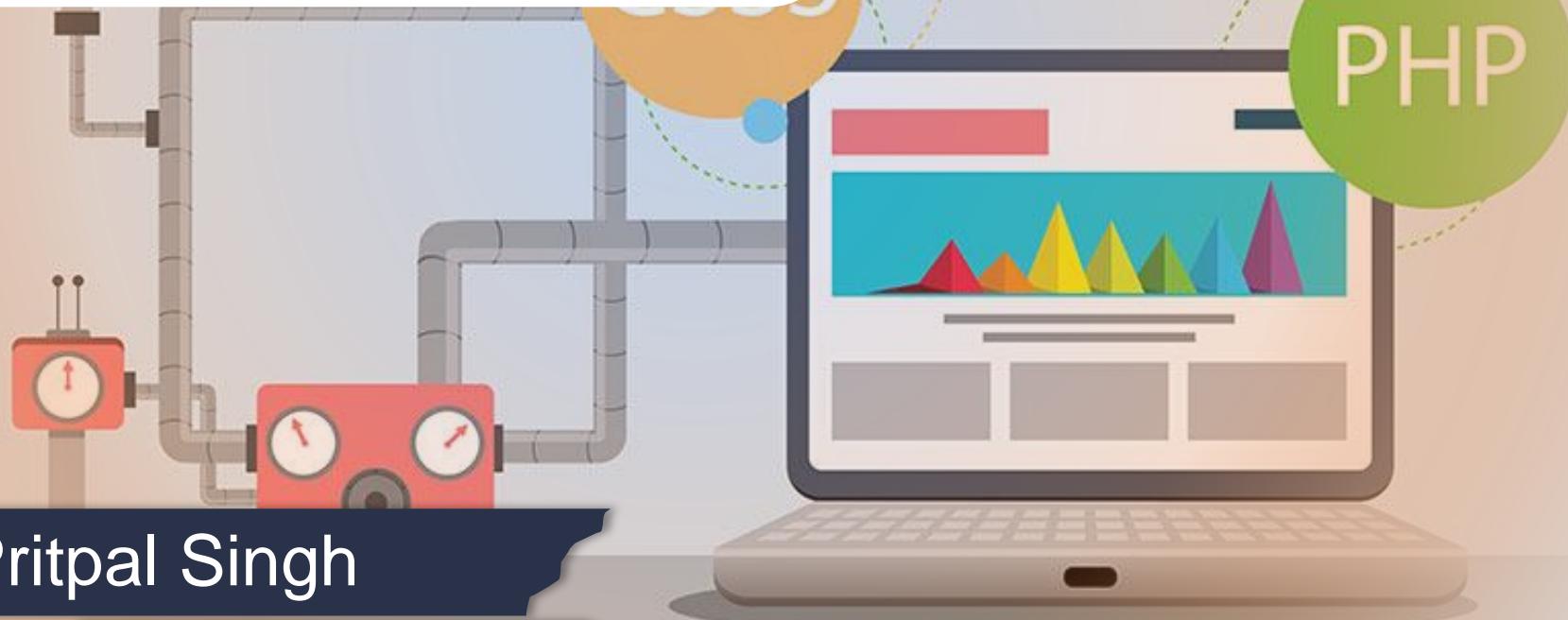


ECAP472

WEB TECHNOLOGIES



Dr. Pritpal Singh

Associate Professor

Learning Outcomes



After this lecture, you will be able to

- Understand concept of React Props and React Components.

React Components

Components are like functions
that return HTML elements.

React Components

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.

React Components

Components come in two types, Class components and Function components, in this tutorial we will concentrate on Function components.

Create Your First Component

- 01 When creating a React component, the component's name MUST start with an uppercase letter.
- 02 A class component must include the extends React.Component statement. This statement creates an inheritance to React.Component and gives your component access to React.Component's functions.
- 03 The component also requires a render() method, this method returns HTML.

Create Your First Component

- 01 When creating a React component, the component's name MUST start with an uppercase letter.
- 02 A class component must include the extends **React.Component statement**. This statement creates an inheritance to React.Component and gives your component access to React.Component's functions.
- 03 The component also requires a render() method, this method returns HTML.

Create Your First Component

- 01 When creating a React component, the component's name MUST start with an uppercase letter.
- 02 A class component must include the extends React.Component statement. This statement creates an inheritance to React.Component and gives your component access to React.Component's functions.
- 03 The component also requires a render() method, this method returns HTML.

Example

Create a Class component called Car

```
class Car extends React.  
Component  
{  
  render() {  
    return <h2>Hi, I am a Car!</h2>;  
  }  
}
```

Function Component

A Function component also returns HTML, and behaves much the same way as a Class component, but Function components can be written using much less code, are easier to understand.

Example

Create a **Function component** called Car

```
function Car()  
{  
  return <h2>Hi, I am a Car!</h2>;  
}
```

Example

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure for "PRITAPP" with files like App.css, App.js, App.test.js, index.css, index.js, logo.svg, reportWebVitals.js, setupTests.js, .gitignore, package-lock.json, package.json, and README.md.
- Editor:** The file "index.js" is open, containing the following code:

```
src > JS index.js > ...
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3
4 //function Header()
5 //{
6 //return<h1>hello world welcome to session of react</h1>;
7 //}
8
9 //class Header extends React.Component{
10 //    //render()
11 //{
12 //    //return<h1> Welcome to session of react</h1>;
13 //}
14 //}
15
16 const Header=()=>
17 {
18     return<h1> Welcome to session of react with three components</h1>;
19 }
20
21
22 ReactDOM.render(<Header/>,
23     document.getElementById('root')
24 );
```
- Annotations:** Three arrows highlight specific parts of the code:
 - A blue arrow points to the opening brace of the class definition: `//class Header extends React.Component{`.
 - An orange arrow points to the opening brace of the const declaration: `const Header=()=>`.
 - A yellow arrow points to the closing brace of the function declaration: `}`.
- Status Bar:** Shows file statistics: Ln 27, Col 1, Spaces: 2, UTF-8, LF, {}, JavaScript.
- Taskbar:** Shows icons for File Explorer, Terminal, and other applications like Google Chrome, Microsoft Edge, and Microsoft Word.

Why Should You Use React?

- React is remarkably flexible. Once you have learned it, you can use it on a vast variety of platforms to build quality user interfaces. React is a library, NOT a framework. Its library approach has allowed React to evolve into such a remarkable tool.
- React was created with a single focus: to create components for web applications. A React component can be anything in your web application like a Button, Text, Label, or Grid.

Why Should You Use React?

- React is remarkably flexible. Once you have learned it, you can use it on a vast variety of platforms to build quality user interfaces. React is a library, NOT a framework. Its library approach has allowed React to evolve into such a remarkable tool.
- React was created with a single focus: to create components for web applications. A React component can be anything in your web application like a Button, Text, Label, or Grid.

Why Should You Use React?

- React is remarkably flexible. Once you have learned it, you can use it on a vast variety of platforms to build quality user interfaces. React is a library, NOT a framework. Its library approach has allowed React to evolve into such a remarkable tool.
- React was created with a single focus: to create components for web applications. A React component can be anything in your web application like a Button, Text, Label, or Grid.

React Props

Props are arguments passed into React components.

Props are passed to components via HTML attributes.

Props stands for properties.

React Props

- Props stand for "**Properties.**"
- They are read-only components. It is an object which stores the value of attributes of a tag and work similar to the HTML attributes.
- It gives a way to pass data from one component to other components.
- It is similar to function arguments. Props are passed to the component in the same way as arguments passed in a function.

React Props

- Props are immutable so we cannot modify the props from inside the component. Inside the components, we can add attributes called props. These attributes are available in the component as `this.props` and can be used to render dynamic data in our render method.
- When you need immutable data in the component, you have to add props to `reactDom.render()` method in the `main.js` file of your ReactJS project and used it inside the component in which you need

React Props

Props can be used to pass any kind of data such as:

String

Array

Integer

Boolean

Objects or,

Functions

React Props

- To send props into a component, use the same syntax as HTML attributes:
- Example
 - Add a "brand" attribute to the Car element:
 - `const myelement = <Car brand="Ford" />;`

React Props

- The component receives the argument as a props object:
- Example
 - Use the brand attribute in the component:

```
function Car(props) {  
  return <h2>I am a { props.brand }!</h2>;  
}
```

Pass Data

Props are also how you pass data from one component to another, as parameters.

Example-Send the "brand" property from the Garage component to the Car component:

```
function Car(props) {  
  return <h2>I am a { props.brand }!</h2>;  
}  
  
function Garage() {  
  return (  
    <>  
      <h1>Who lives in my garage?</h1>  
      <Car brand="Ford" />  
    </>  
  );  
}  
  
ReactDOM.render(<Garage />, document.getElementById('root'));
```

React Props

- Props stand for "**Properties.**"
- They are read-only components. It is an object which stores the value of attributes of a tag and work similar to the HTML attributes.
- It gives a way to pass data from one component to other components.
- It is similar to function arguments. Props are passed to the component in the same way as arguments passed in a function.

Properties validation

- Properties validation is a useful way to force the correct usage of the components. This will help during development to avoid future bugs and problems, once the app becomes larger. It also makes the code more readable, since we can see how each component should be used.

Need of Validating Props in React JS

- Props are used to passing the read-only attributes to React components. For the proper functioning of components and to avoid future bugs and glitches it is necessary that props are passed correctly. Hence, it is required to use props validation for improving react component's performance.

Need of Validating

- React JS has an inbuilt feature for validating props data type to make sure that values passed through props are valid. React components have a property called propTypes which is used to setup data type validation.

Syntax

The syntax to use propTypes is shown below.

```
class Component extends React.Component {  
  render() {}  
}  
  
Component.propTypes = {/* definition goes  
here */};
```

Validators

Prop Types object contains a list of validators for basic data types, some of them are:

PropTypes.any

This means the prop can be of any data type.

PropTypes.bool

This means the prop should be a boolean.

PropTypes.number

This means the prop should be a number.

Validators

PropTypes.string

This means the prop should be a string.

PropTypes.func

This means the prop should be a function.

PropTypes.array

This means the prop should be an array.

PropTypes.object

This means the prop should be an object.

Validators

PropTypes.symbol

This means the prop should be a symbol.

PropTypes.instanceOf

This means the prop should be an instance of a particular JavaScript class.

PropTypes.isRequired

This means the prop should be provided.

PropTypes.oneOf()

This means the props should be one of several types of specified values.

PropTypes.element

This means the props must be an element.

Why PropTypes is used in React?

PropTypes are a mechanism to ensure that components use the correct data type and pass the right data, and that components use the right type of props, and that receiving components receive the right type of props.

What Do You Need to Know About React Web Development

- Netflix is a very good React website example; as well as Salesforce, Yahoo Mail, Flipboard, and many React examples. Some of these were early adopters while others joined in late. What's common to all of them is that, at some point, they took a decision to switch from a previous tech stack. Then, we need to ask, what makes this JS library stand out?

What Do You Need to Know About React Web Development

- It is one of the most popular open-source JavaScript libraries today that prides itself upon enabling developers to build interactive user interfaces. Thus, there are popular websites built with React in abundance. It was originally developed as an internal technology for use within Facebook but became open-source in 2013

Popular React JS Websites List

Facebook

React was created by a software engineer at Facebook. Since then, Facebook has maintained the framework even though it remains open-source. As such, it would be unusual if Facebook does not use it while others do. There is a lot of interactivity on the Facebook website.

British Broadcasting Corporation

This is another one of the top websites built with React. The BBC has migrated its website since 2015 and it still uses it till now, although it moved to a new React-based application. Single Page Application built by the BBC World Service as a rendering platform.

Salesforce

Salesforce is another one of the React site examples that were an early adopter. According to Salesforce' blog, the experience as more of a library than a framework. Particularly, because it is presentational and declarative, it could be built into existing projects, conveniently.

Asana

Yet another firmly established among top react websites is Asana. One of Asana's primary motivations for migrating was to build a system that would be optimized for developers to be the most productive. They had a First Experience (FX) team whose goal was to deliver the maximum experience for first time users in the first few seconds. This FX team had to facilitate cohesion between Asana's Luna framework and React.

Instacart

Instacart joined the list of best React sites in 2016, when the grocery delivery and pick-up service moved the technology stack for its website, which had been built as a single-page application (SPA) using jQuery, Haml, Underscore, and Backbone. However, this app started having issues with performance and debugging as it grew in size.

Dropbox (HelloSign)

One of the best websites built with React is Dropbox.

Last year, the Dropbox tech team documented how it migrated the HelloSign Editor from jQuery to React.

Before then, it already used the library for the Signer.

HelloSign is a company owned by Dropbox and it offers a way to electronically sign documents.

Scribd

- Another one of the top websites built with React is the book subscription service, Scribd, which uses it for the frontend of its website. Initially, implementing it was challenging, due issues with updates. Particularly, the use of JSX for code in React made it ‘joyfully easy to read and write.’ The Scribd team was also motivated by the fact that it enjoyed good support from the developer community. Thus, coding in JSX was less challenging than the alternatives.

Practical

That's all for
now...