**What is React?**

React (aka React.js or ReactJS) is a front-end and open-source JavaScript library which is useful in developing user interfaces specifically for applications with a single page. It is helpful in building complex and reusable user interface (UI) components of mobile and web applications as it follows the component-based approach.

React was created by [Jordan Walke](https://github.com/jordwalke), a software engineer working for Facebook. React was first deployed on Facebook's News Feed in 2011 and on Instagram in 2012.

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

The major features of React are:

* Uses **JSX** syntax, a syntax extension of JS that allows developers to write HTML in their JS code.
* It uses **Virtual DOM** instead of Real DOM considering that Real DOM manipulations are expensive.
* Supports **server-side rendering** which is useful for Search Engine Optimizations(SEO).
* Follows **Unidirectional or one-way** data flow or data binding.
* Uses **reusable/composable** UI components to develop the view.

**What is JSX?**

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

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

export default function App() {

return (

<h1 className="greeting">Hello, this is a JSX Code!</h1>

);

}

If you don't use JSX syntax then the respective JavaScript code should be written as below,

import { createElement } from 'react';

export default function App() {

return createElement(

'h1',

{className: 'greeting'},

'Hello, this is a JSX Code!'

);

}

### 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 cannot be mutated.

The JavaScript representation(Without JSX) of React Element would be as follows:

const element = React.createElement("div", { id: "login-btn" }, "Login");

and this element can be simiplified using JSX

<div id="login-btn">Login</div>

The above React.createElement() function returns an object as below:

{

type: 'div',

props: {

children: 'Login',

id: 'login-btn'

}

}

Finally, this element renders to the DOM using ReactDOM.render().

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:

const Button = ({ handleLogin }) => (

<div id={"login-btn"} onClick={handleLogin}>

Login

</div>

);

Then JSX gets transpiled to a React.createElement() function tree:

const Button = ({ handleLogin }) =>

React.createElement(

"div",

{ id: "login-btn", onClick: handleLogin },

"Login"

);

### How to create components in React?

Components are the building blocks of creating User Interfaces(UI) in React. There are two possible ways to create a component.

1. **Function Components:** This is the simplest way to create a component. Those are pure JavaScript functions that accept props object as the first parameter and return React elements to render the output:

function Greeting({ message }) {

return <h1>{`Hello, ${message}`}</h1>;

}

1. **Class Components:** You can also use ES6 class to define a component. The above function component can be written as a class component:

class Greeting extends React.Component {

render() {

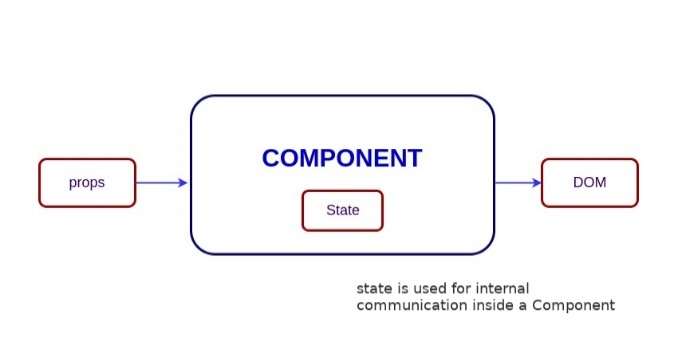
return <h1>{`Hello, ${this.props.message}`}</h1>;

}

}

**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. The important point is whenever the state object changes, the component re-renders. It is always recommended to make our state as simple as possible and minimize the number of stateful components.

[](https://github.com/sudheerj/reactjs-interview-questions/blob/master/images/state.jpg)

Let's take an example of **User** component with message state. Here, **useState** hook has been used to add state to the User component and it returns an array with current state and function to update it.

import React, { useState } from "react";

function User() {

const [message, setMessage] = useState("Welcome to React world");

return (

<div>

<h1>{message}</h1>

</div>

);

}