



React

# React JS From Beginning - Part 1

**DEBASIS SAHA**

# Agenda

- ❖ What is ReactJs?
- ❖ Why use ReactJs?
- ❖ Generate a ReactJs Project
- ❖ Create ReactJs Elements
- ❖ Create ReactJs Component
- ❖ Define Component Properties
- ❖ Define Custom Methods
- ❖ Create Function Component
- ❖ Create Nested Component
- ❖ QnA

# **What is React Js?**

- ❖ **It is an open source JavaScript Based Library**
- ❖ **Used mainly for build User Interfaces**
- ❖ **Facebook Released React in 2013.**
- ❖ **React Native used in Mobile App Development**
- ❖ **React help us to define View Layer of the Web Applications**

# **Why to Use ReactJS?**

- ❖ **HTML is perfect for Static Type Web Application – where data not changed frequently.**
- ❖ **For Frequent Change data based application, we can use other Client Side framework like Angular, Ember etc.**
- ❖ **These framework generates dynamic HTML Content in the application.**

# Why to Use ReactJS?

- ❖ This worked great, but in case of big applications, some issue arises –
  - ❖ **Performance** – Since we no more hit refresh button every time on link click, we need to manage the memory very carefully.
  - ❖ **Slow Rendering** in the Browser for the Dynamic HTML Elements.
  - ❖ **The DOM is a bottleneck**; we should avoid it as much as possible

# Why to Use ReactJS?

- ❖ These same issues faced by Facebook engineers and they take one major decision –

*Represent Every HTML Element with a JavaScript Objects.*

## **PROS**

**SEO FRIENDLY**

**Component Can be Reused**

**Easy to Create UI Test Cases & Run**

**Fast Performance for Large Volume of Component**

**Easy Debugging using Chrome Extension Tool**

## **CONS**

**View Oriented**

**Some time requires large amount of Code**

**Requires Manual Processing for Data Updating in UI Controls**

# Generate React Project

## Steps to Make React Project

---



# **1 Need to Create-React-App**

- ✓ **Create-React-App is a tool which allow us to setup create projects.**
- ✓ **It does not require any Build Configuration.**
- ✓ **After Install, we can directly focus on our React Code directly.**

## **2** **Install Create-React-App**

- ✓ Open the command prompt and run the below command –
- ✓ **npm install -g create-react-app**

### **3** **Generate React Project**

- ✓ After install create-react-app tool, we can directly generate react project using below commands -
- ✓ **npm init react-app <project\_name>**
- ✓ After create project run the below command to view the output in browser
- ✓ **npm start**

## **4** About Package.json

- ✓ **react** – We can create react component using react library.
- ✓ **react-dom** – It provides the environment to display those component into browser DOM.
- ✓ **react-scripts** – It is mainly trans piling react syntax as a browser compatible syntax.

# Create React Element

## Steps to Create React Element

---

# **1 Use React Library**

- ✓ **Open the Index.js file**
- ✓ **First need to import the react library**
- ✓ **Next need to import react-dom library since it will be provide the render method.**

## **2** **Define Content**

- ✓ Now define the content using createElement().
- ✓ First Argument will be HTML Tag Name
- ✓ Second Argument will be Attribute name
- ✓ Third Argument will be the Content or Text need to display.

## 2

# Define Content

```
import React from 'react';  
import ReactDOM from 'react-dom';  
  
const message=React.createElement(  
  'h1',  
  {'id':'message', className: 'header'},  
  'Hello World'  
)
```



### **3** **Use DOM Render()**

- ✓ Now need to use React-DOM Render method.
- ✓ First argument will be content which need to be render into the DOM
- ✓ Second argument will be element id where this content will be render.

# **3** Use DOM Render()

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

const message=React.createElement(
  'h1',
  {'id':'message', className: 'header'},
  'Hello World'
)

ReactDOM.render(
  message,
  document.getElementById('root')
)
```

# Create React Component

## Steps to Create React Component

---

# **1 Create React Component**

- ✓ **React is Very much popular due to its Component Based Structure.**
- ✓ **React component is a class objects inherits from `React.Component`**
- ✓ **Every React Component require at least a `render()`. This method is expected to return a virtual DOM representation of the browser elements.**

# 1

# Create React Component

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

class Message extends React.Component{
  render() {
    return (
      <h1>Hello World</h1>
    )
  }
}

ReactDOM.render(
  <Message/>,
  document.getElementById('root')
)
```

# Create Function Component

## Steps to Create Function Component

---

# **1 Functional Component**

## **Class Component**

```
class App extends React.Component
{
  render() {
    return (
      <h1>Hello World!</h1>
    );
  }
}
```

## **Functional Component**

```
const App = (props) => (
  <h1>Hello World!</h1>
);
```

# **1 Functional Component**

## **Pros**

- **More Readable and Easy to Understand**
- **Easy to test (I/O Function)**
- **Enforce Good Practice**
- **No this keyword**
- **Not bounded to React**

## **Cons**

- **Can't Use Lifecycle hooks**
- **Can't use State**
- **Can't use ref**
- **Need to convert them into class component if we require to use above**



# Create Nested Component

## Steps to Create Nested Component

---

**Session Related Code Samples and PPT can be downloaded from My GitHub account :-**

[https://github.com/debasis-saha/Learn-ReactJs/tree/master/Part\\_01](https://github.com/debasis-saha/Learn-ReactJs/tree/master/Part_01)

# Thank you

QUESTIONS?