React JS PropTypes

* **In React JS, props, which stand for “properties”, pass data from one component to another.**
* **In React JS, If a component receives the wrong type of props, it can lead to bugs and unexpected errors in your app.**
* **As we all know that, JavaScript does not have a built-in type-checking solution, many developers use extensions such as TypeScript and Flow.**
* **However, React has an internal mechanism for props validation called prop types (type checking).**
* **In React JS, We pass different types of information, such as integers, strings, arrays, etc., as props to the components.**
* **In React JS, we can either create default props or pass the props directly as attributes of the components.**
* **In React JS, we passed props from outside a component and used them inside that component.**
* **But did we have to check what kind of values we are getting inside our component via props.**
* **Before React 15.5.0, proptypes are available in the React package, but in later versions of React, you need to add a dependency to your project.**

Wide Range Of Validators

* **PropTypes.array,**
* **PropTypes.bool,**
* **PropTypes.func,**
* **PropTypes.number,**
* **PropTypes.object,**
* **PropTypes.string,**
* **PropTypes.symbol,**

React Arrow Function

* **It gets shorter! If the function has only one statement, and the statement returns a value, you can remove the brackets and the return keyword.**
* **We can also create arrow functions with parameters.**
* **Arrow functions were introduced in ES6.**

Class Component In React JS

* **In React JS, these components are simple classes (made up of multiple functions that add functionality to the application).**
* **In React JS, all class based components are child classes for the Component class of ReactJS (React.Component).**
* **In React JS, the class must implement a render() member function which returns a React component to be rendered, similar to a return value of a functional component.**
* **In React JS, component name always starts with Capital Letter e-g: <App/> not <app/>**
* **In React JS, If you write tag in lowercase like <div/> then react treats this as a DOM tags but If you write tag in First letter uppercase like <App/> then it represents a react component.**

Short-Cuts Of Creating Functional Component In VS Code

* **rfc – react functional component**
* **rfce – react functional component with export**
* **rfcp – react functional component with prop types**
* **rafc – react functional component with arrow function**
* **rafce – react functional component with arrow function and export**
* **rafcp – react functional component with arrow function and prop types**

CONSTRUCTOR IN REACT JS

* **In React JS, the constructor is a method used to initialize an object’s state in a class.**
* **In React JS, constructor automatically called during the creation of an object in a class.**
* **In React JS, constructor’s concept of a constructor is the same in React.**
* **In React JS, If you don’t initialize state and you don’t bind methods, you don’t need to implement a constructor for your React component.**
* **In React JS, when you implement the constructor for a React component, you need to call super(props) method before any other statement.**
* **In React JS, If you do not call super(props) method, this.props will be undefined in the constructor and can lead to bugs.**

he setState() Method

* **In React JS, state can be updated in response to event handlers, server responses or props changes, this is done using the setState() method.**
* **In React JS, the setState() method enqueues all of the updates made to the component state and instructs React to re-render the component and its children with the updated state.**
* **In React JS, always use the setState() method to change the state object, since it will ensure that the component knows it’s been updated and calls the render() method.**

## ifference B/W Props And State

| **PROPS** | **STATE** |
| --- | --- |
| **In React JS, Props get passed to the component.** | **In React JS, State is created and managed within the component.** |
| **Function Parameters.** | **Variables.** |
| **Props are immutable / un-changeable** | **State is mutable / changeable.** |
| **props -> Functional Components** | **useState Hook –> Functional Components.** |
| **this.props -> Class Components** | **this.state -> Class Component** |