

Tutorial 10: ReactJS (2)

Objectives

In this tutorial, we are modifying the default react app to:

- Understand how React app runs to output the expected HTML (index.html, index.js, App.js)
- Differentiate between Class & Function components
- Create your own components using JSX
- Handle events (basic) with use of arrow functions

Discussions

Discussion 1: How React works (10mins)

- Create & your first React app.
- Observe & explain again how these components work with each other: *index.html*, *index.js*, *App.js*?

Discussion 2: When to use Class component & Function component? (10mins)

We are introduced two types of React components, including Class & Function component.

- What are the similarities & differences between them?
- When to choose Class or Function component?
- How to nest components with each other?

Tutorial Exercises

Recall: in some previous tutorials we create frontend application with pure HTML & JavaScript, including equation, card boards, flash-cards, dictionary app. All these frontend app can be refactored to use React. In this tutorial, we will refactor flash-cards app into React.

Exercise 1: FlashCards – component App (20 mins)

- Consider all flash-cards index as a big component named App. Modify `App.js` to display flash-cards as normally.

- *Should App be a class or function component?*

Exercise 2: FlashCards – other components (30 mins)

- Now decompose App into two components as designed before (solution of tut5 OOP), including FlashCard & StatusBar.
- *For each component, what should be in state & props?*
- Refactor App to use these 2 new created components
- *Are 2 components enough? Any other ways to decompose App?*

Exercise 3: FlashCards – Handle events (15 mins)

- We are now using JSX, then how to addEventListener? React makes it easier. Have a look at the example below.

```
class App extends Component {
  constructor() {
    super();
    this.state = {
      name : 'Cong'
    }
  }

  click = () => {
    alert(this.state.name);
  }

  render() {
    return (
      <div className="App">
        <h1>Hello World!</h1>
        <button onClick={this.click}>click</button>
      </div>
    );
  }
}

export default App;
```

Note that arrow function was used, the meaning of this always to be an object of class App.

- Now refactor FlashCard & StatusBar to handle user click events.