1. **What is React?**

React is an open-source front-end JavaScript library that is used for building user interfaces, especially for single-page applications. It is used for handling view layer for web and mobile apps. React was created by Jordan Walke, a software engineer working for Facebook. React was first deployed on Facebook's News Feed in 2011 and on Instagram in 2012.

1. **What are the major features of React?**

The major features of React are:

* It uses VirtualDOM instead of RealDOM considering that RealDOM manipulations are expensive.
* Supports server-side rendering.
* Follows Unidirectional data flow or data binding.
* Uses reusable/composable UI components to develop the view.

<https://reactkungfu.com/2015/10/the-difference-between-virtual-dom-and-dom/>

<https://viblo.asia/p/hieu-sao-ve-virtual-dom-trong-reactjs-bWrZngDblxw>

1. **What is JSX?**

JSX is a XML-like syntax extension to ECMAScript (the acronym stands for JavaScript XML). Basically it just provides syntactic sugar for the React.createElement() function, giving us expressiveness of JavaScript along with HTML like template syntax.

In the example below text inside <h1> tag is returned as JavaScript function to the render function.

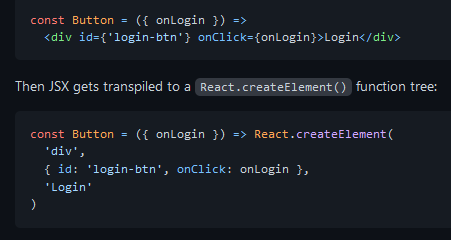


1. **What is the difference between Element and Component?**

An Element is a plain object describing what you want to appear on the screen in terms of the DOM nodes or other components. Elements can contain other Elements in their props. Creating a React element is cheap. Once an element is created, it is never mutated.

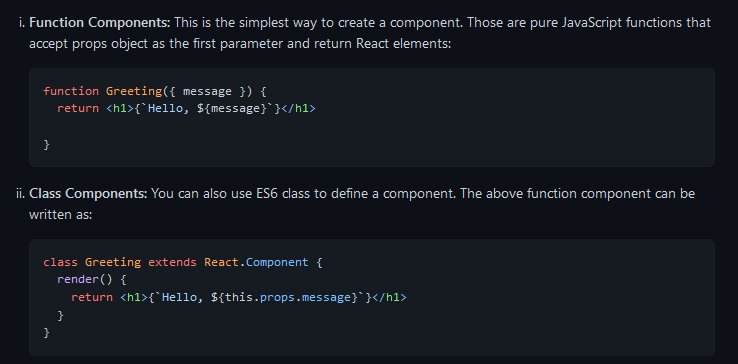


Whereas a **component** can be declared in several different ways. It can be a class with a render() method or it can be defined as a function. In either case, it takes props as an input, and returns a JSX tree as the output:



1. **How to create components in React?**

There are two possible ways to create a component.



1. **When to use a Class Component over a Function Component?**

If the component needs state or lifecycle methods then use class component otherwise use function component.

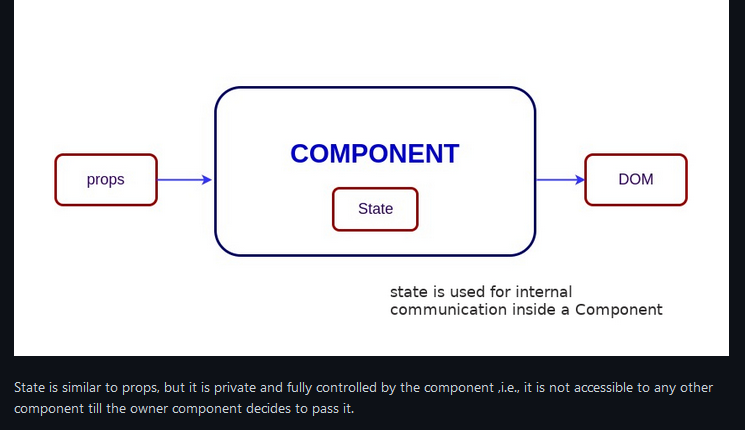
However, from React 16.8 with the addition of Hooks, you could use state , lifecycle methods and other features that were only available in class component right in your function component.

\*So, it is always recommended to use Function components, unless you need a React functionality whose Function component equivalent is not present yet, like Error Boundaries \*

1. **What is state in React?**

State of a component is an object that holds some information that may change over the lifetime of the component. We should always try to make our state as simple as possible and minimize the number of stateful components.

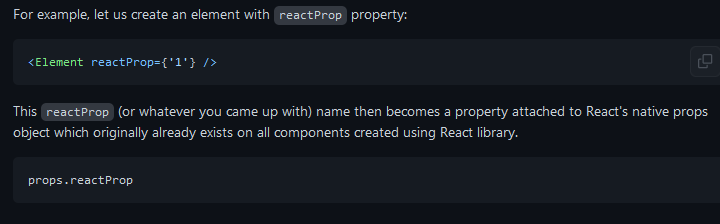




1. **What are props in React?**

Props are inputs to components. They are single values or objects containing a set of values that are passed to components on creation using a naming convention similar to HTML-tag attributes. They are data passed down from a parent component to a child component.

The primary purpose of props in React is to provide following component functionality:

* Pass custom data to your component.
* Trigger state changes.
* Use via this.props.reactProp inside component's render() method.

1. **What is the difference between state and props?**

Both props and state are plain JavaScript objects.

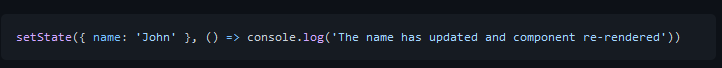
While both of them hold information that influences the output of render, they are different in their functionality with respect to component.

Props get passed to the component similar to function parameters whereas state is managed within the component similar to variables declared within a function.



1. **What is the purpose of callback function as an argument of setState()?**

The callback function is invoked when setState finished and the component gets rendered. Since setState() is asynchronous the callback function is used for any post action.

Note: It is recommended to use lifecycle method rather than this callback function.

1. .
2. .
3. 