

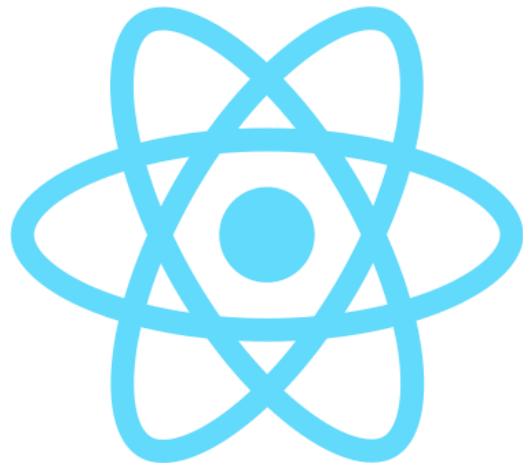
React.js 시작하기

박용준



Microsoft MVP 2020-2021

<https://www.dotnetkorea.com>





✓ 리엑트는 자바스크립트 라이브러리입니다.

리엑트는  웹과  모바일을 위한
하나의 완성된 자바스크립트 라이브러리입니다.

현재 가장 인기있는? 자바스크립트 라이브러리입니다.

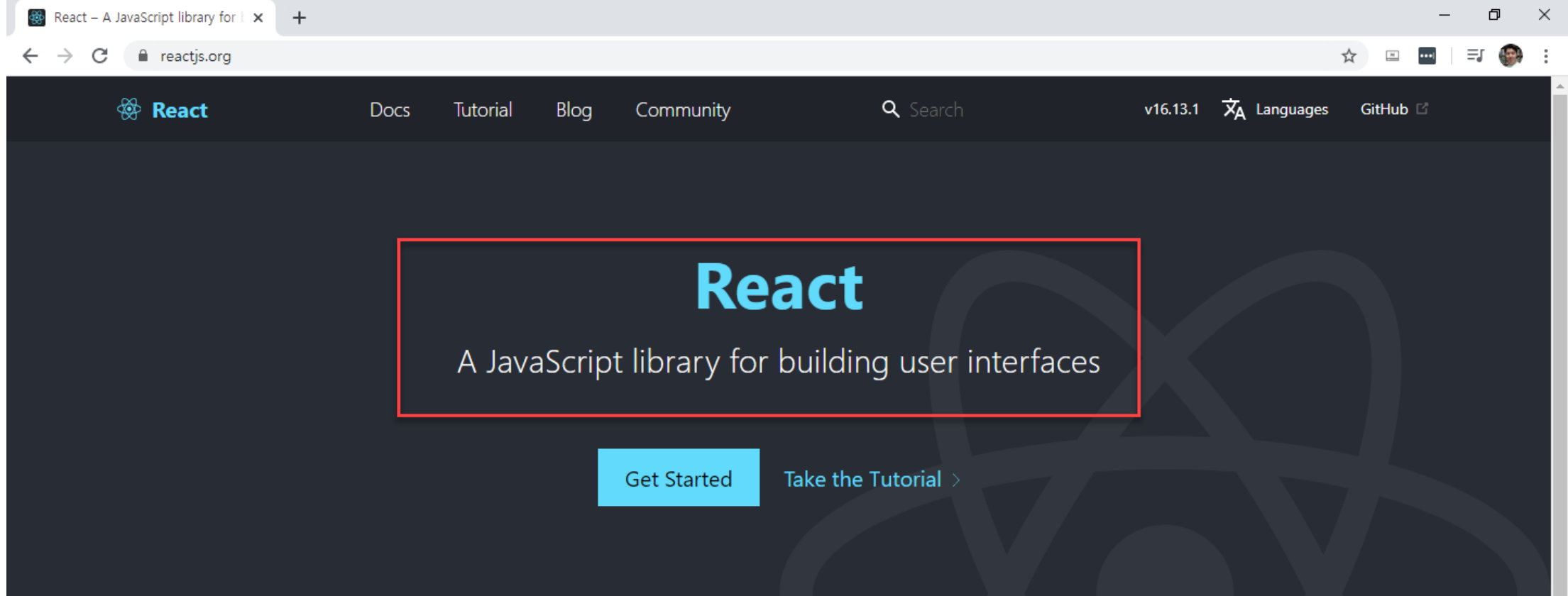


리엑트는

Angular, Blazor와 같은 프레임워크가 아닙니다.

jQuery와 같은 자바스크립트 라이브러리입니다.

MVC와 같은 디자인 패턴이 아닙니다.



Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs.

Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

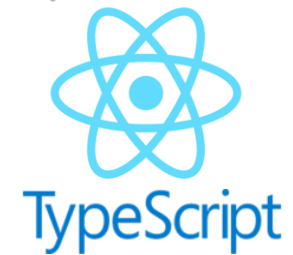
Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using [React Native](#).

리액트

- 자바스크립트 라이브러리
 - 인터랙티브한 UI 제작을 위한
- 페이스북에서 시작해서 현재 오픈소스
- 여러 회사에서 사용
 - Microsoft
 - Airbnb
 - Facebook
 - ...



- . 1995 - Introduction to JavaScript
- . 2002 - XMLHttpRequest (XHR)
- . 2006 - jQuery
- . 2008 - Chrome browser and V8 Engine – The web gets faster
- . 2009 - Nodejs
- . 2010 - Knockout JS
- . 2010 - Angular 1.0 by Google
- . 2013 – React
- . 2013 - TypeScript
- . 2016 – Angular 2.0 and ES6
- . 2017 – Angular 4.0
- . 2018 – Angular 8.0

리액트 발전

- 2011
 - 페이스북에서 개발
- 2012
 - 인스타그램에서 사용
- 2013
 - 오픈소스
- 2014
 - 확장
- 2015
 - React Native 출시
- 2016
 - React 15

리액트의 인기

- 컴포넌트
- Virtual DOM
- 페이스북에서 관리하는 오픈소스
- JSX(JS + HTML)

개발 환경 및 서비스 환경 구축

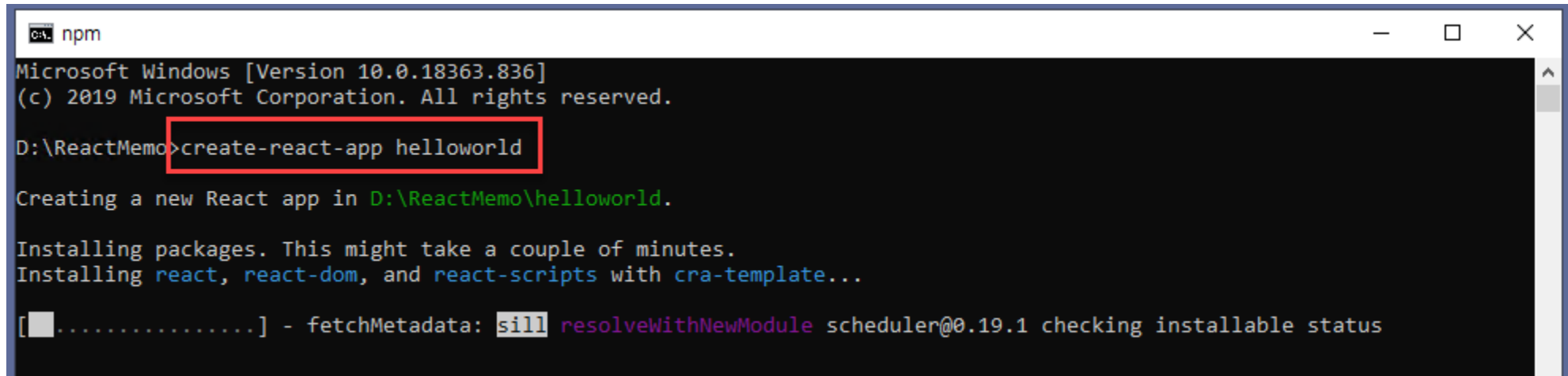
- 개발 환경
 - Windows 10
 - Visual Studio Community 2019
 - Node.js와 NPM
 - Visual Studio Code
 - Create-React-App
- 서비스 환경
 - Azure Web App
 - SQL Database
 - GitHub
 - Azure DevOps

Visual Studio Code 확장

- Simple React Snippets
- Prettier – Code formatter

기본 생성 프로젝트 살펴보기

- > create-react-app
- > dotnet new react



```
C:\> npm
Microsoft Windows [Version 10.0.18363.836]
(c) 2019 Microsoft Corporation. All rights reserved.

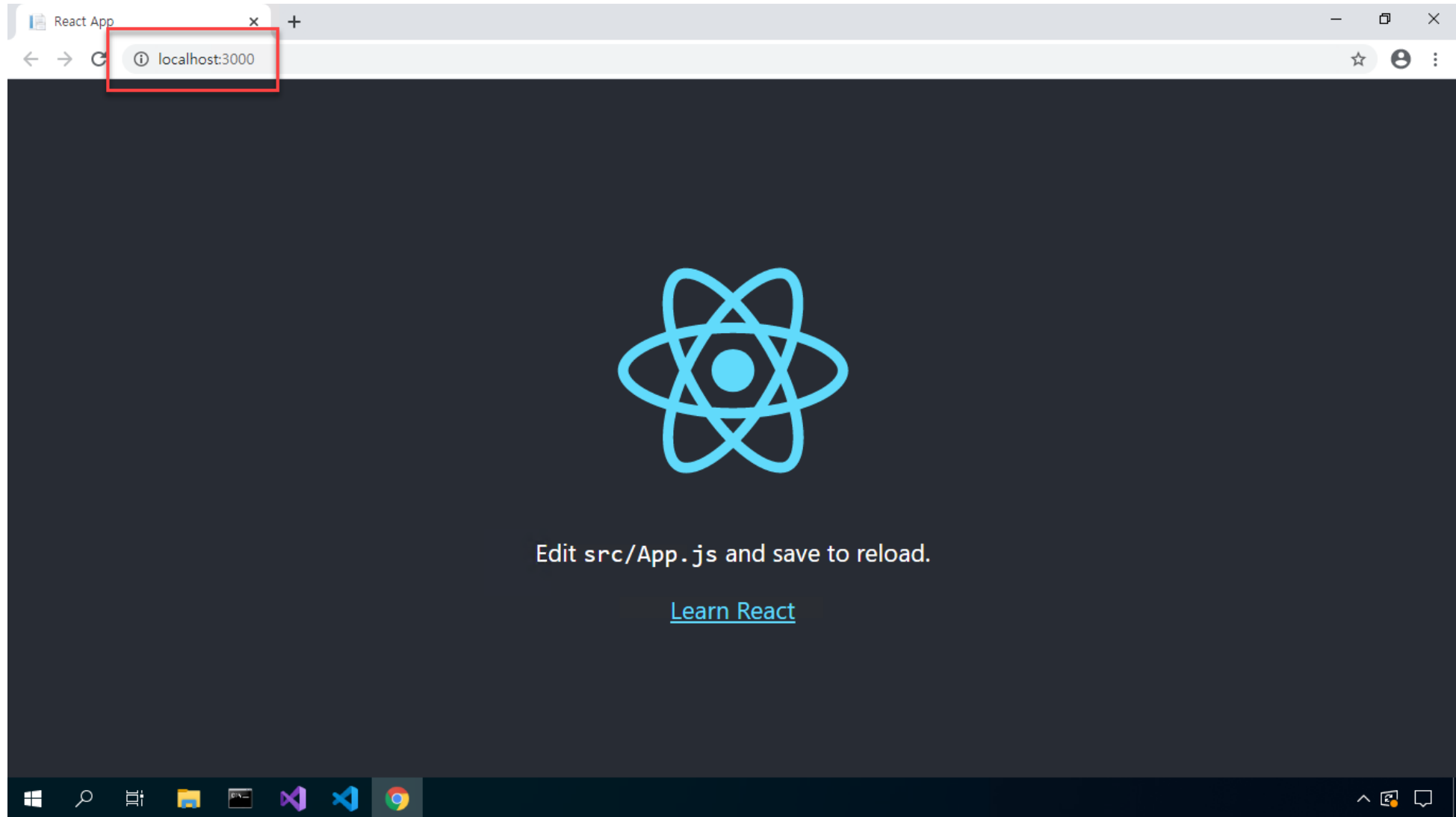
D:\ReactMemo> create-react-app helloworld

Creating a new React app in D:\ReactMemo\helloworld.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

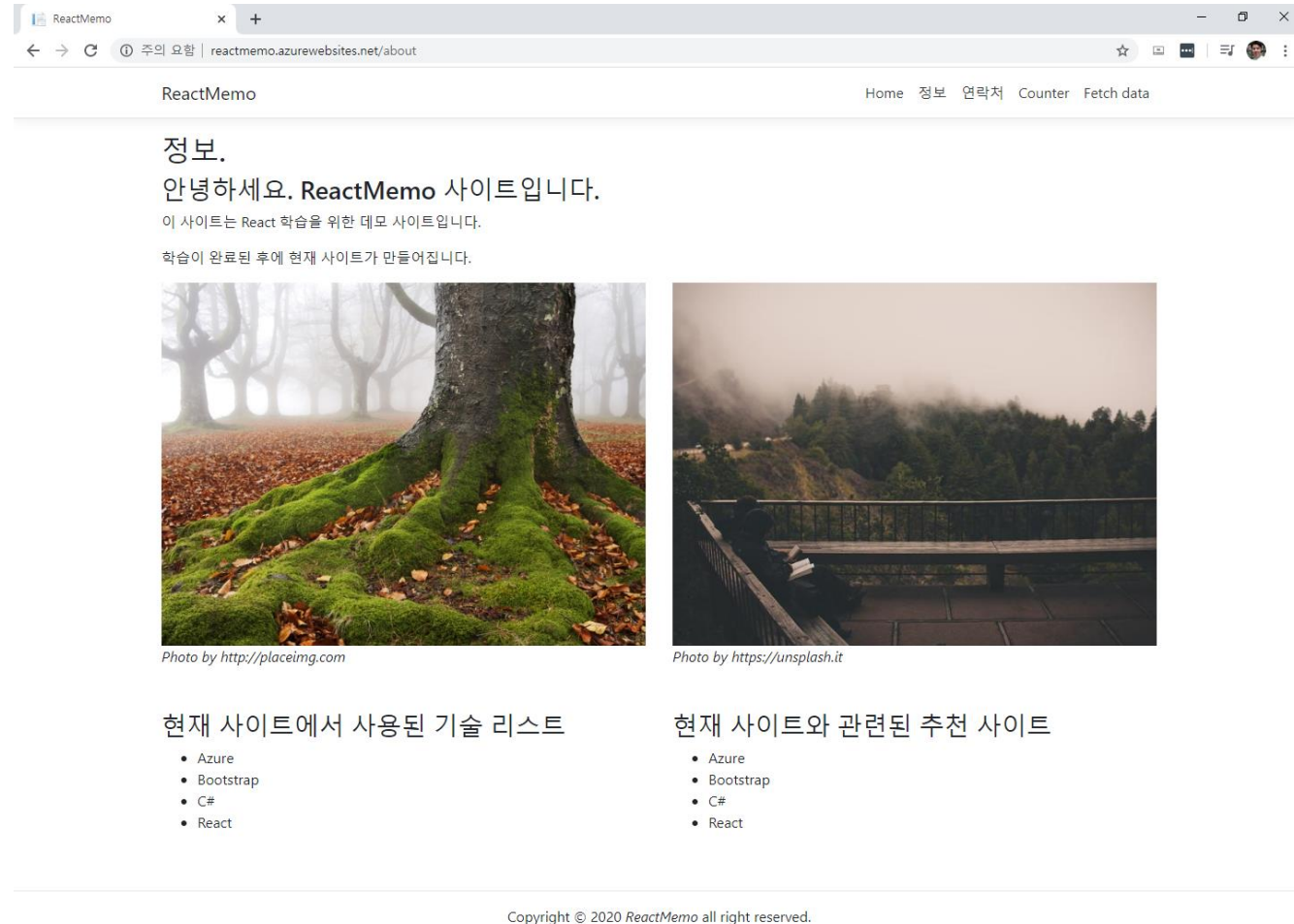
[■.....] - fetchMetadata: sill resolveWithNewModule scheduler@0.19.1 checking installable status
```

> npm start



강의 데모 사이트

- <https://reactmemo.azurewebsites.net>

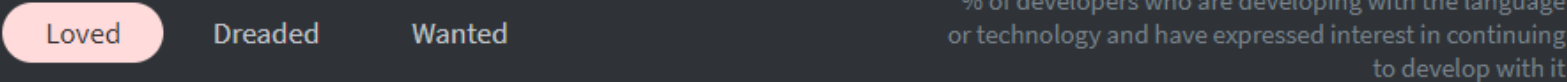


Most Loved, Dreaded, and Wanted Web Frameworks



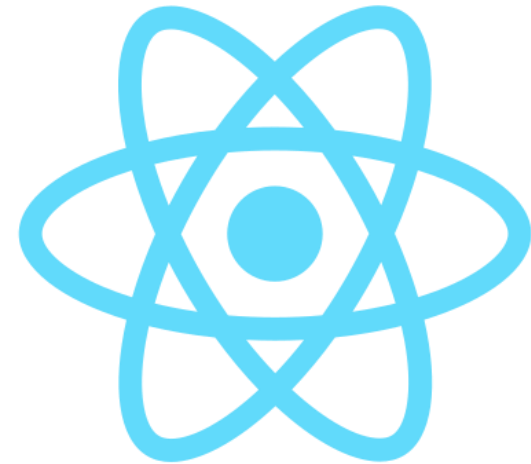
ASP.NET Core is the most loved web framework, beating out React.js. Gatsby, a newcomer on the survey, is already sitting at 5th, being loved by 60% of the respondents.

Although it is amongst the most popular web frameworks, Angular.js is also considered to be the most dreaded.

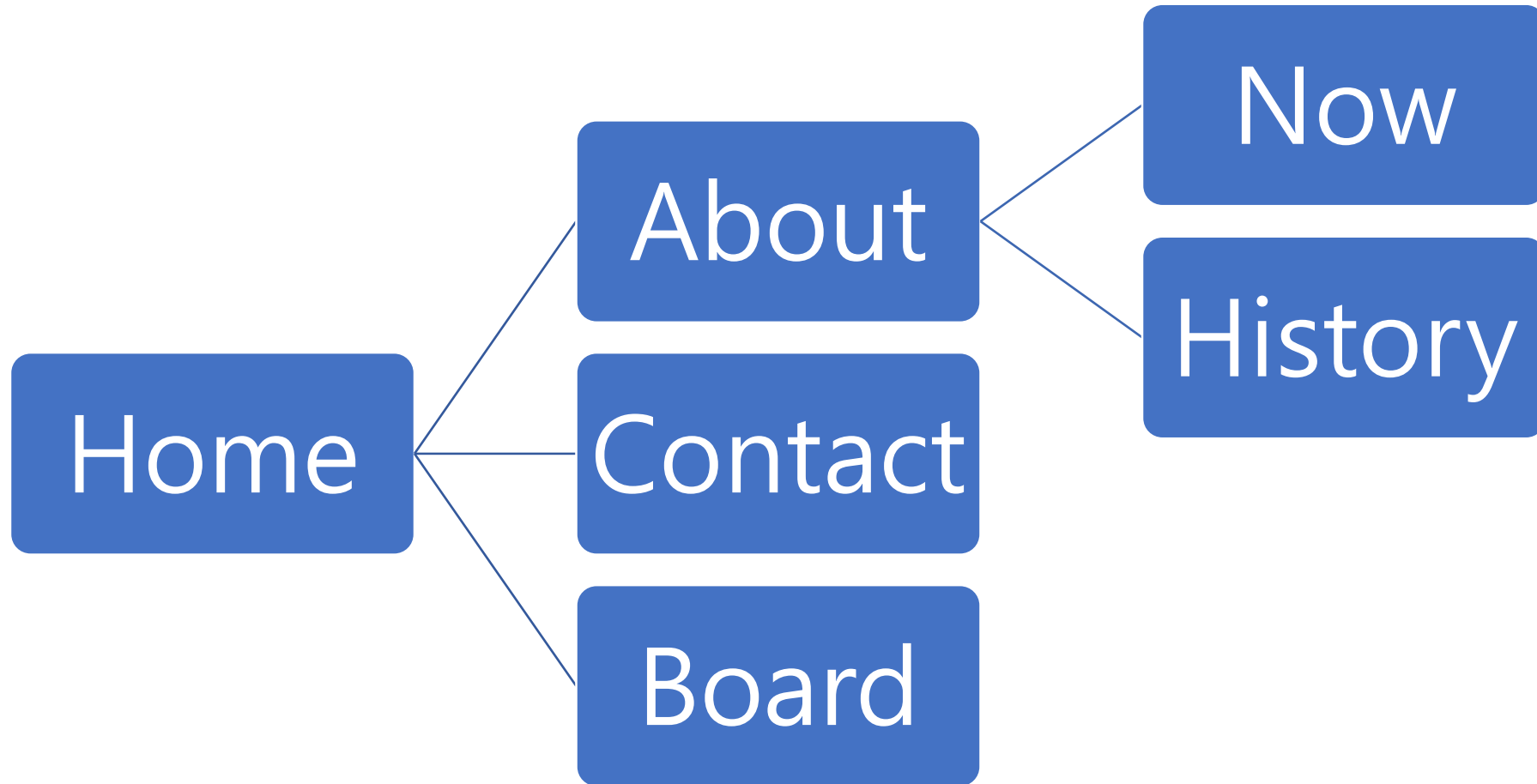


Single Page Application (SPA)

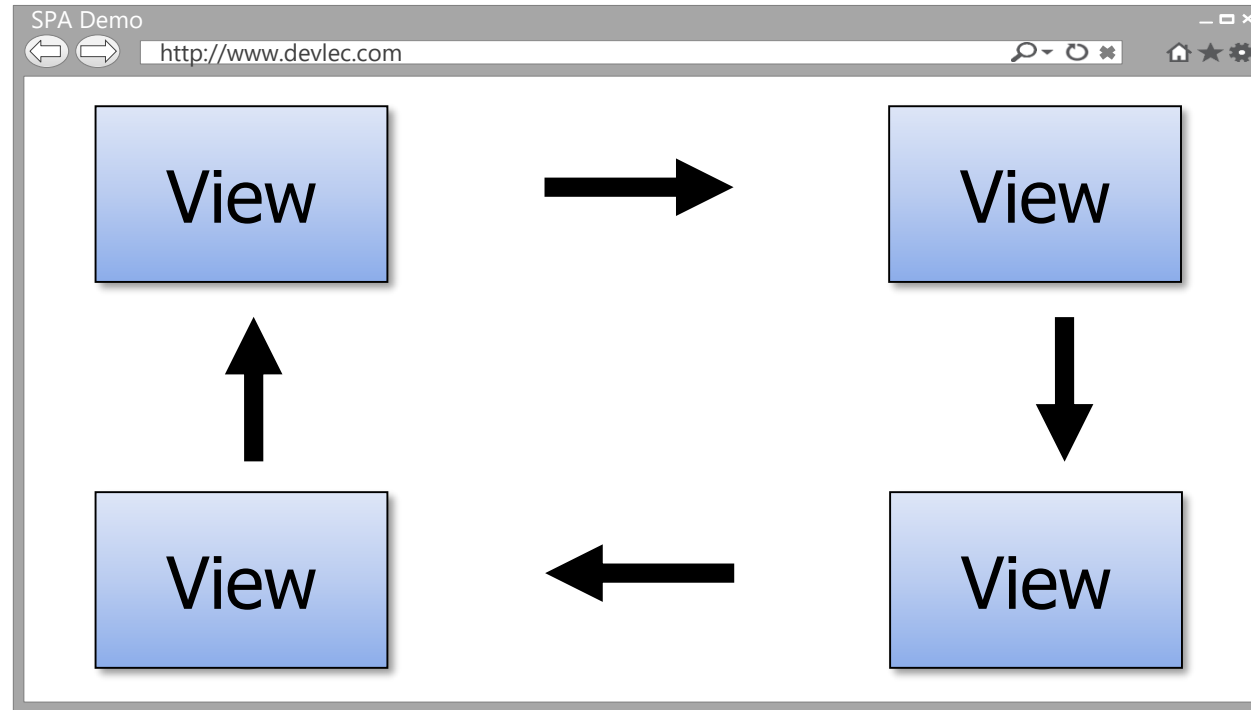
- SPA
 - Angular
 - Blazor
 - React



Multi Page Application에서 SPA로 이동

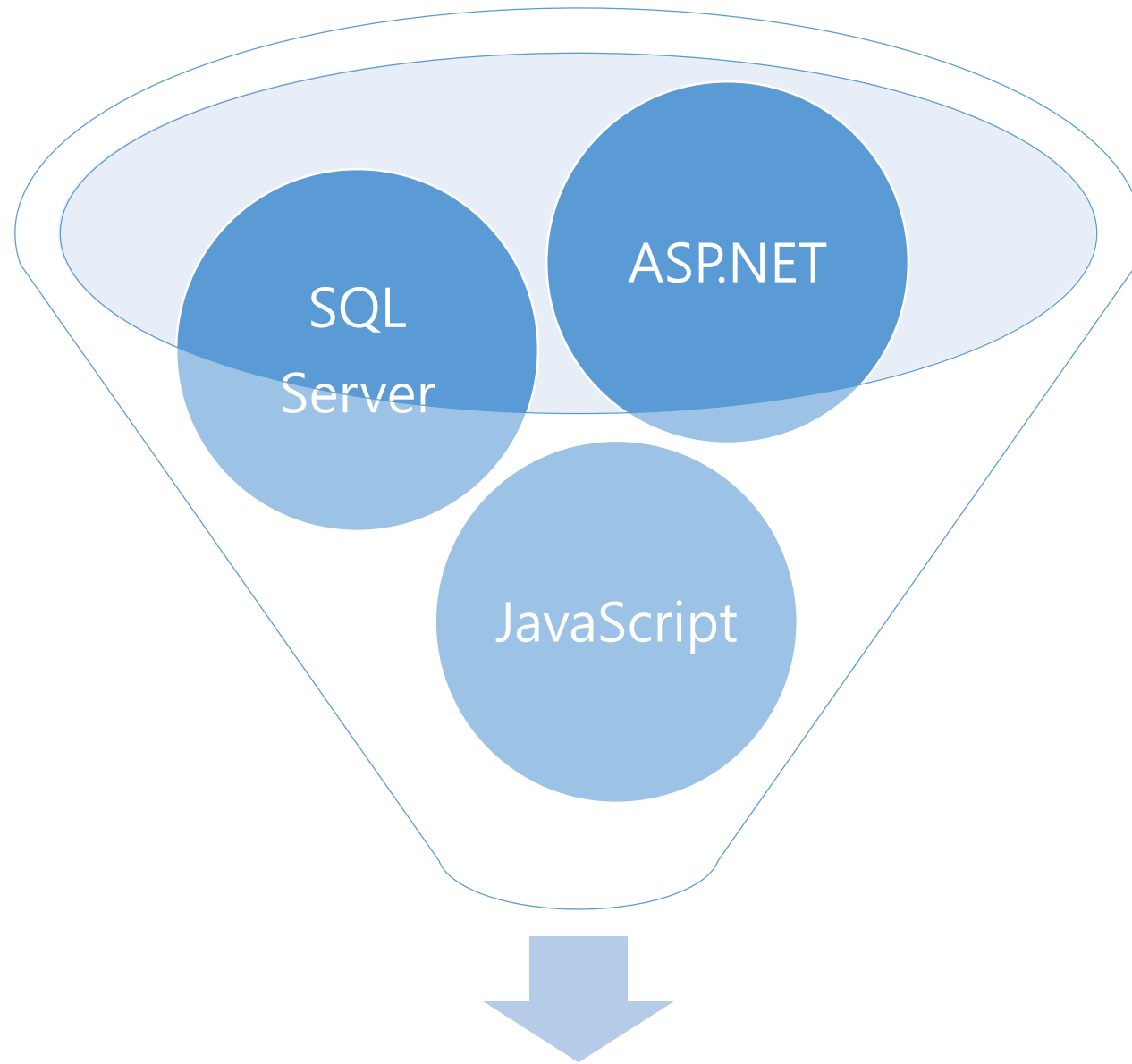


Single Page Application (SPA)



SPA 필요 없는 경우 React도 필요 없...

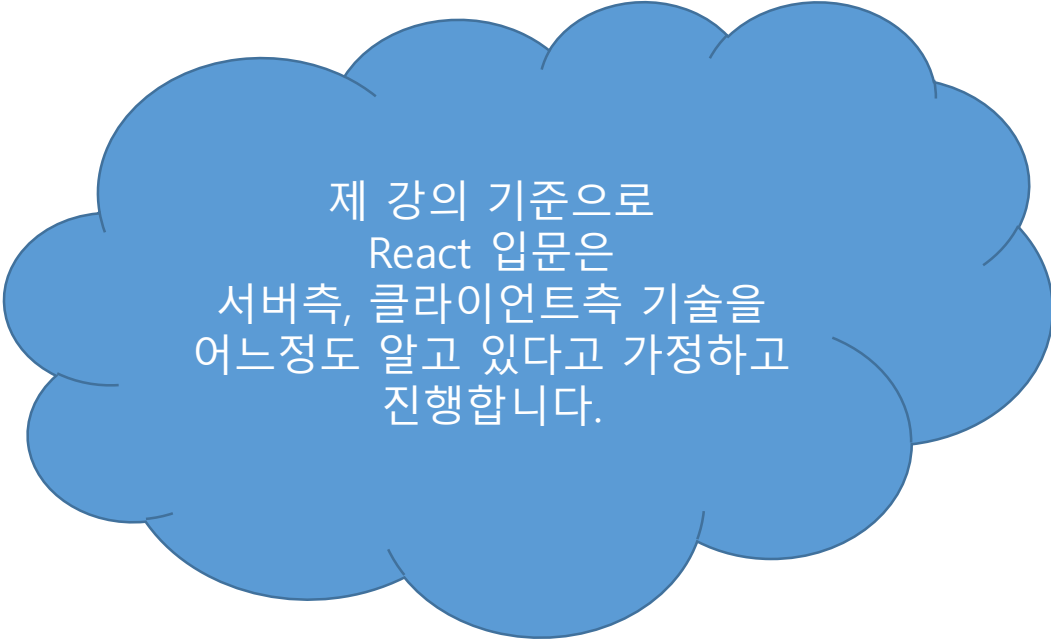




리액트 + ASP.NET Core 3.X

선수 학습(박용준 강사 강의 기준)

- 필수
 - Web
 - HTML5
 - CSS3
 - JavaScript
 - Bootstrap
 - ASP.NET Core Web API
 - 강의에서 따라하기로 진행



제 강의 기준으로
React 입문은
서버측, 클라이언트측 기술을
어느정도 알고 있다고 가정하고
진행합니다.

이 강의는 친절한 버전의 강의를 아닙니다. 이미 위와 같은 많은 기술을 알고 있다고 가정하고 진행합니다.

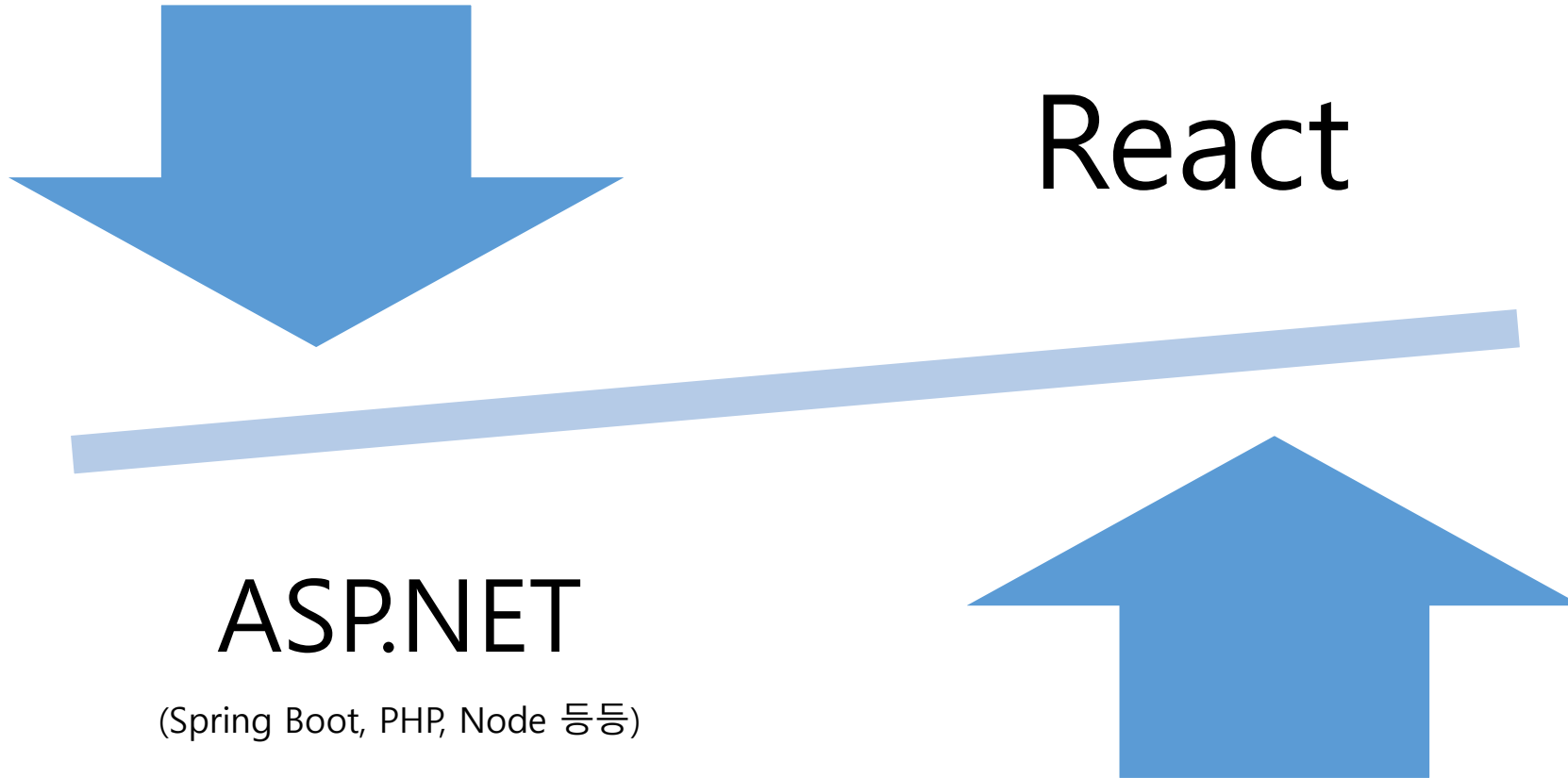
선수 학습 필수 과목(서버 측, 백 엔드)

- 백엔드: ASP.NET Core 기본
 - 컨트롤러
 - 액션
 - 모델 바인딩
 - Web API(GET, POST, PUT/PATCH, DELETE)
- 백엔드(서버 측) 기능은 강의에서 함께 다룹니다.

이 과정을 통해서 우리가 배울 내용

- 리액트기본
 - 컴포넌트
 - 폼
 - 라우팅
 - HTTP
 - 모듈
- 배포에 대해서는 따로 다루지 않습니다.
 - Visual Studio 2019에서 Azure Web App에 업로드하는 내용은 데모로 다룹니다.

클라이언트측과 서버측



리액트와 어울리는 서버 측 기술들

ASP.NET
Core

SQL Server

Node.js

MongoDB

ASP.NET Core Web API

- 최고의 RESTful 서비스 구축
- JSON
 - XML

강의 환경

- Visual Studio 2019 + ASP.NET Core 3.X SDK
 - <https://www.visualstudio.com>
- Visual Studio Code
- Postman
 - <https://www.getpostman.com>
- 웹 브라우저
- Node.js & NPM
- Git
- create-react-app

강의 소스

- <https://github.com/VisualAcademy/ReactMemo>
- <https://github.com/VisualAcademy/BookApp.React>

강의 시작합니다.



탐색기

▼ 열려 있는 편집기

▼ CLIENTAPP

> node_modules

▼ public

★ favicon.ico

<> index.html

{ } manifest.json

> src

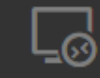
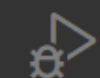
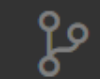
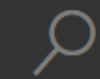
⚙ .env

📄 .gitignore

{ } package.json

{ } package-lock.json

ℹ README.md



> 개요

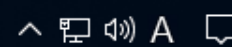
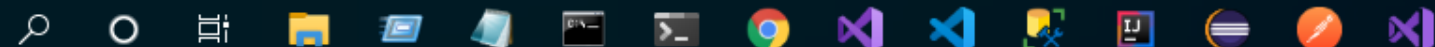
> 타임라인

> NPM 스크립트

> MAVEN PROJECTS



0 0



프로젝트 구조

- > public/index.html
- > src/index.js
- > src/App.js
- > src/components/Home.js

리액트 주요 개념

create-react-app

컴포넌트

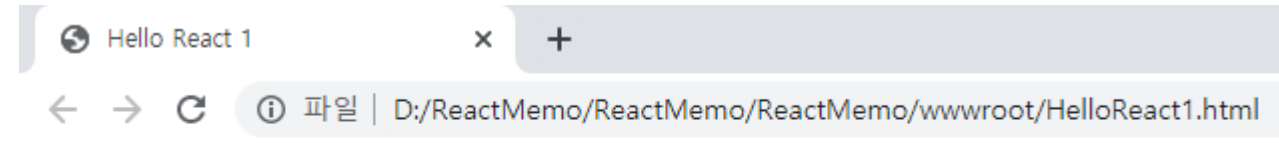
HTTP

이벤트

데이터 바인딩

순수 HTML 파일에 리액트 적용

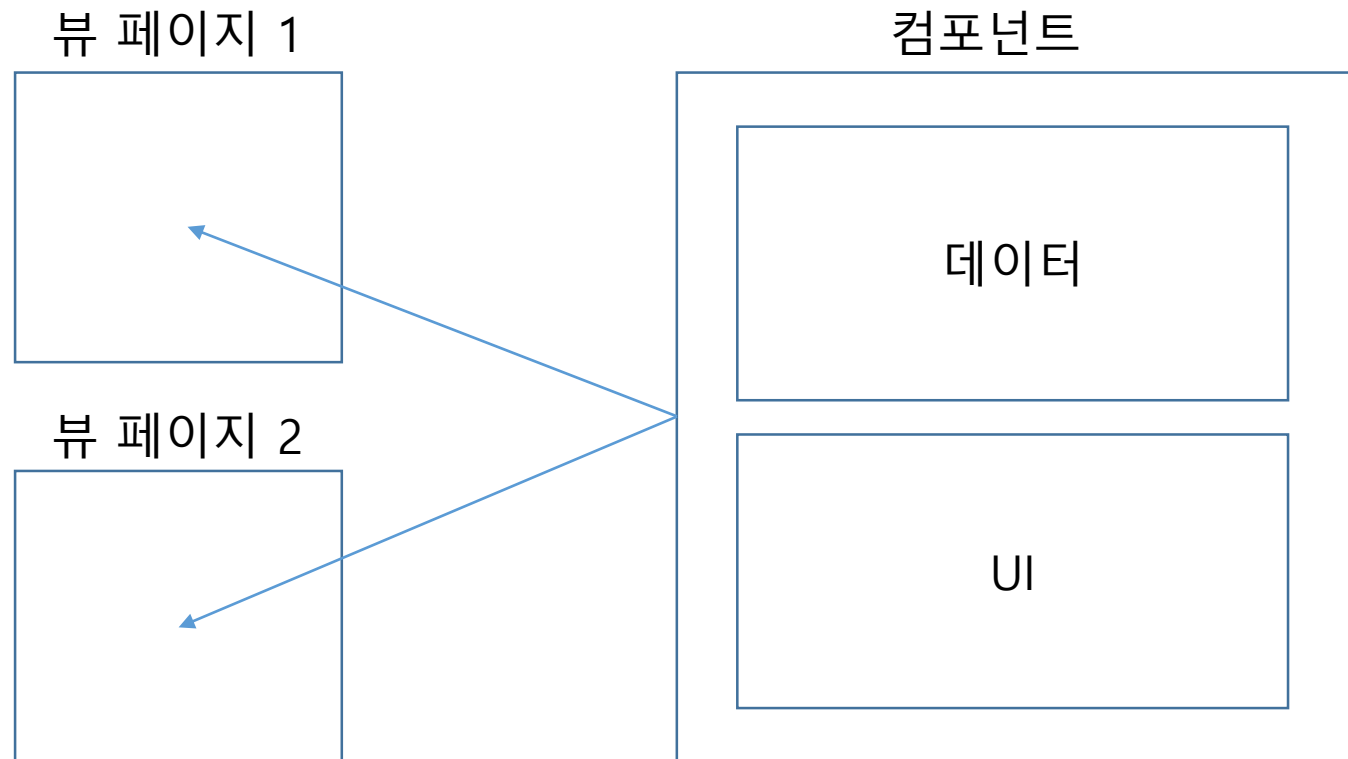
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Hello React 1</title>
  <script crossorigin src="https://unpkg.com/react@16/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
</head>
<body>
  <div id="root"></div>
  <script type="text/javascript">
    ReactDOM.render(React.createElement("h1", null, "안녕하세요."), document.getElementById("root"));
  </script>
</body>
</html>
```



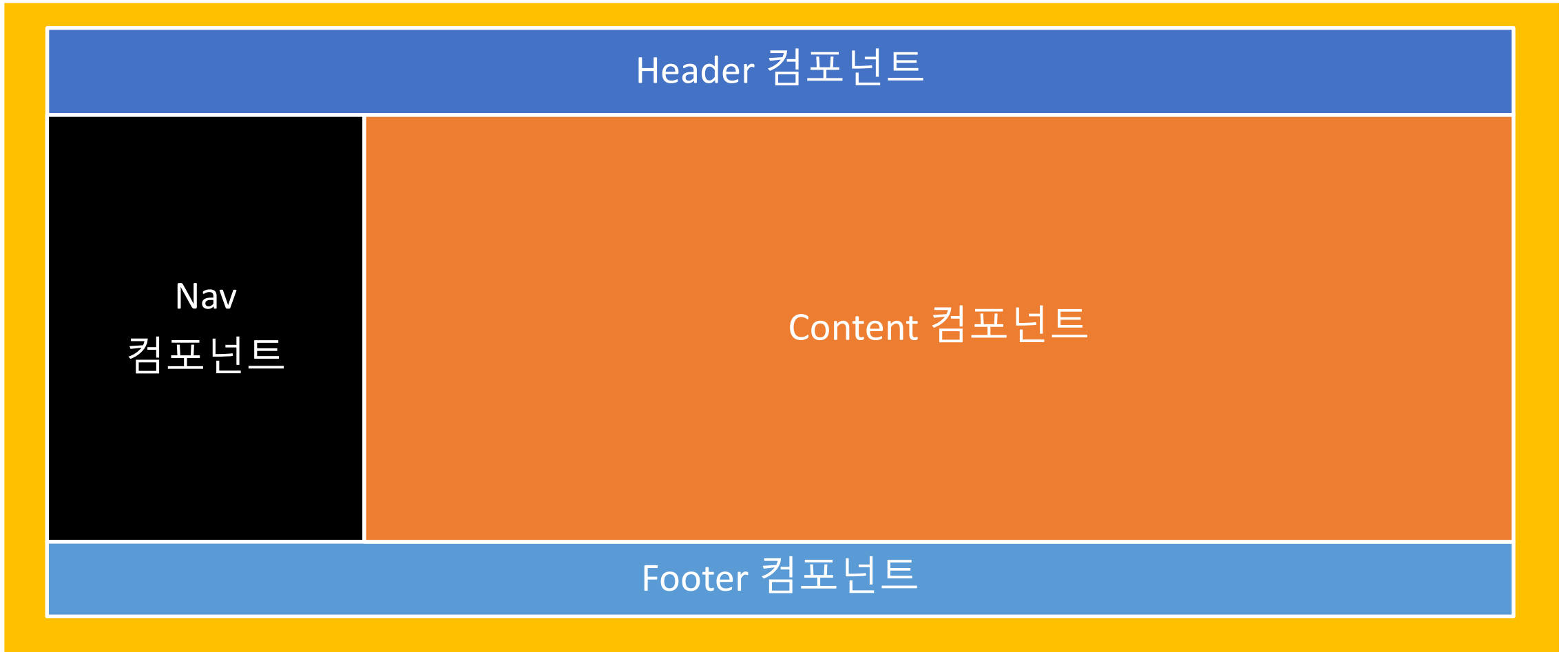
안녕하세요.

컴포넌트(Component)

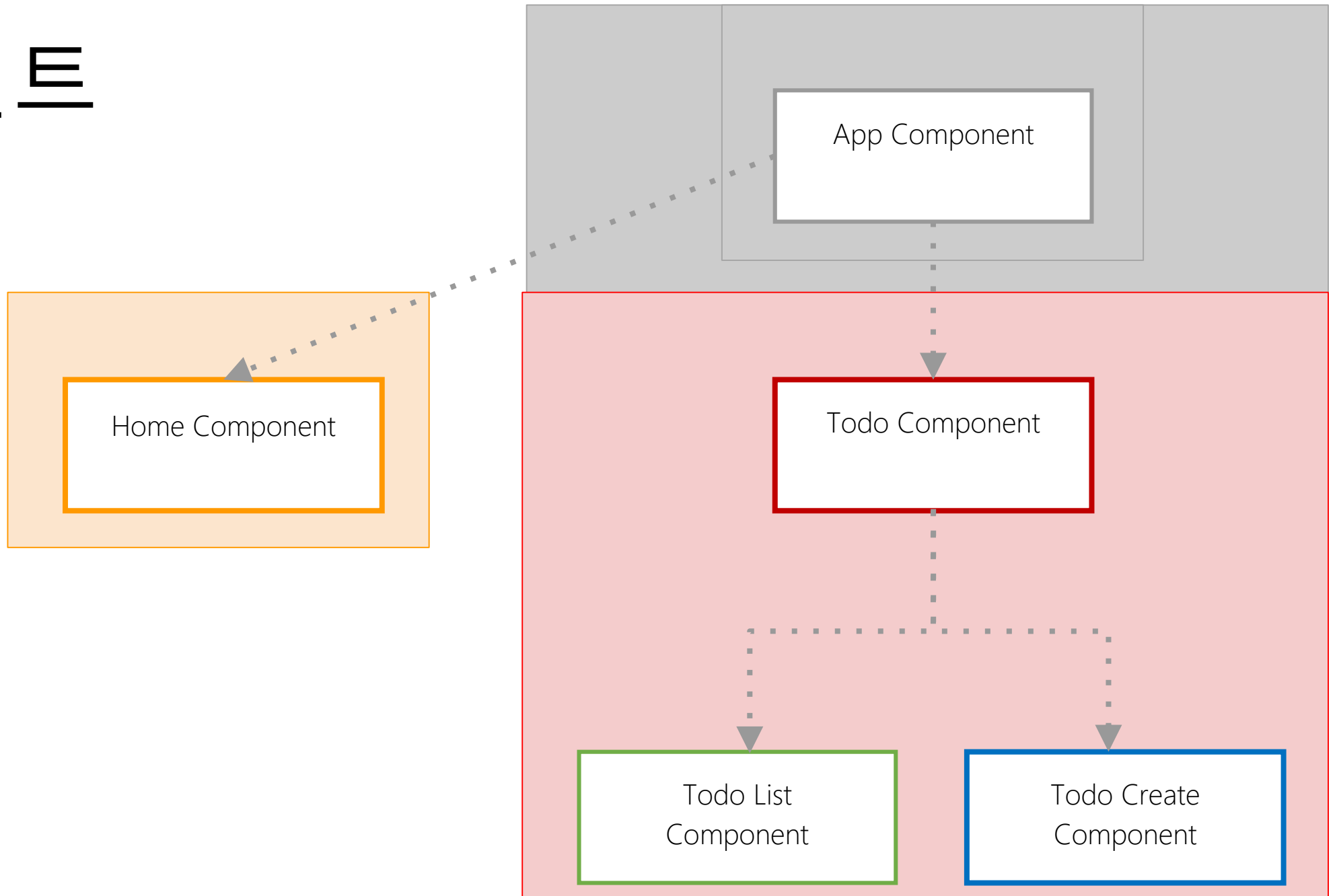
컴포넌트(Component): 뷰 컴포넌트



컴포넌트를 사용하여 앱 조립



컴포넌트



React 컴포넌트

- App 컴포넌트
- Child 컴포넌트

표현식(Expressions)

- 표현식(바인딩 표현식, 핸들바, 데이터 바인딩)
 - { } 로 사용됨
 - JavaScript의 모든 문법을 지원하지 않음
 - Subset
 - 배열 등 선언 가능
- 사용 예
 - { 표현식 }
 - { name }
 - { 3 % 5 }

```
{ expression }  
{ name }  
{ amount * 123 + 4 }
```


컴포넌트 종류

- 함수형 컴포넌트
 - 순수 자바스크립트 함수를 사용하여 컴포넌트 생성
- 클래스형 컴포넌트
 - ES6 class 키워드를 사용하여 컴포넌트 생성

ES6 모듈

JS

math.ts

×

```
export function sum(x, y)
  return x + y;
}
```

calculator.ts

×

```
import {sum} from './math';

sum(1, 2);
```

모듈 내보내기 및 가져오기

- > export

- > import

리액트 사용하기

표준 import 문

클래스

모듈

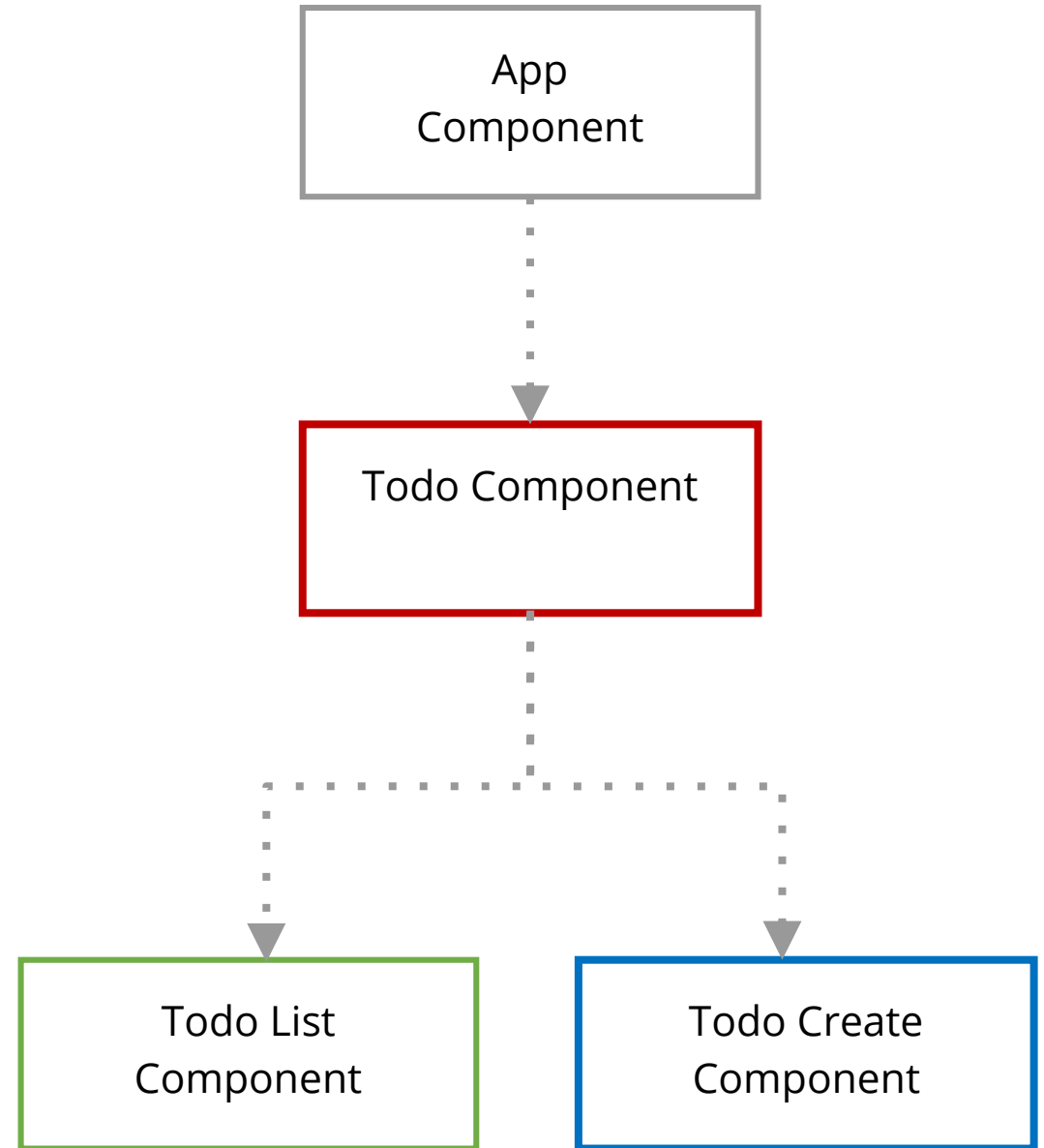
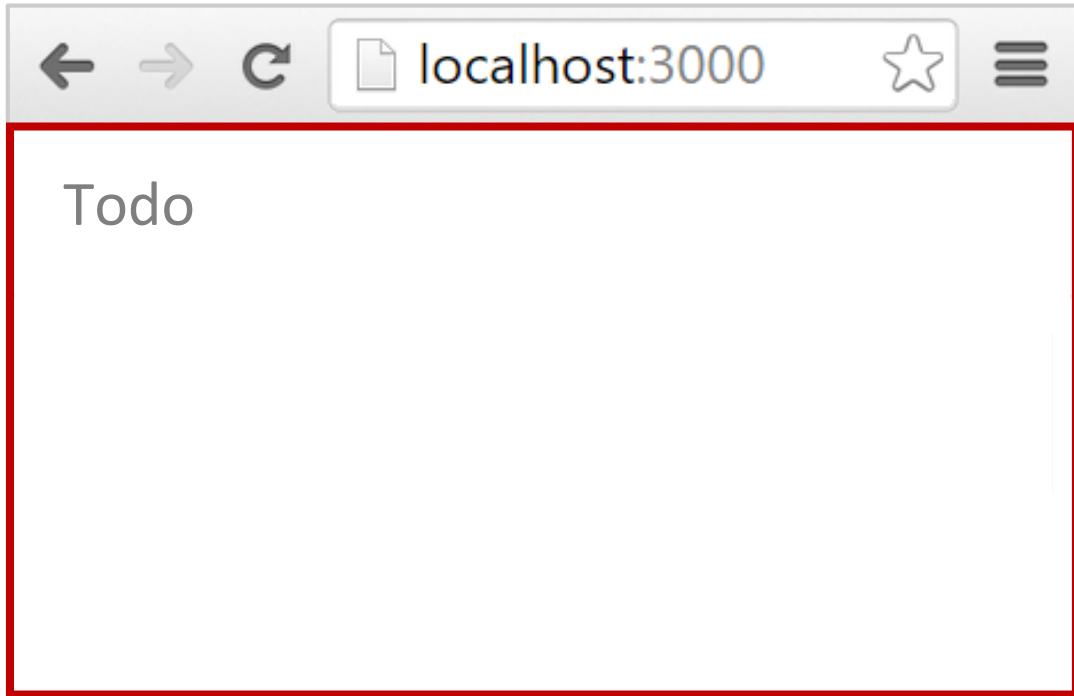
```
import React, { Component } from "react";
```

개체

모듈: export와 export default

- export
 - class MyComponent {}
 - export MyComponent;
 - import { MyComponent } from "모듈";
- export default
 - class MyComponent {}
 - export default MyComponent;
 - import MyComponent from "모듈";

컴포넌트 트리



<React.Fragment />

- 리액트 컴포넌트는 루트 요소를 무조건 하나 가져야 함
- <div> 태그도 좋지만 <Fragment>로 묶어주면 실제로 렌더링 될 때 추가 태그가 생성되지 않음
- 줄여서 <></>로 사용 권장

src > components > Shared > Copyright > JS Copyright.js > ...

```
1  import React, { Component } from "react";
2
3  export class Copyright extends Component {
4      render() {
5          const date = new Date();
6          const year = date.getFullYear();
7          return (
8              <div className="text-center">
9                  Copyright &copy; {year * 1} <em>ReactMemo</em> all right reserved.
10             </div>
11         );
12     }
13 }
```


컬렉션 리스트 출력하기

- `{this.state.links.map(link => <li key={link}>{link})}`

> map() 사용 예제

```
import React from 'react';

export class MapFunctionDemo extends React.Component {
  render() {

    const books = [
      { id: 1, title: 'C# 교과서' },
      { id: 2, title: 'ASP.NET Core를 다루는 기술' },
    ];

    return (
      <div>
        <h1>MapFunctionDemo</h1>
        {
          books.map((book) => (
            <div>{book.id} - {book.title}</div>
          ))
        }
      </div>
    );
  }
}
```

MapFunctionDemo

1 - C# 교과서

2 - ASP.NET Core를 다루는 기술

> map() 사용 예제: yield return 스타일

```
import React from 'react';

// yield return
const displayBookList = (bookList) => {
  return bookList.map((lst) => {
    return `${lst.title}: ${lst.price}원`;
  });
}

export class MapFunctionBookList extends React.Component {
  render() {
    const books = [
      { id: 1, title: 'C# 교과서', price: 30000 },
      { id: 2, title: 'ASP.NET Core를 다루는 기술', price: 55000 },
    ];

    const booksList = displayBookList(books);

    return (
      <div>
        <h1>MapFunctionDemo</h1>
        {booksList.map((lst) => (<div>{lst}</div>))}
      </div>
    );
  }
}
```

MapFunctionDemo

C# 교과서: 30000원

ASP.NET Core를 다루는 기술: 55000원

버튼에 인라인 스타일 적용하기

```
import React from 'react';
```

```
const btnStyles = {  
  color: 'white',  
  backgroundColor: 'green',  
  border: '1px solid red',  
};
```

```
function InlineStyle() {  
  return (  
    <div>  
      <button style={btnStyles}>버튼에 스타일 적용하기</button>  
    </div>  
  );  
}
```

```
export default InlineStyle;
```

버튼에 스타일 적용하기

{props.children}

- Props(Properties)에 전달된 RenderFragment

> props 제약 조건

```
static propTypes = {  
    title: PropTypes.string.isRequired  
}
```

이벤트 처리기: onClick=""

- onClick="함수 포인터"
 - onClick="함수 참조"

이벤트

- > componentDidMount()
- > componentWillReceiveProps(nextProps)
- > shouldComponentUpdate()
 - > shouldComponentUpdate(nextProps, nextState)
- > componentWillUnmount()
- > getDerivedStateFromProps(props, state)
- > getSnapshotBeforeUpdate()

상태 관리: state = { count: 1 };

- {this.state.count}

상태 관리: state

```
// StateDemo.js
import React, { Component } from 'react';

export class StateDemo extends Component {
  constructor(props) {
    super(props);
    this.state = {
      name: "[1] 처음 상태값 로드"
    };
  }

  render() {
    return (
      <div>
        <h1>상태값: {this.state.name}</h1>
        <button onClick={
          () => {
            this.setState({ name: '[2] 새로운 상태값으로 변경' });
          }
        }>변경
      </button>
    </div>
    );
  }
}
```

상태값: [1] 처음 상태값 로드

변경

상태값: [2] 새로운 상태값으로 변경

변경

상태 관리: state

```
// StateObjectLiteral.js
import React, { Component } from 'react';

export class StateObjectLiteral extends Component {
  state = {
    message: "[1] 처음 상태값 로드",
    title: "상태값",
  };
  render() {
    const { title, message } = this.state;
    return (
      <div>
        <h1>{title}: {message}</h1>
        <button onClick={
          () => {
            this.setState({ message: '[2] 새로운 상태값으로 변경' });
          }
        }>변경
      </button>
    </div>
    );
  }
}
```

상태값: [1] 처음 상태값 로드

변경

상태값: [2] 새로운 상태값으로 변경

변경

이벤트: onClick

```
import React, { Component } from 'react';

export class OnClickDemo extends Component {
  render() {
    return <button onClick={
      () => { window.alert('클릭이벤트 발생'); }
    }>클릭테스트</button>
  }
}
```

클릭테스트

localhost:44345 내용:

클릭이벤트 발생

확인

하나 이상의 경로 지정: React.js

```
<Route exact path="/" component={Home} />  
<Route path={['/Books', '/Books/Index']} component={BooksIndex} />  
<Route path="/counter" component={Counter} />
```

동일 경로 해제: exact={true}

```
<Route exact path="/" component={Home} />  
<Route path={['/Books', '/Books/Index']} component={BooksIndex} exact="true" />  
<Route path={['/Books/Create', '/Books/Create:Id']} component={BooksCreate} />
```

라이프 사이클: componentDidMount

- 컴포넌트가 렌더링된 후 최종 실행

특정 라우팅으로 이동

```
goCreatePage() {  
  //console.log("Create");  
  const { history } = this.props;  
  history.push('/Books/Create');  
}  
  
editBy(id) {  
  //console.log("Edit by " + id);  
  const { history } = this.props;  
  history.push("/Books/Edit/" + id);  
}  
  
deleteBy(id) {  
  //console.log("Delete by " + id);  
  const { history } = this.props;  
  history.push("/Books/Delete/" + id);  
}
```

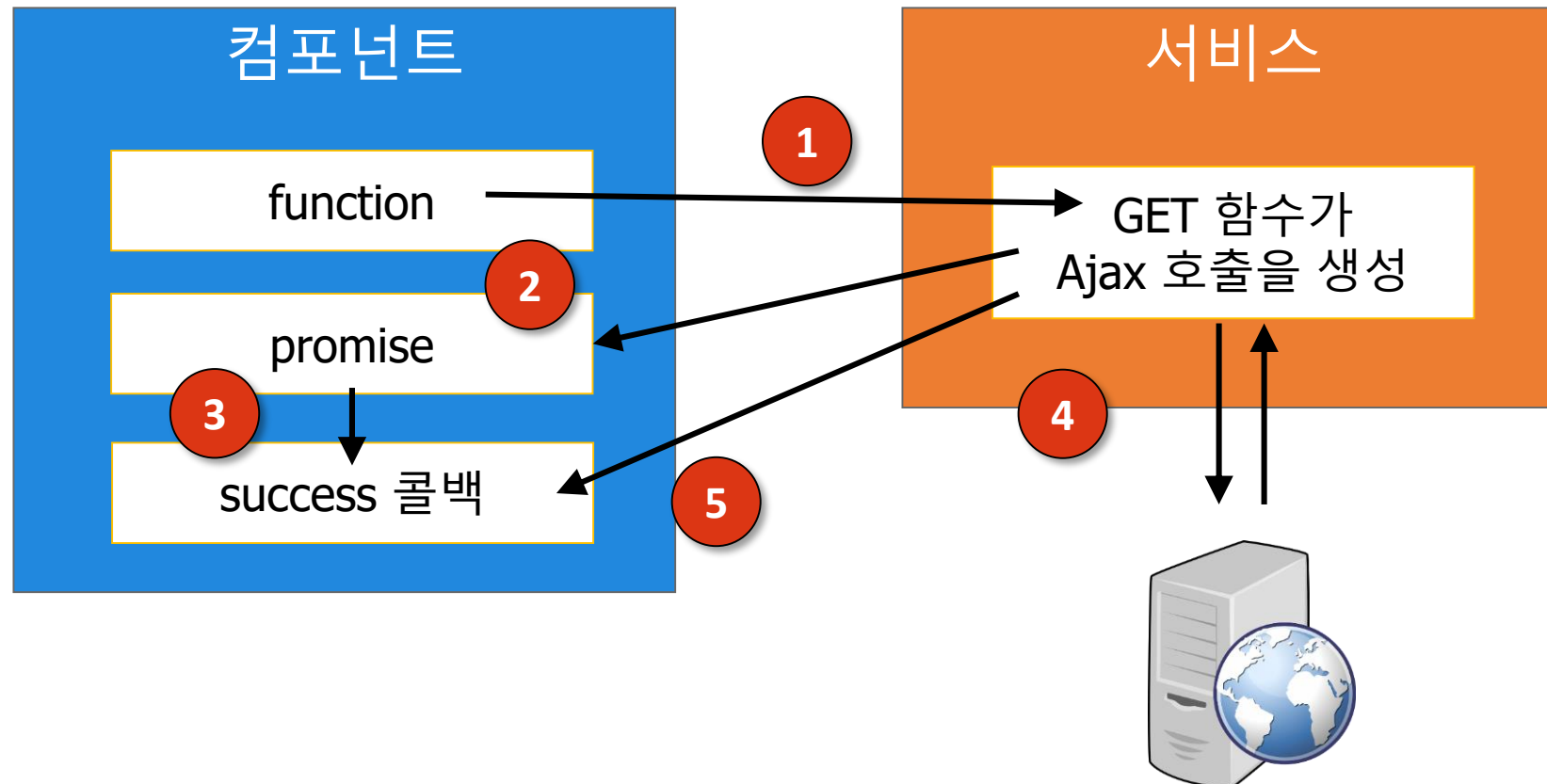

Http와 비동기

Promise를 사용하여 데이터를 주고, 받기

비동기

`fetch()` or `axios.get()` => `then()`

Promise



React와 타입스크립트

ES5

ES6/ES2015+

TypeScript

참고자료

- 리액트 공식 사이트

마무리

- React 기반의 웹 앱 제작

