React Interview Questions & Answers

Core React

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.

   ⬆ Back to Top

2. ### 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.

   ⬆ Back to Top

3. ### 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.

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

```
      </div>
    )
  }
}
```

⬆ Back to Top

4. ## 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.

The object representation of React Element would be as follows:

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

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

```
{
  type: 'div',
  props: {
    children: 'Login',
    id: 'login-btn'
  }
}
```

And finally it renders to the DOM using `ReactDOM.render()`:

```
<div id='login-btn'>Login</div>
```

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 = ({ onLogin }) =>
  <div id={'login-btn'} onClick={onLogin}>Login</div>
```

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

```
const Button = ({ onLogin }) => React.createElement(
  'div',
  { id: 'login-btn', onClick: onLogin },
  'Login'
)
```

⬆ Back to Top

5. How to create components in React?

There are two possible ways to create a component.

i. **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:

```
function Greeting({ message }) {
  return <h1>{`Hello, ${message}`}</h1>

}
```

ii. **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>
  }
}
```

⬆ Back to Top

6. When to use a Class Component over a Function Component?