() hook

# Using useState()

---

- The **useState()** is a function that allows you to create state variables in functional components.
- So basically **useState** is the ability to maintain local state in a **functional component**.
- The **useState hook** is a special function that takes the initial state as an argument and returns an array of two entries.
- useState encapsulate only singular value from the state, for multiple state need to have useState calls.



## Using useState()

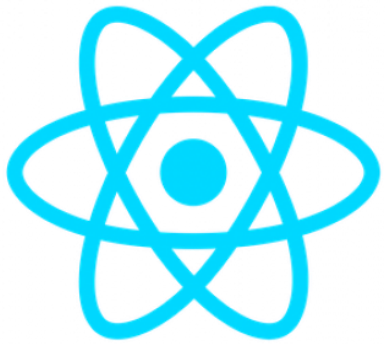
---

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

```
const [state, setState] = useState(initialstate)
```

```
const [uid, setUid] = useState("");  
const [count, setCount] = useState(0);
```





## useEffect() hook

# useEffect

---

```
import {useEffect} from 'react';
```

```
useEffect(callback, []);
```

```
useEffect(() =>
{
    // code
}, []);
```



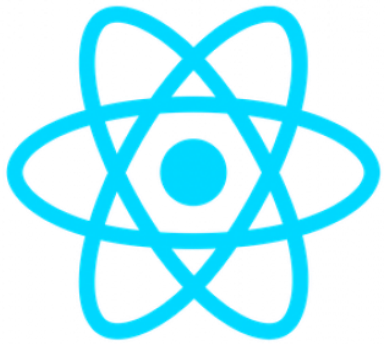


# useEffect

---

```
useEffect(() =>{  
    let url =  
    "https://www.w3schools.com/angular/customers.php";  
    axios.get(url).then( (resData) => {  
        setCustomers(resData.data.records);  
    });  
}, []);
```





# Adding Styles to React Components

# Styles in React

---

## 1. Tag Level - Inline styles

```
<p style={ { color : "blue", textAlign : "center" } }> ....</p>
```

## 2. Component Level : Define styles in separate \*.css

```
import './Demo.css';
```

.....

```
<div className="c1"> Hello World </div>
```

## 3. Project Level Define the css styles in separate \*.css (project level):

provide styles information in index.css / app.css which will be applicable to all components



# Bootstrap Styles in React

---

1. Import bootstrap.min.css in index.html
2. Apply styles to your component ui logic

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"  
      rel="stylesheet" />
```

```
<button className="btn btn-primary" onClick={button1_click}>Hello</button>
```



## Practice Hands-Ons

*Narasimha*

Sr. Corporate Trainer, Mentor

[tnrao.trainer@gmail.com](mailto:tnrao.trainer@gmail.com)



*Narasimha*

Sr. Corporate Trainer, Mentor

[tnrao.trainer@gmail.com](mailto:tnrao.trainer@gmail.com)

