Tutorial 10: ReactJS (1)

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

Follow the instructions from our lecture to create your own first React application named "**flashcards-react**".

- Install *create-react-app* by running this command in your terminal:

```
npm install –g create-react-app
```

- Then you are able to create a React application, let's create one called *flashcards-react*.

```
npx create-react-app flashcards-react
```

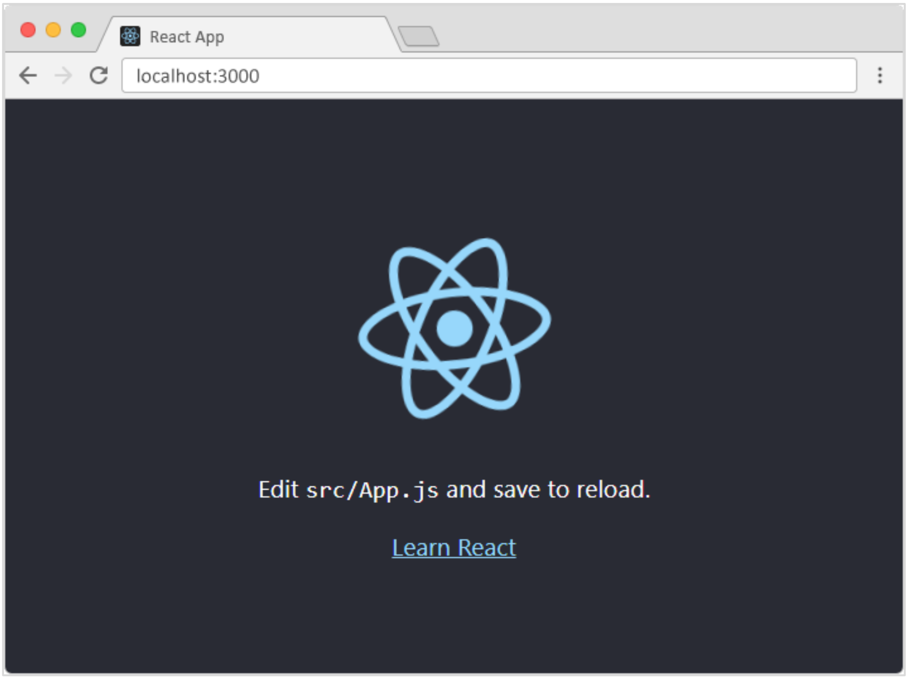
- Move to the *flashcards-react* directory

```
cd flashcards-react
```

- Run application

```
npm start
```

You should see the result like this:



Edit `src/App.js` and save to reload.

Learn React

You can start the next exercise with the generated code or to check your understanding: let's create

Code yourself

- Observe & explain again how these file work with each other: *index.html, index.js, App.js*?
- You can start with exercises using this generated code base; however, to eliminate complexity & clear your understanding, let's empty the folder **/src** then code yourself the 3 files above again.

Discussion 2: Class or Functional 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?

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 apps can be refactored to use React. In this tutorial, we will refactor flash-cards app into React.

Download the *Tutorial 04: flashcards-starterpack* & complete the exercises below.

Why tut 04? Why starter pack? ***Recall: Concept 1: Don't touch the DOM. React will do it.***

Exercise 1: component: App (20 mins)

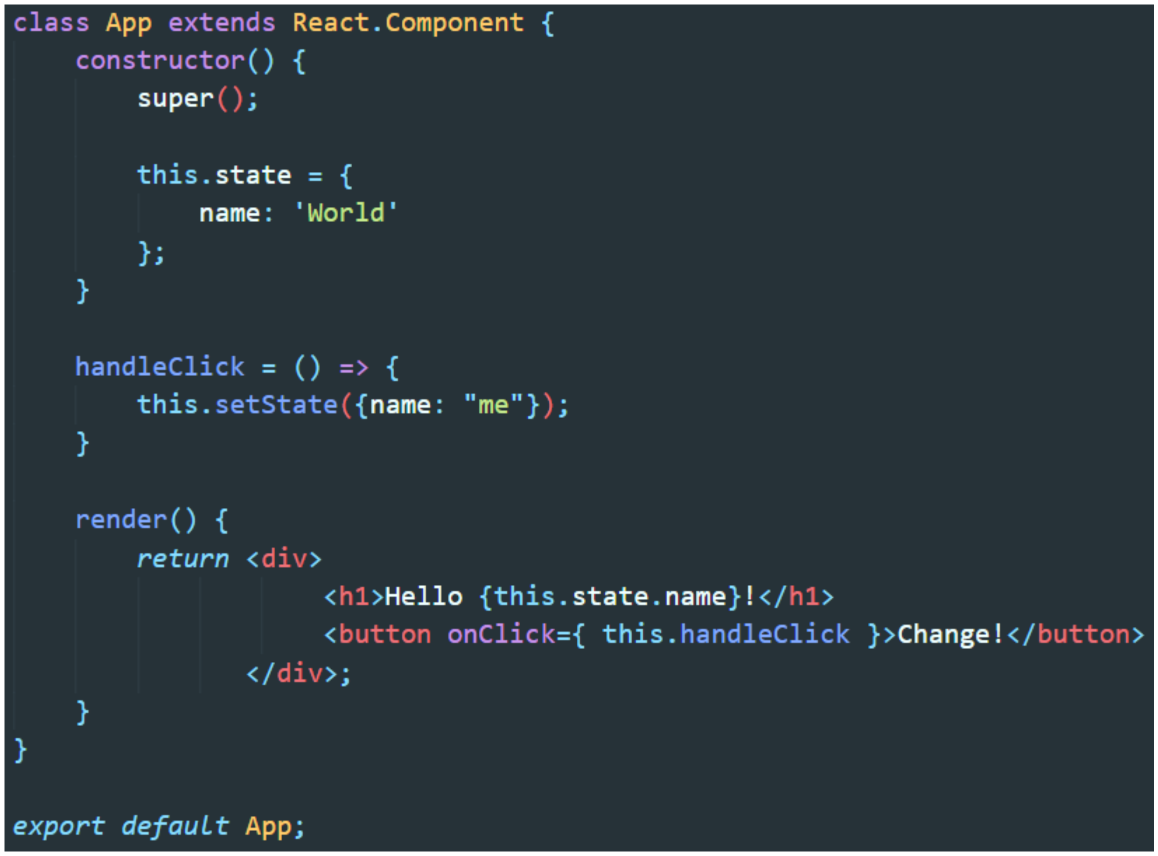
- Consider all flash-cards index as a big component named `App`. Modify `App.js` to display flash-cards as normally.
- ***Should App be a class or function component?***

Exercise 2: Show card by index (15 mins)

- Flashcards App: has an array of cards, but display only one card at a time (specified by index of to-display card in the card array)
- ***What should be in state?***
- Update status-bar to display the index & number of cards also

Exercise 3: events: Next/ Previous card (15 mins)

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



```
class App extends React.Component {
    constructor() {
        super();

        this.state = {
            name: 'World'
        };
    }

    handleClick = () => {
        this.setState({name: "me"});
    }

    render() {
        return <div>
                    <h1>Hello {this.state.name}!</h1>
                    <button onClick={ this.handleClick }>Change!</button>
               </div>;
    }
}

export default App;
```

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

- Now handle user click events on buttons.