Interview Questions

Q1. What is JSX? (Oyo, Uber)

Answer:
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.

```
class App extends React.Component {
  render() {
    return(
      <div>
        <h1>{'Welcome to React world!'}</h1>
      </div>
    )
  }
}
```

Q2. How to create components in React? (XPRate)

Answer:
There are **two** possible ways to create a component.

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

  ```
  function Greeting({ message }) {
  ```

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

}
```

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

```
class Greeting extends React.Component {
  render() {
    return <h1>{Hello, ${this.props.message}}</h1>
  }
}
```

Q3. When to use a class component over a functional component? (Google)

Answer:

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 components right in your function component.

Q4. What is the difference between state and props? (Amazon)

Answer:

Both **props** and **state** are plain JavaScript objects. While both hold information that influences the output of **render**, they are different in their functionality with respect to components. Props get passed to the component similar to function **parameters**, whereas the state is managed within the component similar to variables declared within a function.

Q5. Why should we not update the state directly? (Expedia)

Answer:

If you try to update the state directly, then it won't **re-render** the component.

```
//Wrong
```

this.state.message = 'Hello world'
Instead use setState() method. It schedules an update to a component's state object. When the state changes, the component responds by re-rendering.

//Correct
this.setState({ message: 'Hello World' })
Note: You can directly assign to the state object either in the constructor or using the latest javascript's class field declaration syntax.