

React 기초

멋쟁이사자처럼 9기

멋사 8기 심현아



CONTENTS

React란?

React, React 특징

React를 위한 JS

let/const, arrow functions, map & filter, module system

Event Handler와 수명주기

Event Handler – onClick, onDrag, onChange 등

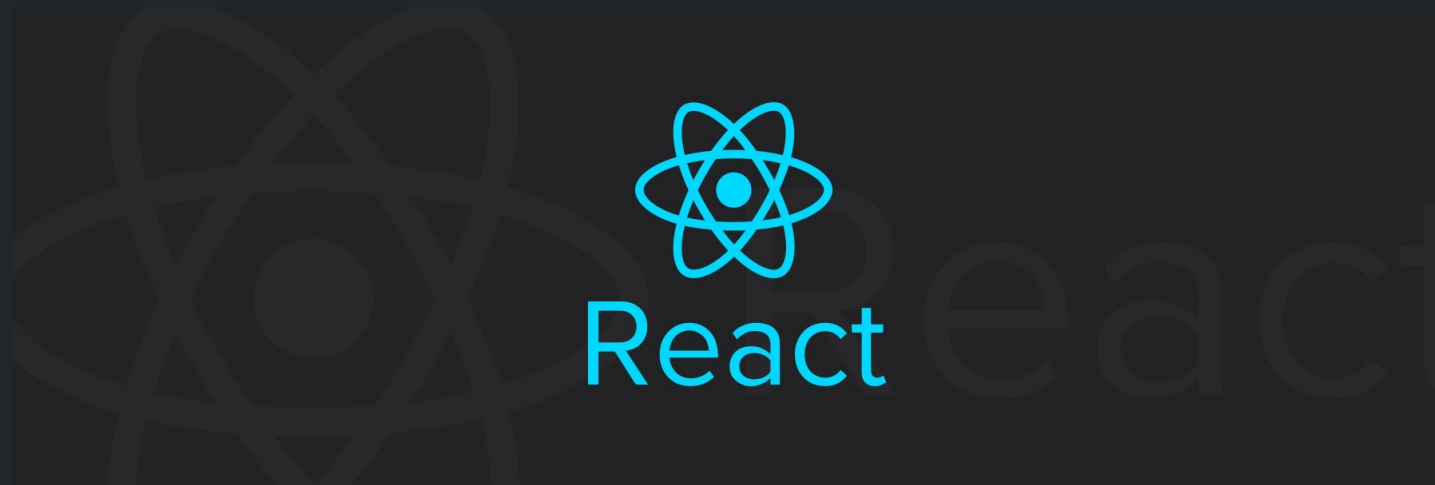
Class Component - Props와 State, React 컴포넌트 수명주기

● React 시작하기

실습(Create React App)

React란?

React, React의 특징



Javascript로 이루어진 UI 컴포넌트 라이브러리

1. 컴포넌트: 매우 독립적이며 특정 관심사에 집중된 기능 블록 → 일체형 UI에 비해 재사용, 유지보수, 확장이 용이
2. JSX: Javascript XML. 자바스크립트에서 HTML을 사용할 수 있게 해줌
3. 가상 DOM: 전체 DOM을 다시 그리지 않음. 데이터가 변할 때 바뀐 부분만을 업데이트
4. Unidirectional Data Flow: 데이터가 단방향으로 흐름. 상위 → 하위 컴포넌트

React를 위한 Javascript

let/const, arrow functions

```
1  const a = "hello";
2  a = "hellohello"; // error message
3
4  let b = "hello";
5  b = "hellohello"; // hellohello
6
7  var c = "hello";
```

Var: 기존에 쓰던 변수 선언 방법. 함수 레벨 스코프

Const, let: ES6에서 새로 나온 변수 선언 방법. 블록 레벨 스코프

→ const는 재할당 불가, let은 재할당 가능

<https://poiemaweb.com/es6-block-scope>

Keyword: hoisting

```
1  const test = function () {
2      console.log("hi");
3  };
4
5  const test = () => {
6      console.log("hihi");
7  };
```

Arrow function 화살표 함수

This 바인딩의 차이

<https://poiemaweb.com/es6-arrow-function>

React를 위한 Javascript

map & filter, module system

```
1 const data = [1, 2, 3];
2 const newData = data.map(element => {
3   return element * 2;
4 });
5
6 const filteredData = data.filter(element => element > 1);
7
8 console.log("newData: ", newData);
9 console.log("filteredData: ", filteredData);
```

```
1 export function sayhi () {
2   console.log("hello");
3 };
```

```
1 export default () => {
2   console.log("hello");
3 };
```

```
1 import { sayhi } from "test.js";
2 import sayhi from "test.js";
```

Map

https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/Array/map

Filter

https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/Array/filter

Module

<https://ko.javascript.info/modules-intro>

Event Handler와 수명주기

onClick, onChange, onDrag 등

가장 기본적인 React

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5
6 ReactDOM.render(
7   <React.StrictMode>
8     <App />
9   </React.StrictMode>,
10  document.getElementById('root')
11 );
```

Event 추가

```
1 import './App.css';
2 import Test from './test';
3
4 function App() {
5   const onClickDiv = (e) => {
6     console.log("You clicked div with " + e.target.innerHTML);
7   };
8
9   return (
10     <div className="App" onClick={onClickDiv}>
11       hello world
12     </div>
13   );
14 }
15
16 export default App;
17
```

```
1 import './App.css';
2
3 function App() {
4   return (
5     <div className="App">
6       hello world
7     </div>
8   );
9 }
10
11 export default App;
```

여기서 전달되는 event 객체는 내장 DOM 이벤트를 감싼 합성 이벤트(SyntheticEvent) – 브라우저 호환성

React가 지원하는 이벤트 핸들러: onClick, onDrag, onMouseLeave, onKeyPress, onKeyUp, onChange...

Keyword: 이벤트 전파 단계(js개념, 이벤트 캡처, 이벤트 버블링)

Event Handler와 수명주기

Class component, Function component

Class Component? Function Component?

Class Component

- state를 이용해 상태를 나타냄
- Render함수의 return 값으로 원하는 UI를 보여줄 수 있음
- componentWillMount() 와 같은 lifecycle에 정의된 메서드 이용

```
1 class ClassComp extends React.Component {
2   state = {
3     // ...
4   };
5   render() {
6     return (
7       <div className="container">
8         // ...
9       </div>
10    );
11  }
12 }
```

Function Component

- React Hook을 이용해 상태를 나타냄

```
1 function FuncComp(props) {
2   return (
3     <div className="container">
4       // ...
5     </div>
6   );
7 }
```

Event Handler와 수명주기

Props와 State

Props

```
1 import './App.css';
2 import Test from './test';
3
4 function App() {
5   const onClickDiv = (e) => {
6     console.log("You clicked div with " + e.target.innerHTML);
7   };
8
9   return (
10     <div className="App" onClick={onClickDiv}>
11       <Test title={"props 전달"}/>
12     </div>
13   );
14 }
15
16 export default App;
17
```

```
1 import React from 'react';
2
3 class Test extends React.Component {
4   render(){
5     return (
6       <div>
7         <h2>{ this.props.title }</h2>
8       </div>
9     );
10   }
11 }
12
13 export default Test;
```


Event Handler와 수명주기

Props와 State

State



```
1  import React from 'react';
2
3  class StateExample extends React.Component {
4    constructor(props) {
5      super(props);
6
7      this.state = {
8        header: "Header Initial state",
9        content: "Content Initial State"
10     };
11   }
12
13   updateHeader(text){
14     this.setState({
15       header: "Header has changed"
16     });
17   }
18
19   render() {
20     return (
21       <div>
22         <h1>{this.state.header}</h1>
23         <h2>{this.state.content}</h2>
24         <button onClick={this.updateHeader.bind(this)}>Update</button>
25       </div>
26     );
27   }
28 }
29
30 export default StateExample;
```

Header Initial state

Content Initial State

Update

Header has changed

Content Initial State

Update

Event Handler와 수명주기

수명주기

Class Component 에서의 lifecycle

Mount

1. state, context, defaultProps 저장
2. componentWillMount
3. render
4. componentDidMount

State Update

- 1.shouldComponentUpdate
- 2.componentWillUpdate
- 3.render
- 4.componentDidUpdate

Props Update

1. componentWillReceiveProps
2. shouldComponentUpdate
3. componentWillUpdate
4. render
5. componentDidUpdate

UnMount

componentWillUnmount

리액트 17부터는 componentWillMount, componentWillUpdate, componentWillReceiveProps
라이프 사이클이 deprecated됩니다

참고) <https://www.zerocho.com/category/React/post/579b5ec26958781500ed9955>

```
1  componentWillMount() {  
2      console.log('componentWillMount');  
3  }  
4  
5  componentDidMount() {  
6      console.log('componentDidMount');  
7  }  
8  
9  componentWillReceiveProps(nextProps) {  
10     console.log('componentWillReceiveProps');  
11 }  
12  
13 shouldComponentUpdate(nextProps, nextState) {  
14     console.log('shouldComponentUpdate');  
15     return true / false;  
16 }  
17  
18 componentWillUpdate(nextProps, nextState) {  
19     console.log('componentWillUpdate');  
20 }  
21  
22 componentDidUpdate(prevProps, prevState) {  
23     console.log('componentDidUpdate');  
24 }  
25  
26 componentWillUnmount() {  
27     console.log('componentWillUnmount');  
28 }
```

React 시작하기

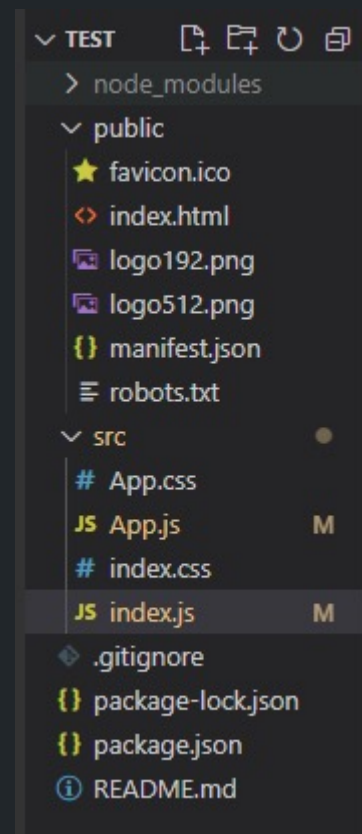
실습 (Create React App)

1. Node.js 설치

- 윈도우: <https://nodejs.org/ko/download/>
- 맥OS: `curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.8/install.sh | bash` `nvm install --lts`

2. create-react-app 설치 및 사용

설치 → `npm install -g create-react-app`
사용 → `create-react-app test`
프로젝트 폴더 들어가기 → `cd test`
프로젝트 시작 → `npm start`



3. React 기본 코드

참고) <https://velopert.com/3621>

함께보면 좋은 자료

추천추천

- 🎈 튜토리얼을 따라하고 싶다면?
 - Velopert 블로그 – React [튜토리얼](#) ([책](#)으로도 있어요!)
 - 공식문서 [튜토리얼](#)

- 🎈 공식문서?
 - [React Docs](#)

- 🎈 강의?
 - 프로그래머스 React 기본 [강의](#)
 - 노마드 코더 React로 영화 웹서비스 만들기 [강의](#)
 - 생활코딩 React [강의](#)
- 🎈 심화? 구글링 해보자!
 - React Router
 - React Hook
 - Redux