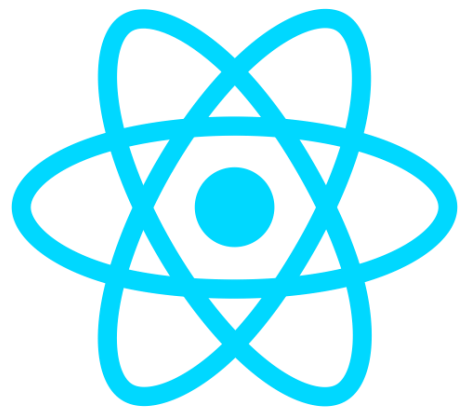


# React.js 기본 및 활용

시스템소프트웨어 연구실  
이건희

# 목차

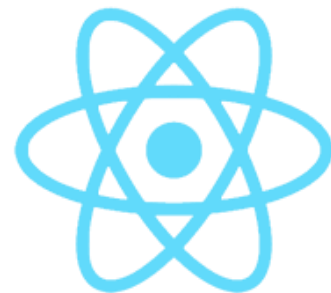
- React.js 개요
- Component, JSX, Virtual DOM
- React.js를 이용한 간단한 웹페이지 설계 및 구현



# 1. React.js 개요

# React.js ?

- 페이스북에서 개발한 유저인터페이스 라이브러리
- 개발자로 하여금 재사용 가능한 UI를 생성할 수 있게 함
- 페이스북, 인스타그램, 트위터 등 많은 큰 서비스에서 사용



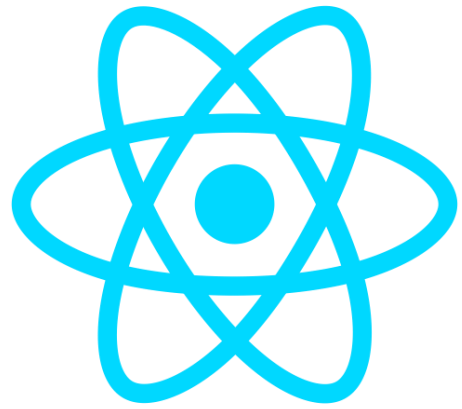
React

# React.js의 특징

- Component의 재사용성
- JSX 문법 : js의 확장 문법 (xml같은 방법으로)
- Virtual DOM 사용 : 변경된 부분만 변경 가능

# React.js의 장점

- Virtual DOM을 사용한 어플리케이션의 성능 향상
- Component에 대한 높은 가독성 및 쉬운 유지보수
- 프레임워크가 아닌 라이브러리이기에 UI 위주로 구현 가능



## 2. Component, JSX Virtual DOM

# Component

- UI를 구성하는 개별적인 **View**
- **Input**에 따라 어떻게 **output**을 할지 정하는 요소
- **React**는 개별 Component들을 하나씩 구현해서 쌓아올림
- 성능보다는 **유지보수**가 용이하게 하기 위함



# Component

```
import React, { Component } from 'react';
import './TodoItem.css';

class TodoItem extends Component {

  shouldComponentUpdate(nextProps, nextState) {
    return this.props.checked !== nextProps.checked;
  }

  render() {
    const { text, checked, id, color, onToggle, onRemove } = this.props;

    return (
      <div className="todo-item" onClick={() => onToggle(id)}>
        <div className="remove" onClick={(e) => {
          e.stopPropagation(); // onToggle 이 실행되지 않도록 함
          onRemove(id)}
        >&times;</div>
        <div style={{ color }} className={`todo-text ${checked && 'checked'}`>
          <div>{text}</div>
        </div>
        {
          checked && (<div className="check-mark">#x2713;</div>)
        }
      </div>
    );
  }
}

export default TodoItem;
```

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import registerServiceWorker from './registerServiceWorker';

ReactDOM.render(<App />, document.getElementById('root'));
registerServiceWorker();
```

# JSX (Javascript + XML)

- Javascript와 XML 합쳐진 형태의 문법을 가짐
- 순수 Javascript에 비해 상당히 **알아보기 쉽게 함**
- 직관적으로 개발을 할 수 있다는 장점이 있음

# JSX 예시 – React와 기본 Javascript

```
import React from 'react';
import ReactDOM from 'react-dom';

class Hello extends React.Component{
  render(){
    return '<div><label>{this.props.name}</label></div>'
  }
}

ReactDOM.render(<Hello name="hello" />, mountNode)
```

React.js

```
class hello {
  render(name){
    var payload_parent = document.createElement("div");
    var payload_child = document.createElement("label");
    payload_child.innerHTML = name;
    payload_parent.appendChild(payload_child);

    var old_div = document.getElementById("getter");
    document.body.insertBefore(payload_parent, old_div);
  }
}
```

기본 Javascript

# Virtual DOM

- 기존의 DOM(Document Object Model)은 바로 Rendering
- React.js의 경우 render()함수의 return 이후 이 과정을 거침
- 브라우저에서 보여지는 것과 Virtual DOM의 결과를 비교,  
그리고 변경사항이 있는 것을 진짜 DOM에 적용

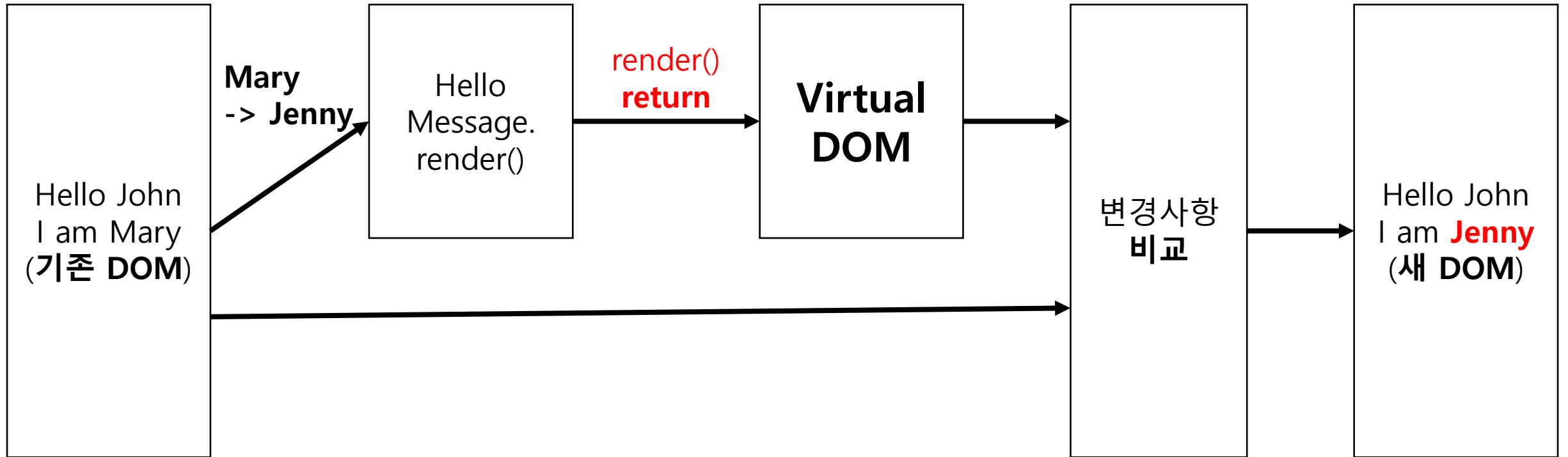
# Virtual DOM

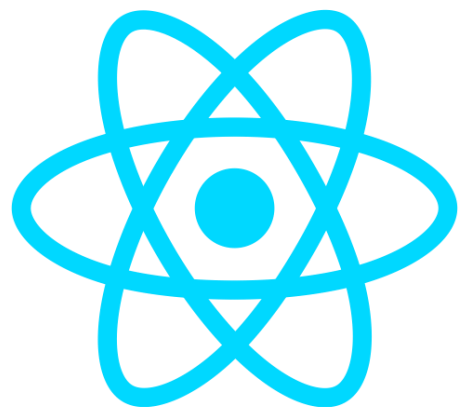
```
import React from 'react';
import ReactDOM from 'react-dom';

class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        <div>Hello {this.props.name}</div>
        <div>I am {this.state.chatName}</div>
      </div>
    );
  }
}

ReactDOM.render(<HelloMessage name="John" chatName="Mary"/>, document.getElementById('root'));
```

# Virtual DOM





### 3. React.js 를 이용한 간단한 웹페이지 설계 및 구현

# React.js 개발환경

<b>OS</b>	<b>Ubuntu 16.04 LTS</b>
<b>Language</b>	<b>Javascript, html, css</b>
<b>Tool</b>	<b>Visual Studio Code</b>



# React.js 개발환경 구축

- Node.js 설치
- Global Package 설치 (Dependency)
- create-react-app 설치 및 루트디렉토리 생성

# React.js 개발환경 구축

**<!-- Package 설정 -->**

```
$ npm install -g babel webpack webpack-dev-server
```

```
$ npm install --save react react-dom
```

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

**<!-- 루트 디렉토리 설정 및 웹어플 실행 -->**

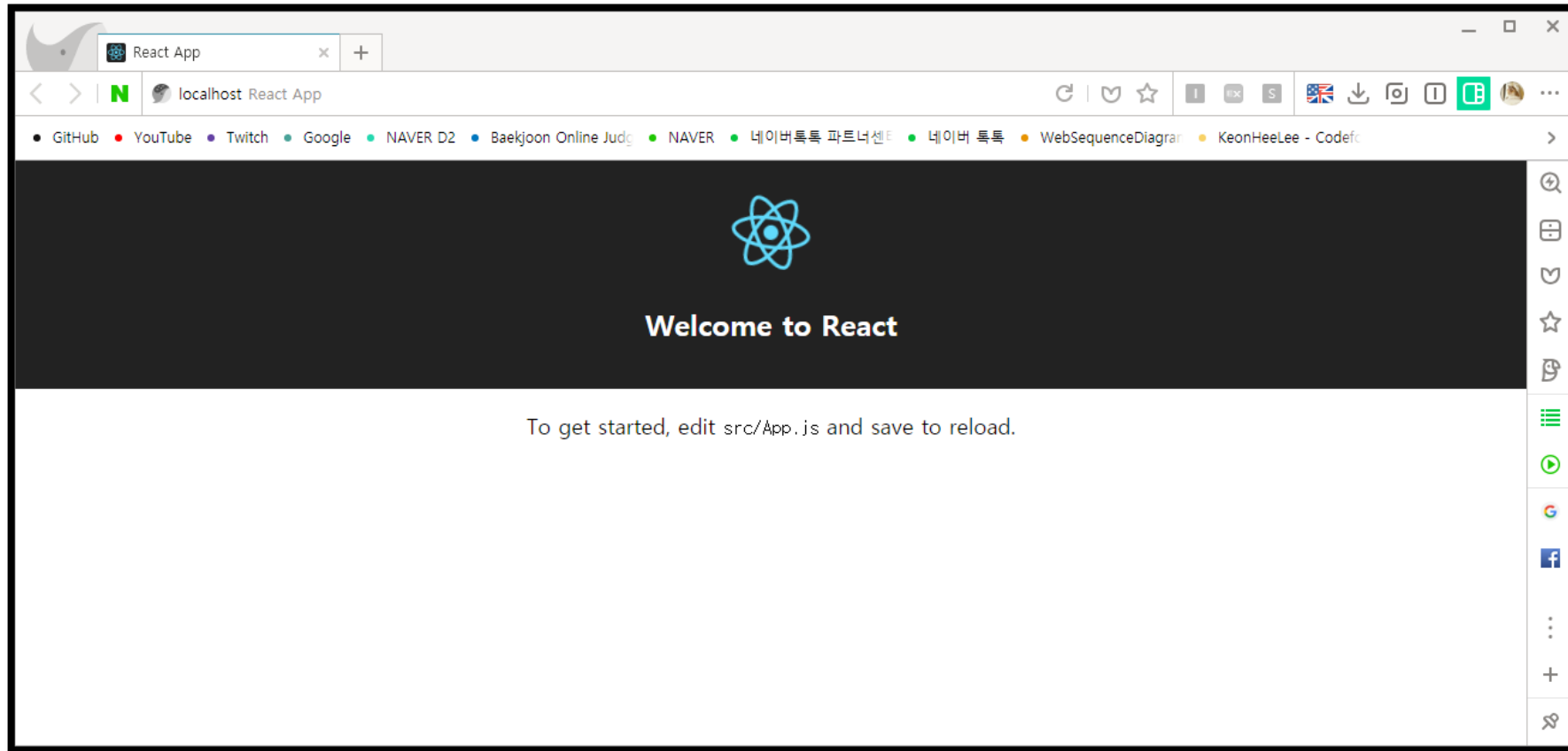
```
$ create-react-app <root-dir-name>
```

```
$ cd <root-dir-name>
```

```
$ npm start
```

Mode	LastWriteTime	Length	Name
d----	2018-07-17 오후 3:21		node_modules
d----	2018-07-17 오후 3:21		public
d----	2018-07-17 오후 3:21		src
-a----	2018-07-17 오후 3:21	285	.gitignore
-a----	2018-07-17 오후 3:21	411150	package-lock.json
-a----	2018-07-17 오후 3:21	342	package.json
-a----	2018-07-17 오후 3:21	119451	README.md

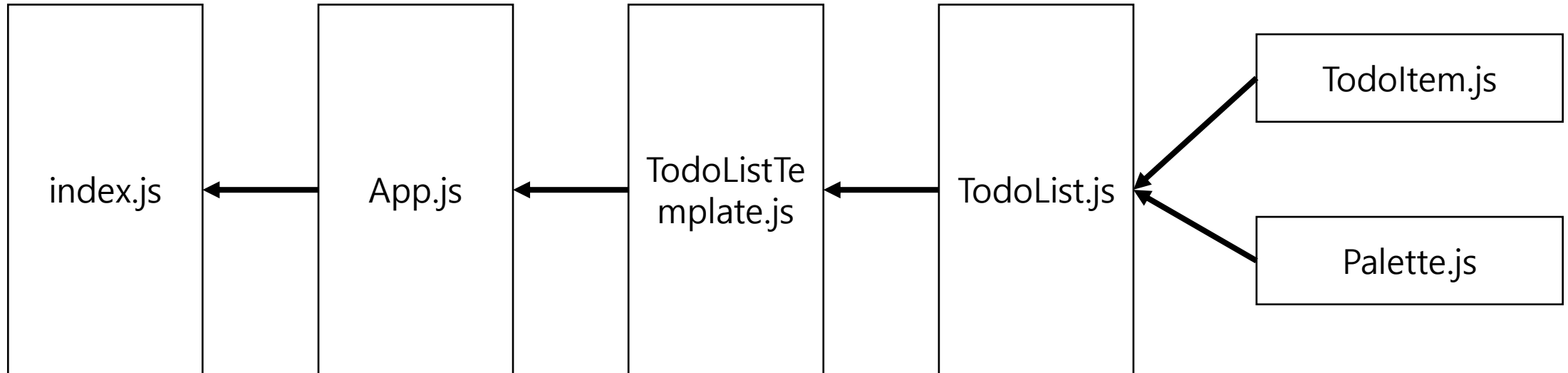
# React.js 개발환경 테스트



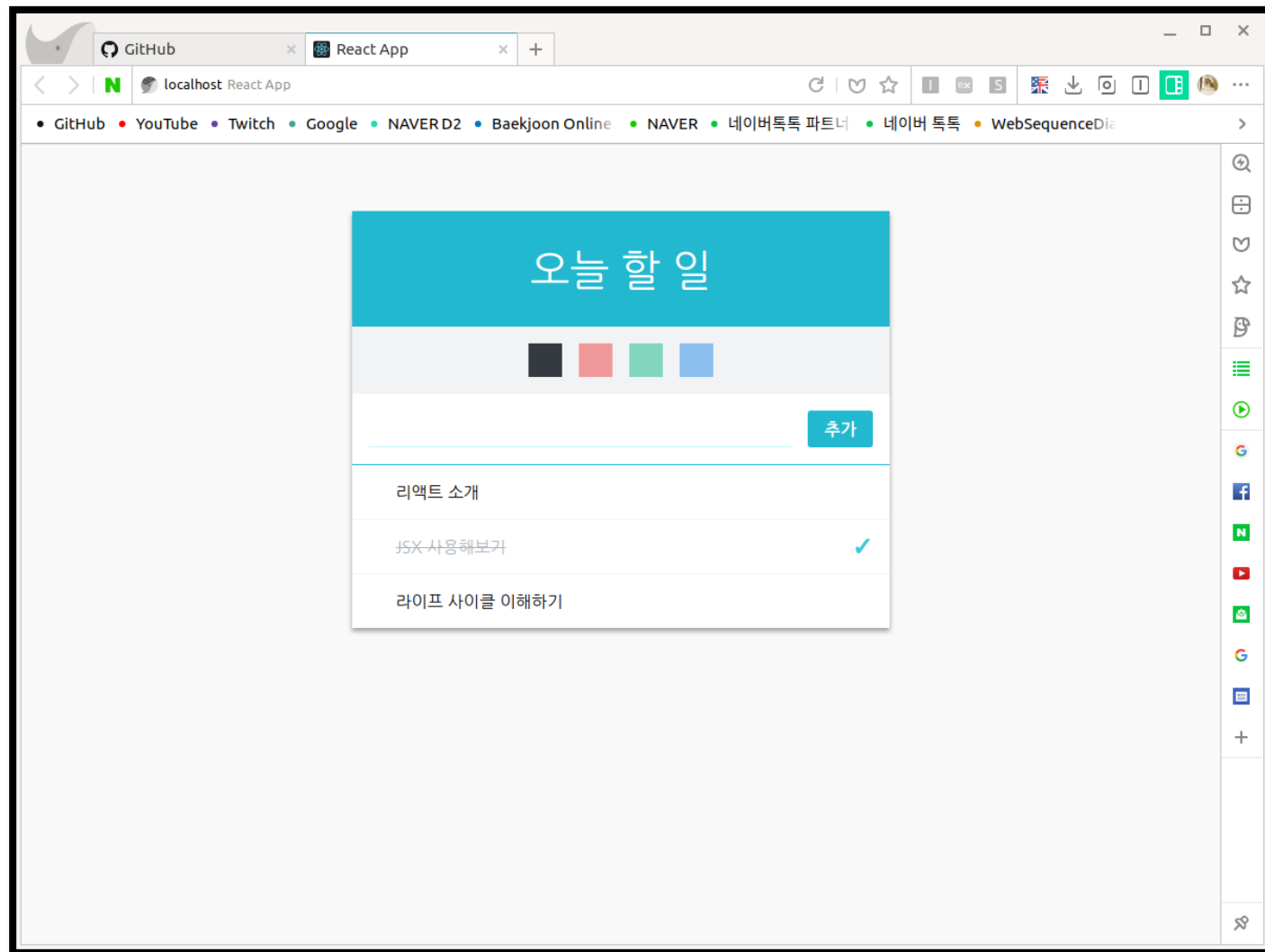
# 기본 설계

- 기본적으로 주어지는 js 파일
  - DOM rendering module => **index.js**
  - main component => **App.js**
- 개발 방법
  - 큰 컴포넌트 프로토타입 -> 작은 컴포넌트 -> 큰 컴포넌트에 반영

# 컴포넌트 설계



# 구현 결과



# 참고자료

- Facebook – React.js Github Repository  
<https://github.com/facebook/react>
- sujinleeme - React.js Tutorial 번역본 Github Repository  
<https://github.com/sujinleeme/the-road-to-learn-react-korean>
- velopert – React.js 강좌 관련 블로그  
<https://velopert.com/775>