ReactJS

Lecture 5

- ReactJS Components & Life Cycle

React Component

React component is the building block of a React application.

It makes the task of building UIs much easier.

Each component exists in the same space, but they work independently from one another and merge all in a parent component, which will be the final UI of your application.

Every React component have their own structure, methods as well as APIs.

They can be reusable as per your need. For better understanding, consider the entire UI as a tree.

Here, the root is the starting component, and each of the other pieces becomes branches, which are further divided into sub-branches.

React Component

To summarize, React component provides below functionalities.

- Initial rendering of the user interface.
- Management and handling of events.
- Updating the user interface whenever the internal state is changed.

React Components

React component accomplish these feature using three concepts −

- Properties − Enables the component to receive input.
- Events − Enable the component to manage DOM events and end-user interaction.
- State − Enable the component to stay stateful. Stateful component updates its UI with respect to its state.

Creating a React component

React library has two component types. The types are categorized based on the way it is being created.

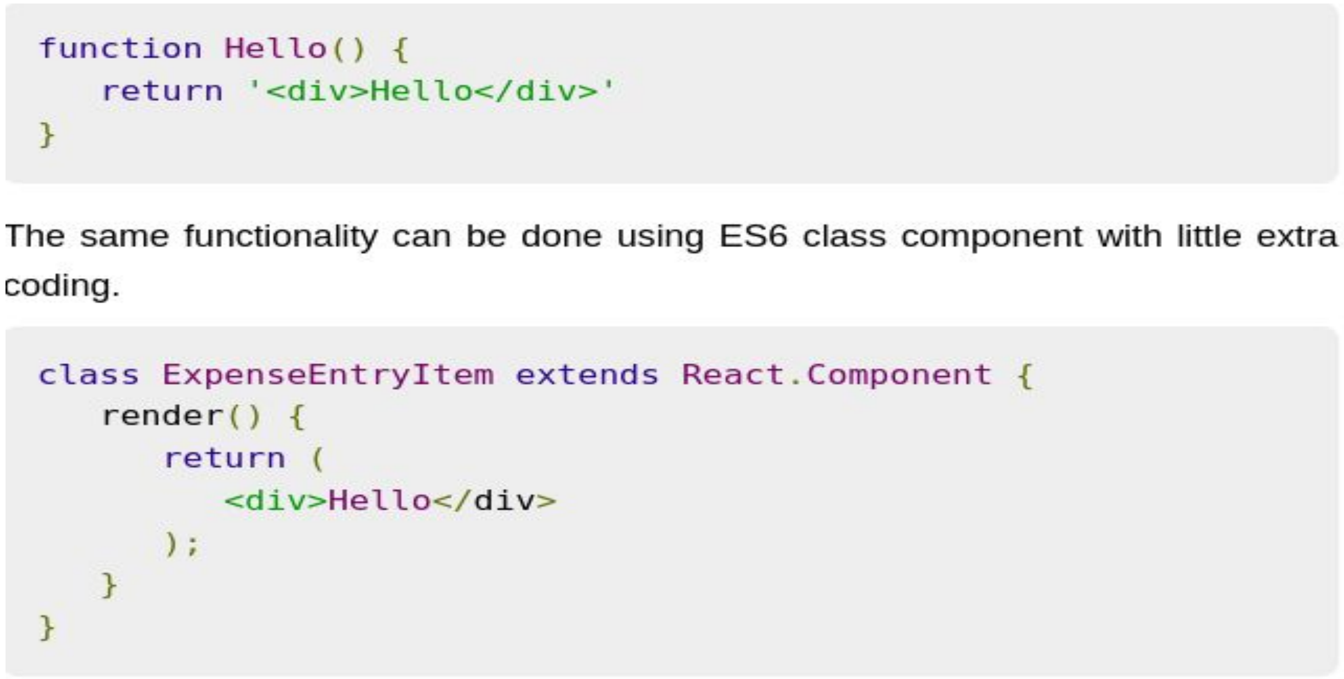
- Function component – Uses plain JavaScript function.
- ES6 class component – Uses ES6 class.

Creating a React component

The core difference between function and class component are −

- Function components are very minimal in nature. Its only requirement is to return a *React element*.
- Class components supports state management out of the box whereas function components does not support state management. But, React provides a hook, *useState()* for the function components to maintain its state.
- Class component have a life cycle and access to each life cycle events through dedicated callback apis. Function component does not have life cycle. Again, React provides a hook, *useEffect()* for the function component to access different stages of the component

Creating a React component



```
function Hello() {
   return '<div>Hello</div>'
}
```

The same functionality can be done using ES6 class component with little extra coding.

```
class ExpenseEntryItem extends React.Component {
   render() {
      return (
         <div>Hello</div>
      );
   }
}
```

Creating a function component

React component can be created using plain JavaScript function but with limited features. Function based React component does not support state management and other advanced features. It can be used to quickly create a simple component.

```
function ExpenseEntryItem() {
   return (
      <div>
         <div><b>Item:</b> <em>Mango Juice</em></div>
         <div><b>Amount:</b> <em>30.00</em></div>
         <div><b>Spend Date:</b> <em>2020-10-10</em></div>
         <div><b>Category:</b> <em>Food</em></div>
      </div>
   );
}
```

Creating a class component

Class components are more complex than functional components.

It requires you to extend from React. Component and create a render function which returns a React element.

You can pass data from one class to other class components. You can create a class by defining a class that extends Component and has a render function.

The class component is also known as a stateful component because they can hold or manage local state. It can be explained in the below example.

Creating a class component

```
import React from 'react';
import './ExpenseEntryItem.css';

class ExpenseEntryItem extends React.Component {
   render() {
      return (
         <div>
            <div><b>Item:</b> <em>Mango Juice</em></div>
            <div><b>Amount:</b> <em>30.00</em></div>
            <div><b>Spend Date:</b> <em>2020-10-10</em></div>
            <div><b>Category:</b> <em>Food</em></div>
         </div>
      );
   }
}
export default ExpenseEntryItem;
```