

03 웹 서비스 프론트엔드 개발

3주차 36시간 (React)

김범준

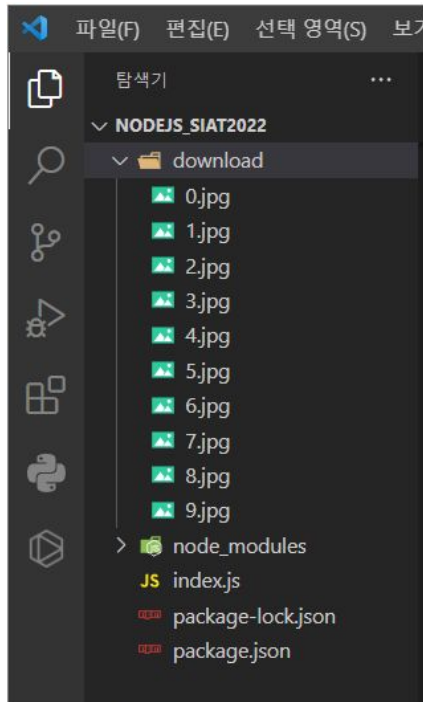
1. 강의정보

과정개요	OpenAPI와 React.js를 이해하고, 기존 웹사이트 복제(클로닝) 실습 진행
학습목표	<ul style="list-style-type: none">- OpenAPI의 이해- React.js의 이해
핵심역량	OpenAPI의 개념 및 활용사례 이해 React.js 개념 이해 및 배포 및 컴포넌트 이해
SW융합요소	<input checked="" type="checkbox"/> 이론 <input checked="" type="checkbox"/> 실습 <input type="checkbox"/> 토론 <input type="checkbox"/> 사례분석

2. 학습내용			
주제	강의내용	강의방법	시간
OpenAPI의 이해	<OpenAPI> - OpenAPI 개념의 이해 - OpenAPI 활용사례	이론강의	
	<OpenAPI 활용> - OpenAPI 활용하기		
React.js의 이해	<React.js> - 개발 환경 - Create React App 설치 - 샘플 웹앱 실행하기	이론강의	
	<React.js> - 배포하는 법 - 컴포넌트 만들기 - props		
	<React.js> - React Developer Tools - state - 이벤트 state props와 render 함수		
	<React.js> - 컴포넌트 이벤트 만들기 - 베이스 캠프		
	<React.js> - 웹사이트 Create 기능 구현 - 웹사이트 Update 기능 구현 - 웹사이트 Delete 기능 구현		
과정실습	<과정실습 진행> Weekly project : Web site cloning 2	이론 및 실습강의	
강의시간(h)			36

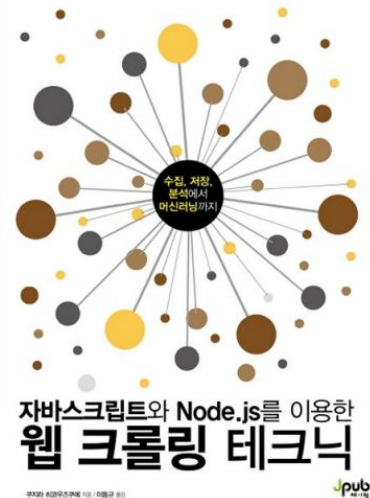
OpenAPI의 이해

Node.js 웹 크롤링



실행 프로젝트에 파일 다운로드

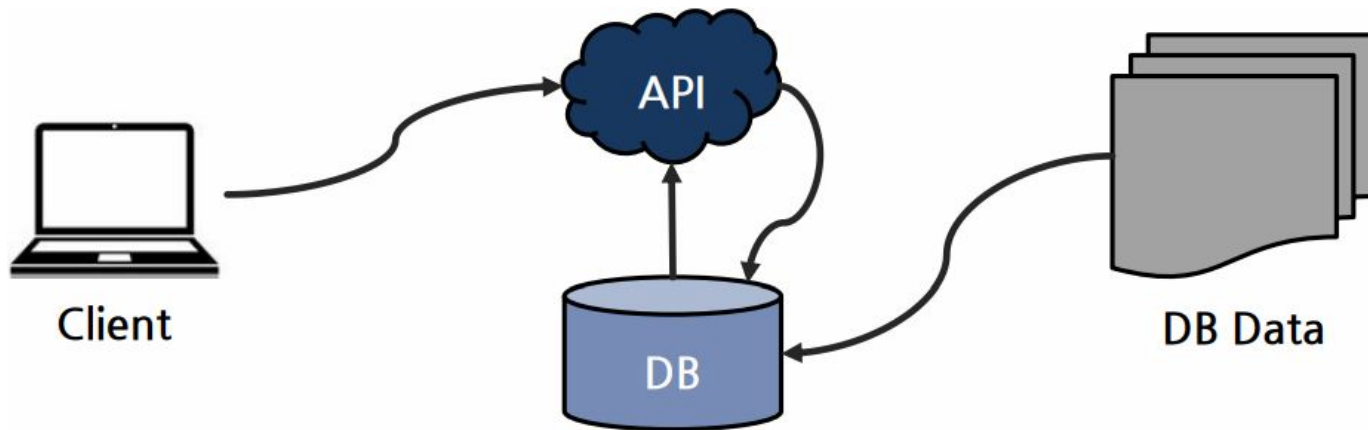
```
app.get('/axios', (req, res) => {  
  // Promise - 콜백 헬에 빠지는것을 방지(흐름제어) - 메소드체인 .then([콜백])  
  // Async - 리스트 형식으로 한다. [콜백, 콜백, 콜백 ...]  
  let getUrlVal = "https://news.naver.com/main/main.naver?mode=LSD&mid=shm&sid1=100";  
  axios.get(getUrlVal, {responseType:"arraybuffer"})  
    .then( (response) => {  
      const htmlContent = response.data  
      let htmlCMD = iconv.decode(htmlContent, "EUC-KR").toString();  
      // cheerio를 이용한 DOM셀렉터  
      const $ = cheerio.load(htmlCMD);  
      //div.cluster_text > a  
      let h1Data = $('div.cluster_body div.cluster_text a').text();  
      console.log(h1Data.trim());  
    });  
  res.end();  
});
```



추천 도서

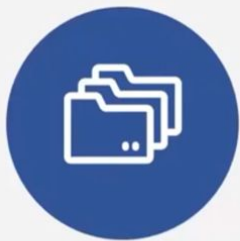
OpenAPI 개념 이해

- API (Application Programming interface)
 - 특정 프로그램을 만들기 위해 제공되는 모듈(함수 or 클래스)

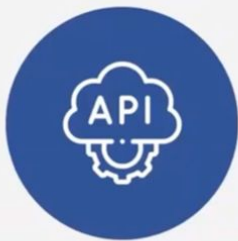


공공데이터 Open API

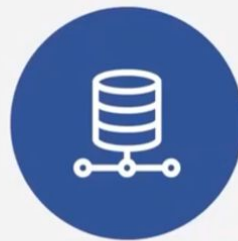
공공데이터란?



File Data



Open API



Standard Data Set

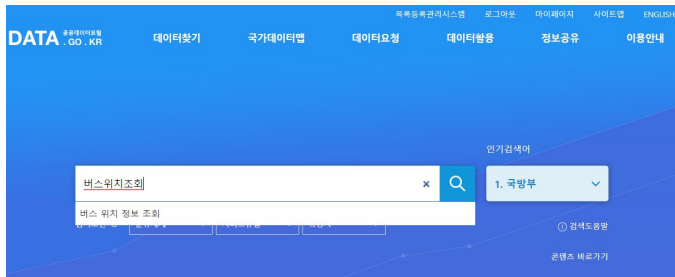
공공데이터란 공공기관이 만들어내는 모든 자료나 정보, 국민 모두의 소통과 협력을 이끌어내는 공적인 정보를 말합니다. 각 공공기관이 보유한 공공데이터 목록과 국민에게 개방할 수 있는 공공데이터를 포털에 등록하면 모두가 공유할 수 있는 양질의 공공데이터로 재탄생하게 됩니다.

OpenAPI 개념 이해

- Open API
 - 누구나 사용할 수 있도록 공개된 API
 - 주로 Rest API 기술을 많이 사용
- RestAPI
 - Representational State Transfer API의 약자
 - HTTP 프로토콜을 통해서 정보를 제공하는 함수
 - 실질적인 API 사용은 정해진 구조의 URL 문자열 사용
- 일반적으로 XML, JSON의 형태로 응답 출력

Open API를 활용한 공공 DB 수집

- 서울시 버스 정보와 버스 위치 정보
 - [공공데이터포털 \(data.go.kr\)](https://www.data.go.kr) - <https://www.data.go.kr>
 - 공공데이터 포털 사이트에 로그인
 - Open API 활용 가이드 제공 (문서 참조 구현)



사이트 메인 검색 창에서 원하는 데이터 검색

국민과 함께 하는 공공데이터포털에 오신 것을 환영합니다.

네이버 or 카카오 로그인 가능

Open API를 활용한 공공 DB 수집

- 원하는 데이터를 검색 해서 오픈 API를 찾을 수 있다.

전체(48,581건)

파일데이터(33,107건)

오픈 API(7,055건)

표준데이터셋147개(8,419건)

정확도순 ▼ 10개씩 ▼

오픈 API (7,055건)

교통물류

자치행정기관

미리보기

XML

JSON

서울특별시_노선정보조회 서비스

노선에 대한 정보 제공

제공기관 서울특별시 수정일 2021-11-16 조회수 19893 활용신청 13552 키워드 서울시,노선정보조회,버스노선

활용신청

교통물류

자치행정기관

미리보기

XML

JSON

서울특별시_버스위치정보조회 서비스

실시간 버스위치 정보 제공

제공기관 서울특별시 수정일 2021-11-16 조회수 18218 활용신청 13044 키워드 서울특별시,버스위치,버스위치정보

활용신청

교통물류

자치행정기관

미리보기

XML

JSON

서울특별시_버스도착정보조회 서비스

특정 정류소에 대한 버스 도착예정 정보 제공

제공기관 서울특별시 수정일 2021-11-16 조회수 17361 활용신청 5775 키워드 정류소 도착예정정보,저상버스 도착예정정보,노선의 도착예정정보

활용신청

Open API를 활용한 공공 DB 수집

- 참고문서에서 서비스 활용 가이드 다운로드

오픈API 상세



XML **JSON** 서울특별시_노선정보조회 서비스

노선에 대한 정보 제공

2

0

관심

활용신청

오류신고 및

담당자 문의

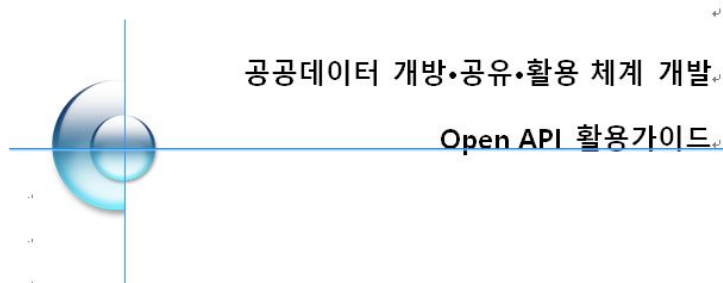
OpenAPI 정보

메타데이터 다운로드

분류체계	교통및물류 - 도로	제공기관	서울특별시
관리부서명	교통정보과	관리부서 전화번호	02-2133-4969
API 유형	REST	데이터포맷	JSON+XML
활용신청	13552	키워드	서울시,노선정보조회,버스노선
등록	2011-12-03	수정	2021-11-16
비용부과유무	무료	신청가능 트래픽	1,000 / 운영계정은 활용사례 등록시 신청하면 트래픽 증가 가능
심의유형	개발단계 : 자동승인 / 운영단계 : 자동승인		
이용허락범위	 출처표시		
참고문서	서울특별시 노선정보조회 서비스 활용가이드 20211116.docx		

Open API를 활용한 공공 DB 수집

- 참고문서에서 서비스 활용 가이드 다운로드



1. API 서비스 명세	3
1.1 버스위치정보조회 서비스	3
가. API 서비스 개요	3
나. 상세기능 목록	4
다. 오류코드 안내	20

Open API를 활용한 공공 DB 수집

- 상세 기능 탭에서 "활용신청" 버튼

상세기능

활용사례

추천데이터

상세기능

목록

getStationsByRouteList

조회

노선별 경유 정류소 조회 서비스

· 활용승인 절차 개발단계 : 자동승인 / 운영단계 : 자동승인

· 신청가능 트래픽 1,000 / 운영계정은 활용사례 등록시 신청하면 트래픽 증가 가능

· 요청주소

· 서비스URL http://ws.bus.go.kr/api/rest/busRouteInfo/getStationByRoute

활용신청

서울특별시	서비스유형	REST
자동승인	신청유형	개발계정 활용신청
신청	활용기간	승인일로부터 24개월 간 활용가능

공공데이터 제공제도

- * 공공데이터중 위치정보를 포함한 서비스를 사용하고자 하는 사업자는 '위치정보의 보호 및 이용 등에 관한 법률'에 따라 방송통신위원회에 '위치정보서비스 허가'를 받거나 '위치기반 서비스사업 신고'를 하여야 합니다.
- * 이에 해당하는 사업자인 경우에는 첨부파일에 '위치기반서비스사업신고필증'을 첨부해 주시기 바랍니다.
- * 활용신청 시 '위치기반서비스사업신고필증'이 등록되지 않으면 반려가 될 수 있으니 참고 하시기 바랍니다.

활용목적 선택

*표시는 필수 입력항목입니다.

* 활용목적

☒ 웹 사이트 개발 ☐ 앱개발 (모바일,솔루션등) ☐ 기타 ☐ 참고자료 ☐ 연구(논문 등)

OpenAPI 활용 사례

- 포털사이트 블로그 글 검색
 - 포털사이트 블로그 검색 결과를 출력 해주는 REST API

N****	Developers	Products	Documents	Application	Support
Document > 서비스 API > 검색 > 블로그					
검색 > 블로그					
API 공통 가이드 SDK&Tools Clova 아이디로 로그인 번역기		*** 블로그 검색 결과를 출력해주는 REST API입니다. 비로그인 오픈 API이므로 GET으로 호출할 때 HTTP Header에 애플리케이션 등록 시 발급받은 Client ID 와 Client Secret 값을 같이 전송해 주시면 활용 가능합니다.			
		오픈 API 이용 신청 >			

Danfo.js

- <https://danfo.jsdata.org/>
 - 자바스크립트로 데이터 분석이 가능한 라이브러리

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://cdn.jsdelivr.net/npm/danfojs@1.1.0/lib/bundle.min.js"></script>
  <title>Document</title>
</head>

<body>

  <script>

    s = new dfd.Series([1, 3, 5, undefined, 6, 8])
    s.print()

  </script>

</body>

</html>
```



```
1 //output
```

2		
3		0
4		
5	0	1
6		
7	1	3
8		
9	2	5
10		
11	3	undefined
12		
13	4	6
14		
15	5	8
16		

Chart.js

- <https://www.chartjs.org/>
 - 자바스크립트를 이용해서 화면에 그래프를 그리는 라이브러리



과정 실습

Open API

- 활용 가능한 Open API 조사하기
 - 네이버 Open API
 - 기상청 동네예보
 - 전국 유료 관광지 입장객 정보
 - 페이스북 데이터 수집
 - 치킨 매장 정보 등.

Open API의 활용

OpenAPI 활용하기

- 포털 사이트 뉴스 글 읽어오기
 - 포털 사이트 뉴스 검색 결과를 출력 해주는 REST API

N****	Developers	Products	Documents	Application Support
Document > 서비스 API > 검색 > 뉴스				
API 공통 가이드 SDK&Tools Clova 아이디로 로그인 번역기 서비스 ADI		검색 > 뉴스 *** 뉴스 검색 결과를 출력해주는 REST API입니다. 비로그인 오픈 API이므로 GET 으로 호출할 때 HTTP Header에 애플리케이션 등록 시 발급받은 Client ID 와 Client Secret 값을 같이 전송해 주시면 활용 가능합니다. 오픈 API 이용 신청 >		

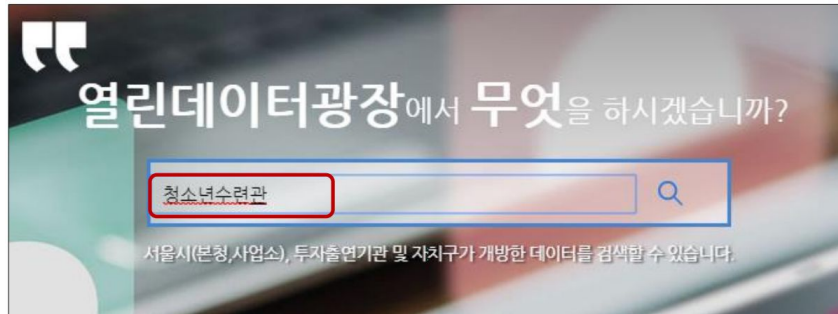
과정 실습

Open API 활용 실습

- 서울열린데이터광장에서 서울시 청소년수련관 강좌 정보 읽어오기



[출처 : 서울열린데이터광장]



[출처 : 서울열린데이터광장]

서울시 청소년수련관 강좌정보 ★★★★★			
분류 보기	원본시스템	업데이트	제작일자
담당자	담당부서	데이터공개일자	데이터수정일자
담당자 연락처	담당부서 연락처	데이터공개일자	데이터수정일자
서울시 청소년수련관에서 시행중인 강좌정보입니다.			
Sheet			
Open Api			
서울시 청소년수련관 강좌정보			
http://openapi.seoul.go.kr:8088/1인용P10xml/LampScgmdr/1/5/			

[출처 : 서울열린데이터광장]

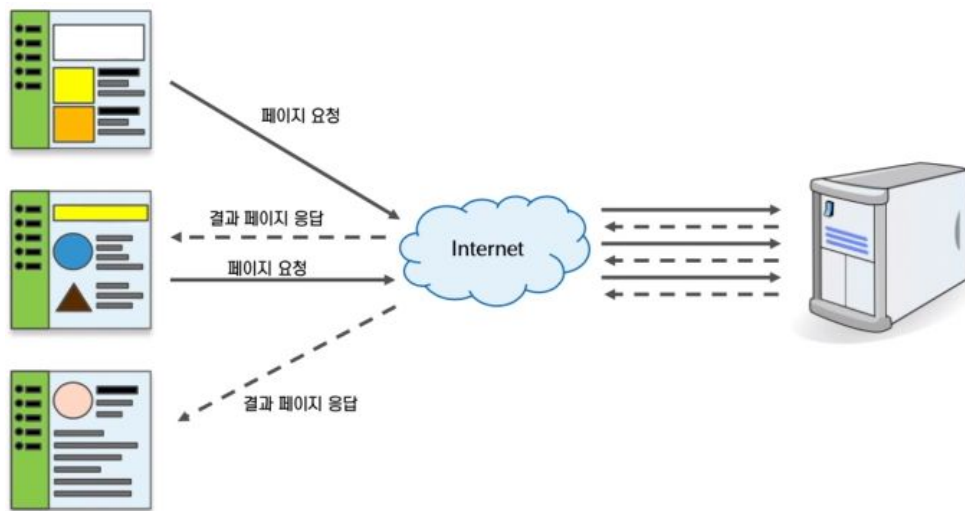
서비스(사유) 환경	<input checked="" type="radio"/> 웹 사이트 개발 <input type="radio"/> 앱 개발 (모바일 솔루션 등) <input type="radio"/> 연구 (논문 등) <input type="radio"/> 기타 참고자료
사용 URL (150자 이내)	<input type="text"/>
관리용 단체 / 기업 / 기관	<input type="text"/> <input type="text"/> <input type="text"/>
표 이메일	제공하여 주신 담당자 정보는 API종단, 서버점검 등에 대한 공지 및 정책 변경에 따른 안내를 위해 사용됩니다.
활용용도	<input type="text"/>
내용	<input type="text"/>

[출처 : 서울열린데이터광장]

React.js 설치

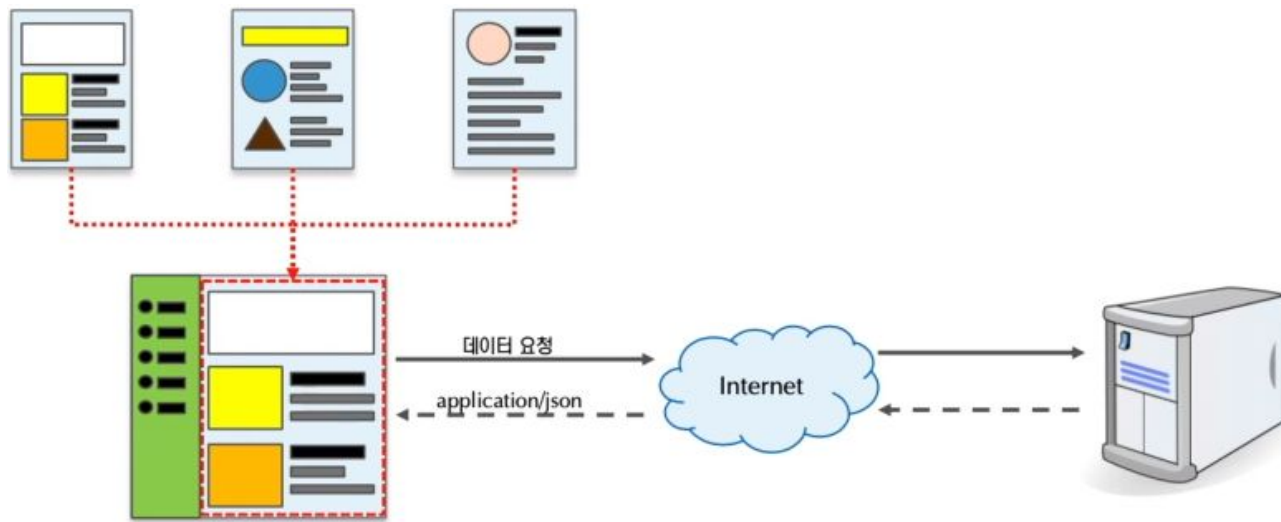
MPA vs SPA

- MPA(Multi-Page Application)는 클라이언트의 요청에 따라 페이지 생성 반환.
- 클라이언트가 요청하는 모든 페이지가 서버에 존재
- 서버가 모든 페이지를 생성하고 응답
- 페이지의 작은 변화에도 항상 **Redirect**.



MPA vs SPA

- SPA(Single-Page Application)는 하나의 페이지에서 구동된다.
- 필요한 데이터는 Ajax등의 기술을 이용해서 서버에서 JSON 형식으로 제공 받는다.
- 클라이언트에서 매번 전체 화면이 모두 갱신 되지 않고 필요한 부분의 데이터만 요청 받는다.
- 트래픽이 적다.



자바스크립트 기반 SPA 프레임워크



자바스크립트 라이브러리 탄생 배경


- jQuery
 - 크로스브라우징 이슈 및 레거시 Web API의 부족한 기능 보완
- Angular
 - 데이터 바인딩 및 컴포넌트 단위 개발 패턴 정립
- React
 - 양방향 바인딩에 의한 성능 저하 문제 해결
 - 짧고 간결한 함수형 컴포넌트 지원



React 튜토리얼

- <https://ko.reactjs.org/tutorial/tutorial.html>

Support Ukraine ua [Help Provide Humanitarian Aid to Ukraine.](#)

 **React** 문서 자습서 블로그 커뮤니티 🔍 v18.0.0 🌐 Languages GitHub

자습서: React 시작하기

이 자습서는 React에 대한 어떤 지식도 가정하지 않습니다.

자습서를 시작하기 전에

우리는 이 자습서에서 작은 게임을 만들 겁니다. **게임을 만들고 싶지 않아서 자습서를 건너뛰고 싶을 수 있습니다. 그래도 한번 해보세요!** 자습서를 통해 React 앱을 만드는 기본적인 사항들을 배웁니다. 이것을 마스터하면 React를 더 깊게 이해할 수 있습니다.

자습서 ^

자습서를 시작하기 전에
무엇을 만들게 될까요?
필요한 선수 지식

자습서를 위한 설정
선택 1: 브라우저에 코드 작성하기
선택 2: 자신의 컴퓨터에서 코드 작성하기
도움이 필요할 때!

개요
React란 무엇일까요?
초기 코드 살펴보기
Props를 통해 데이터 전달하기
사용자와 상호작용하는 컴포넌트 만들기
개발자 도구

React 튜토리얼

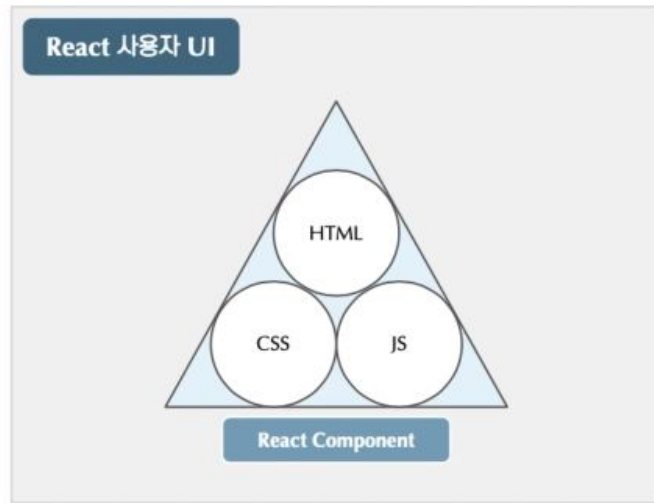
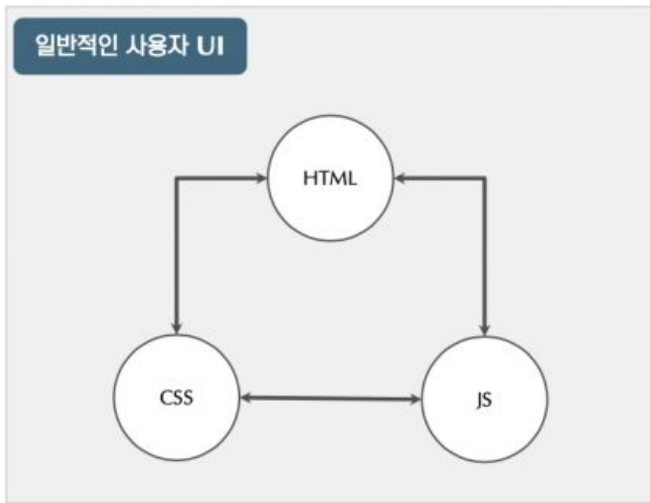
- <https://www.w3schools.com/react/default.asp>



The screenshot shows the top section of the W3Schools website. At the top left is the W3Schools logo. To its right are links for 'Tutorials' and 'References', each with a dropdown arrow. Further right is a 'Website' link and a green 'Log in' button. Below this is a dark navigation bar with icons for a menu, home, and search, and text links for 'HTML', 'CSS', 'JAVASCRIPT' (which is highlighted), and 'SQL'. Below the navigation bar is a banner for 'BRILLIANT.ORG' with the text 'Learn math interactively' and a diagram of a right triangle with squares on its sides labeled a^2 and c^2 . Below the banner, the text 'React Tutorial' is displayed. At the bottom are two green buttons: '< Home' on the left and 'Next >' on the right.

React 소개

- React는 SPA 기반 WEB UI(User Interface) 구성을 위한 Javascript 라이브러리.
- 재사용 가능한 UI 컴포넌트 생성. (사용법이 쉬움)
- HTML UI 컴포넌트 = HTML + CSS + Javascript를 하나로.
- 기존 HTML 화면을 어떻게 React 컴포넌트로 분리 하느냐가 관건.



React CDN

- 기존 HTML문서에서 바로 React 사용 가능.
- React 코드 작성 용
 - react.development.js
 - react-dom.development.js
- JSX 구문 번역 용
 - babel.min.js

```
<head>  
  <script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>  
  <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js" crossorigin></script>  
  <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>  
</head>
```

Create React App 설치

- 먼저 Node.js가 설치 되어있어야 한다.
- npx 명령어와 create-react-app 명령어를 이용해서 React 개발 환경 자동 설정.
 - CRA를 이용하지 않고 설정을 한다면 Nodejs, Babel, Webpack 등을 모두 수동 설정 해야 함.

```
npx create-react-app my-react-app
```

제거 하려면

```
npm uninstall -g create-react-app
```

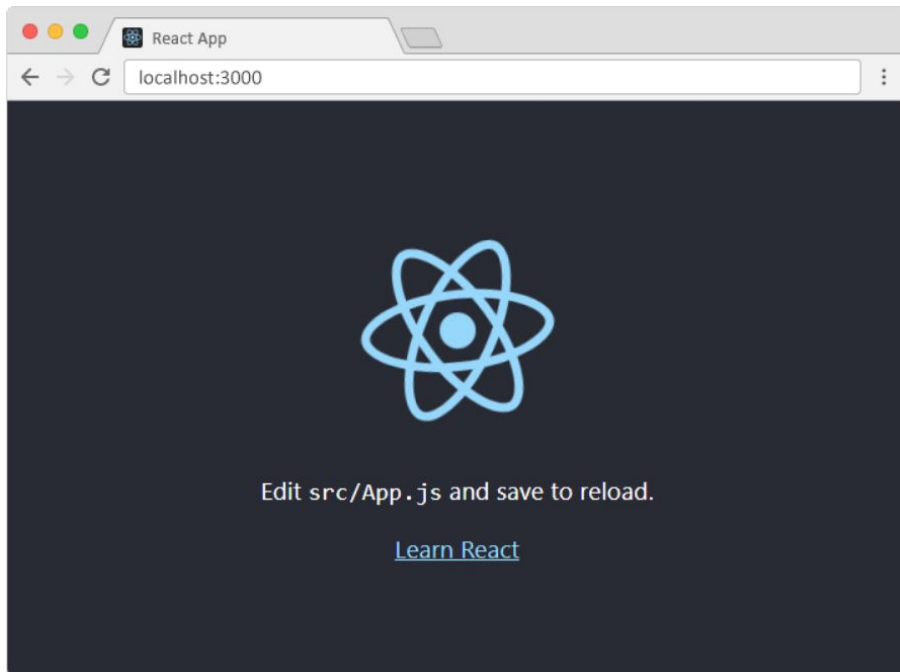
실행 하려면

```
cd my-react-app
```

```
npm start
```

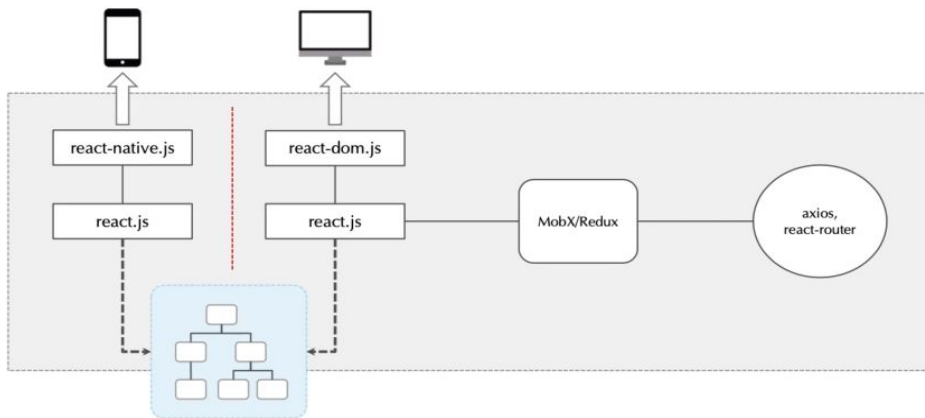

샘플 웹앱 실행하기

- npm start



React Application 개발 구성 요소

- 컴포넌트 : **React** 어플리케이션 구성 기본 단위
 - react.js 코어 라이브러리
 - react-dom.js 라이브러리
 - react-native.js 라이브러리
- React는 UI 요소들을 화면에 표현 하는 것에만 집중.
 - 그 외의 추가 기능은 별도의 Third-party 라이브러리와 함께 구성.



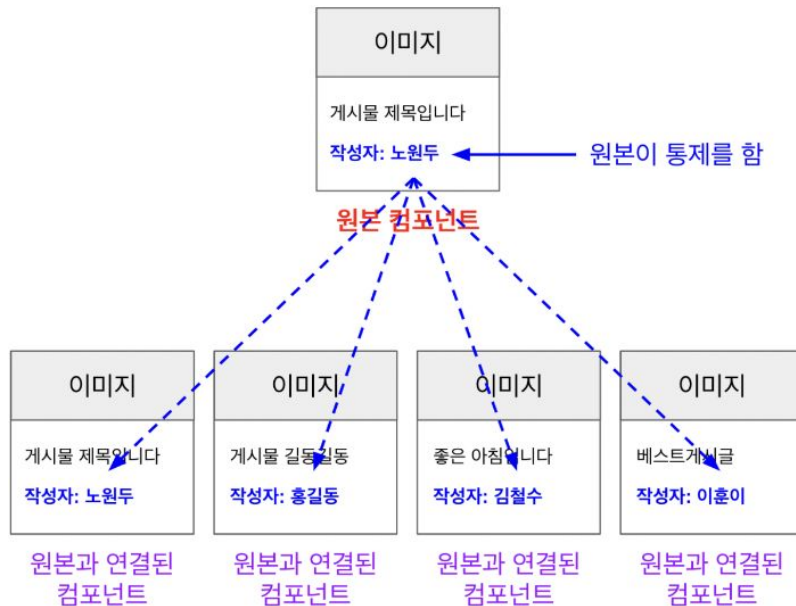
React 동작 방식

- 기존 웹 애플리케이션은 클라이언트의 요청이 있으면 서버에서 UI(화면구성 + 데이터)를 생성.
- React는 필요한 데이터를 서버에 요청하고 응답 받은 데이터를 클라이언트에서 가공해서 UI생성.
- 서버는 요청 파라미터를 받아서 처리 결과 데이터를 JSON 형식으로 클라이언트에 전달.
- Client Side Rendering (CSR)



React 컴포넌트

- 엘리먼트와 기능을 부품화 해서 재 사용 가능.
- 각 컴포넌트에 맞는 데이터 사용.



React.js 컴포넌트

- 클래스 컴포넌트
 - 컴포넌트 라이프사이클
 - 이벤트 바인딩
 - Presentational / Container 패턴
- 함수형 컴포넌트
 - Stateless
 - 고차 컴포넌트
 - Hooks
- 비동기 처리
 - Lifecycle methods, Event handler, useEffect
 - Redux Middleware
 - React Query, SWR
 - Suspense API

```
class Greeting extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

```
function Greeting(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

```
function usePost(postId) {  
  return useQuery(["post", postId], () => getPostById(postId), {  
    enabled: !!postId,  
  });  
}
```

클래스형 컴포넌트와 함수형 컴포넌트

- 실무에선 기존 클래스형 컴포넌트와 함수형 컴포넌트가 혼재되어 있다.

클래스형	함수형
<pre>import { Component } from "react"; class New extends Component { render(){ return <div>이것은 클래스형 컴포넌트입니다.</div> } } export default New</pre>	<pre>function New() { return <div>이것은 함수형 컴포넌트입니다.</div> } export default New</pre>  <pre>const New = () => <div>이것은 함수형 컴포넌트입니다.</div> export default New</pre>

React Element

- 브라우저 DOM 엘리먼트에 대한 정보를 담고 있는 객체.
- React DOM 객체 리터럴.
- 실제 DOM은 아님. JSX형식을 표현된 단순 객체.
- ReactElement는 포함하는 값을 변경할 수 없다. (필요 할 경우 새로 생성)

```
<div id="mydiv"></div>

<script type="text/babel">
  function Hello() {
    return <h1>Hello World!</h1>;
  }

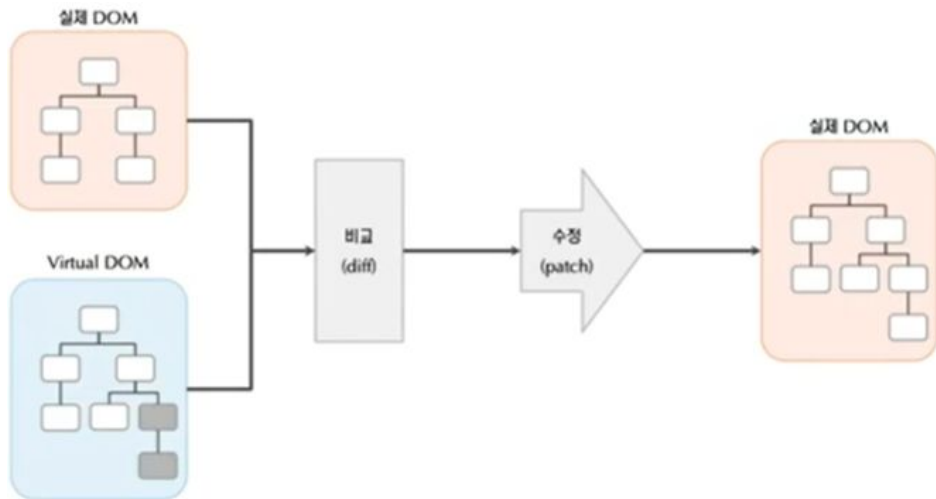
  ReactDOM.render(<Hello />, document.getElementById('mydiv'))
</script>
```

```
const myelement = (
  <table>
    <tr>
      <th>Name</th>
    </tr>
    <tr>
      <td>John</td>
    </tr>
    <tr>
      <td>Elsa</td>
    </tr>
  </table>
);
```

```
ReactDOM.render(myelement, document.getElementById('root'));
```

Virtual DOM

- Virtual DOM은 브라우저의 실제 DOM을 반영하고 있는 데이터의 모음.
- 기존 WEB 방식은 실제 DOM을 직접 접근하고 전체 DOM에 반영.
- React는 Virtual DOM을 변경하면 실제 DOM과 비교 해서 변경 된 부분만 반영.
- 가상 DOM과 실제 DOM을 비교(diff) 수정(patch)

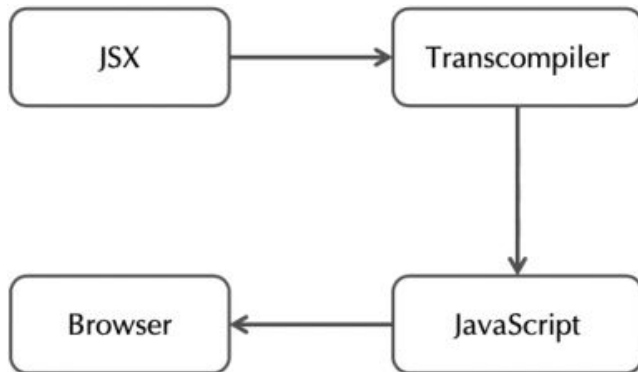


JSX

- JSX (Javascript XML) 문법적 편의를 위해 개발 된 Javascript 확장.
- 트랜스컴파일러 (Transcompiler) : JSX를 Javascript로 변환.
- HTML이 아닌 XML 형식.
- 중첩된 엘리먼트를 자바스크립트만으로 구현 하면 너무 복잡해 진다.

```
//JSX
ReactDOM.render(
  <h1 className='heading'>Hello World</h1>,
  document.getElementById('content')
)
```

```
//JavaScript
ReactDOM.render(
  React.createElement(
    'h1',
    { className: 'heading' },
    'Hello World'
  )
)
```



JSX

- JSX는 HTML과 비슷한 구조.
 - HTML과 구분 하기 위해 JSX는 Pascal Case로 작성.
 - 예) <List />, <Button />
 - XML 문법이므로 태그 종료 / 또는 종료태그 반드시 사용.
 - 중괄호 {} : JSX 코드 내부에 자바스크립트 사용시 중괄호{} 블록으로 묶음.
 - className : 자바스크립트의 class 키워드와 구분하기 위해 엘리먼트의 class 속성은 className으로 사용.
 - htmlFor : 자바스크립트의 for 키워드와 구분하기 위해 엘리먼트의 for 속성은 htmlFor로 사용.

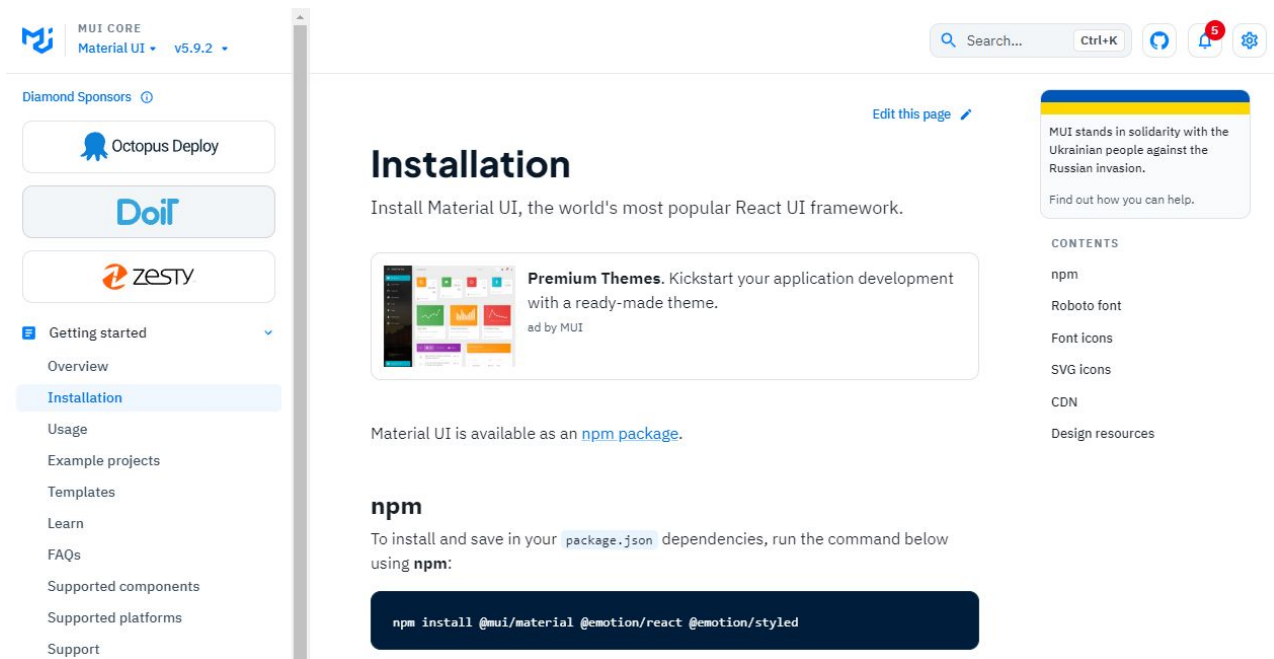
```
class DateTimeNow extends React.Component {  
  render(){  
    let dateTimeNow = new Date().toLocaleDateString();  
    return React.createElement(  
      'span',  
      null,  
      '현재 시간 : ${dateTimeNow}.' // ECMAScript 6  
    );  
  }  
}
```



```
class DateTimeNow extends React.Component {  
  render(){  
    let dateTimeNow = new Date().toLocaleDateString();  
    return <span> 현재 시간 {dateTimeNow}</span>  
  }  
}
```

Material UI

- <https://mui.com/material-ui/getting-started/installation/>



The screenshot shows the Material UI website's installation page. The left sidebar contains the MUI logo, version information (v5.9.2), and a list of navigation links: Getting started (selected), Overview, Installation, Usage, Example projects, Templates, Learn, FAQs, Supported components, Supported platforms, and Support. The main content area is titled 'Installation' and includes a sub-header 'Install Material UI, the world's most popular React UI framework.' Below this is a section for 'Premium Themes' with a thumbnail image and text. A note states 'Material UI is available as an npm package.' The 'npm' section provides instructions on how to install and save dependencies in a package.json file, followed by a code block showing the command: `npm install @mui/material @emotion/react @emotion/styled`. The right sidebar features a search bar, social media links, a notification bell, and a settings gear. A banner at the top right expresses solidarity with the Ukrainian people against the Russian invasion, and a 'CONTENTS' section lists links to npm, Roboto font, Font icons, SVG icons, CDN, and Design resources.

MUI CORE
Material UI v5.9.2

Diamond Sponsors

- Octopus Deploy
- Doit
- zesty

Getting started

- Overview
- Installation
- Usage
- Example projects
- Templates
- Learn
- FAQs
- Supported components
- Supported platforms
- Support

Search... Ctrl+K

Edit this page

Installation

Install Material UI, the world's most popular React UI framework.

Premium Themes. Kickstart your application development with a ready-made theme.
ad by MUI

Material UI is available as an [npm package](#).

npm

To install and save in your `package.json` dependencies, run the command below using **npm**:

```
npm install @mui/material @emotion/react @emotion/styled
```

MUI stands in solidarity with the Ukrainian people against the Russian invasion.
Find out how you can help.

CONTENTS

- npm
- Roboto font
- Font icons
- SVG icons
- CDN
- Design resources

과정 실습

리액트 실습

- 목록 구현하기
 - 글 목록
 - 글 추가
 - 글 검색

영화 평점 정보 관리

오늘날짜: 2022-08-04

영화 정보 입력

영화제목 공각기동대

개봉날짜

영화 정보 검색

영화검색

영화 정보 목록

순서	제목	개봉날짜	평점	삭제
1	로키	1982-01-15	5	<input type="button" value="삭제"/>
2	터미네이터	1984-01-15	4	<input type="button" value="삭제"/>
3	그랑블루	1988-01-15	5	<input type="button" value="삭제"/>

React.js 컴포넌트

배포하는 법

- **npm run build**

- 프로젝트 개발 버전에서 배포할 파일들만 따로 빼내서 좀더 가볍게 관리.
- 프로젝트를 가볍게 함으로 성능 향상.
- **build** 디렉토리가 생성 되고 거기에 배포 할 파일들만 빌드 됨.

- **npm install -g serve**

- 로컬에서 프로젝트를 간단히 실행 시켜주는 서버 설치.

- **serve -s build**

- 서버 실행.

- **npx serve -s build**

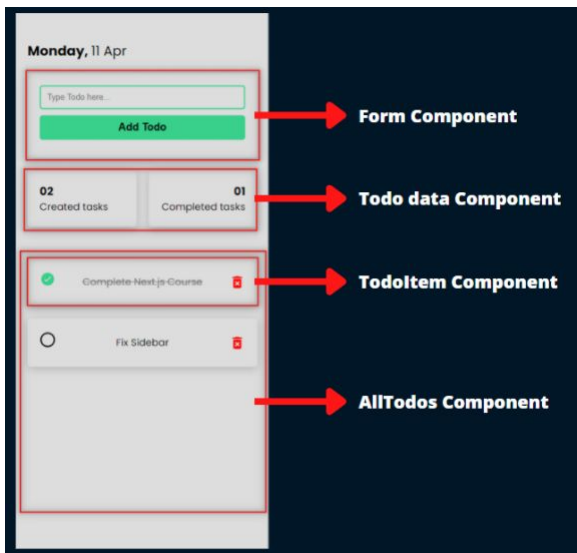
- 로컬에 서버 설치 없이 한번만 실행 할 경우.

리액트 CRA 실행 명령어 정리

- 프로젝트 생성
 - > `npx create-react-app 프로젝트명`
- 프로젝트 실행
 - > `npm start`
- 빌드하기
 - > `npm run build`
 - // 프로젝트에 `build` 폴더가 생기고 압축파일로 생성 (용량 감소)
- `npm`을 이용한 간단한 웹서버 설치 (특정 디렉토리를 간단히 서버로 실행)
 - > `npm install -g serve`
- 빌드 된 프로젝트 실행
 - > `serve -s build`
- `npx`를 이용해서 한번만 실행하는 웹서버 실행(인스턴스 웹서버)
 - // -s 옵션 : `build` 디렉토리를 `document root`로 지정
 - > `npx serve -s build`

React 컴포넌트 만들기

- https://www.w3schools.com/react/react_components.asp



Create a Class component called **Car**

```
class Car extends React.Component {  
  render() {  
    return <h2>Hi, I am a Car!</h2>;  
  }  
}
```

Create a Function component called **Car**

```
function Car() {  
  return <h2>Hi, I am a Car!</h2>;  
}
```

props

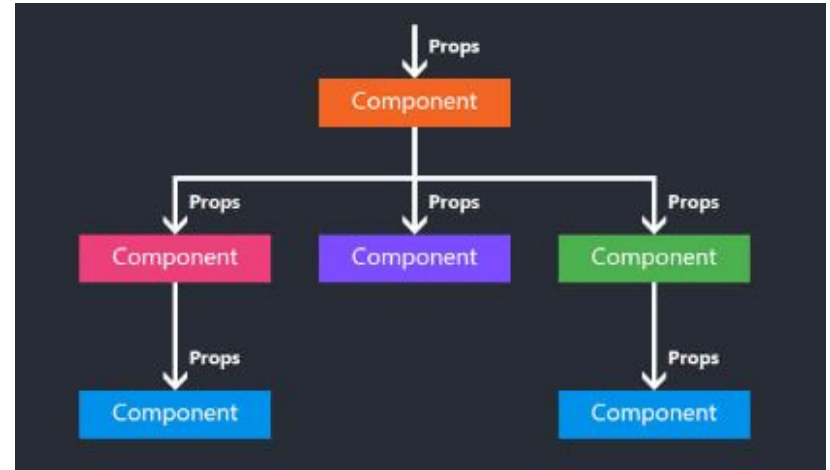
- https://www.w3schools.com/react/react_props.asp

Add a "brand" attribute to the Car element:

```
const myElement = <Car brand="Ford" />;
```

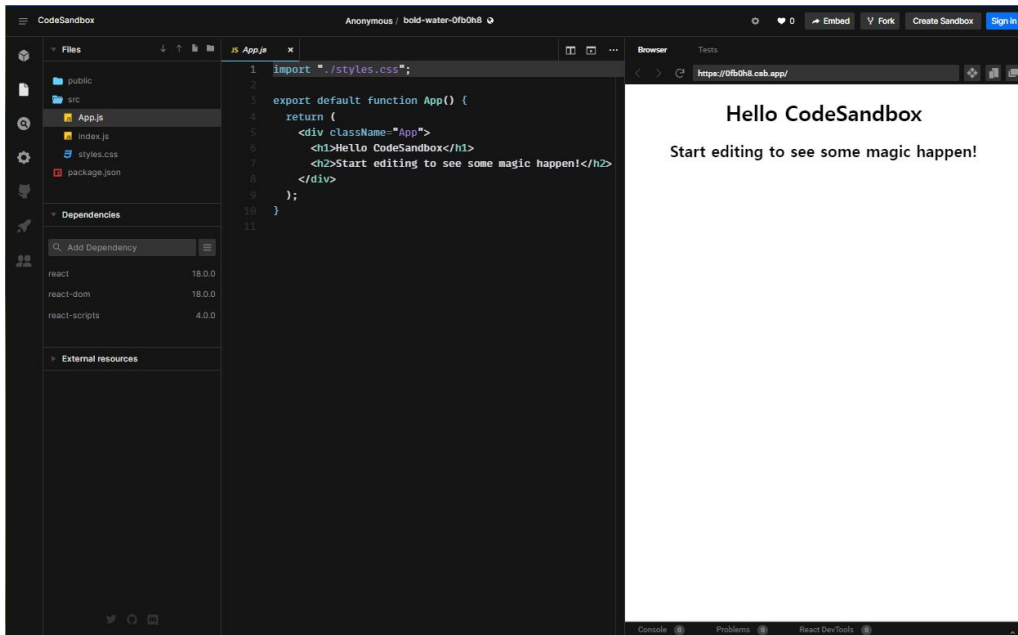
Use the brand attribute in the component:

```
function Car(props) {  
  return <h2>I am a { props.brand }!</h2>;  
}
```



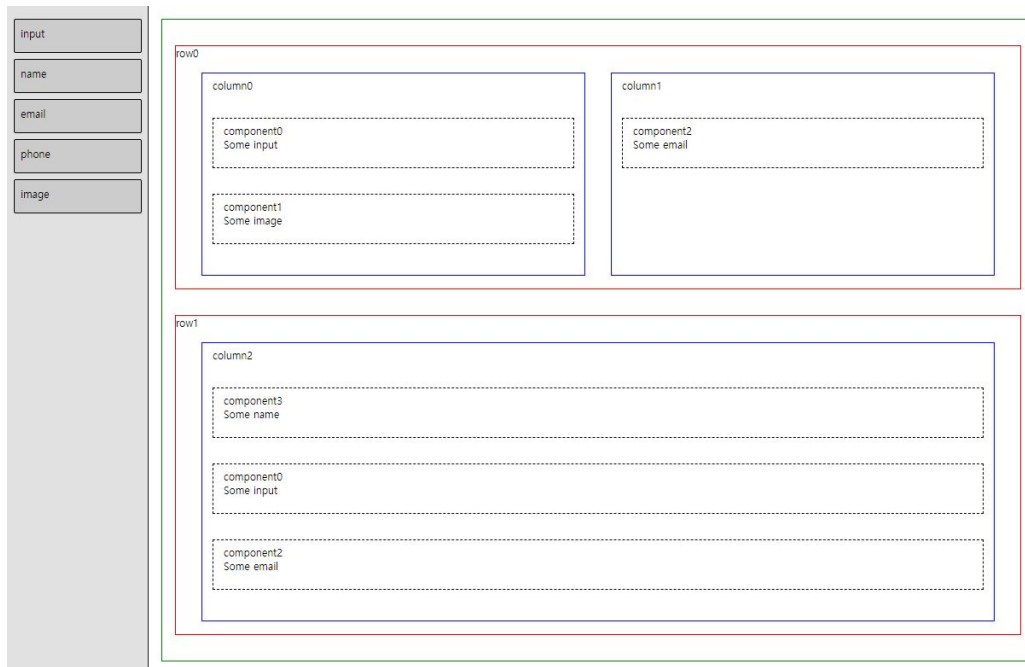
클라우드 개발환경

- Goorm IDE (무료 Private 가능)
 - <https://ide.goorm.io/>
- CodeSandbox (무료 Public)
 - <https://codesandbox.io/s/new>
- StackBlitz (무료 Public)
 - <https://stackblitz.com/>
 - <https://stackblitz.com/?starters=frontend>
- CondPen
- JSfiddle
- GitHub Codespaces (유료)



과정 실습

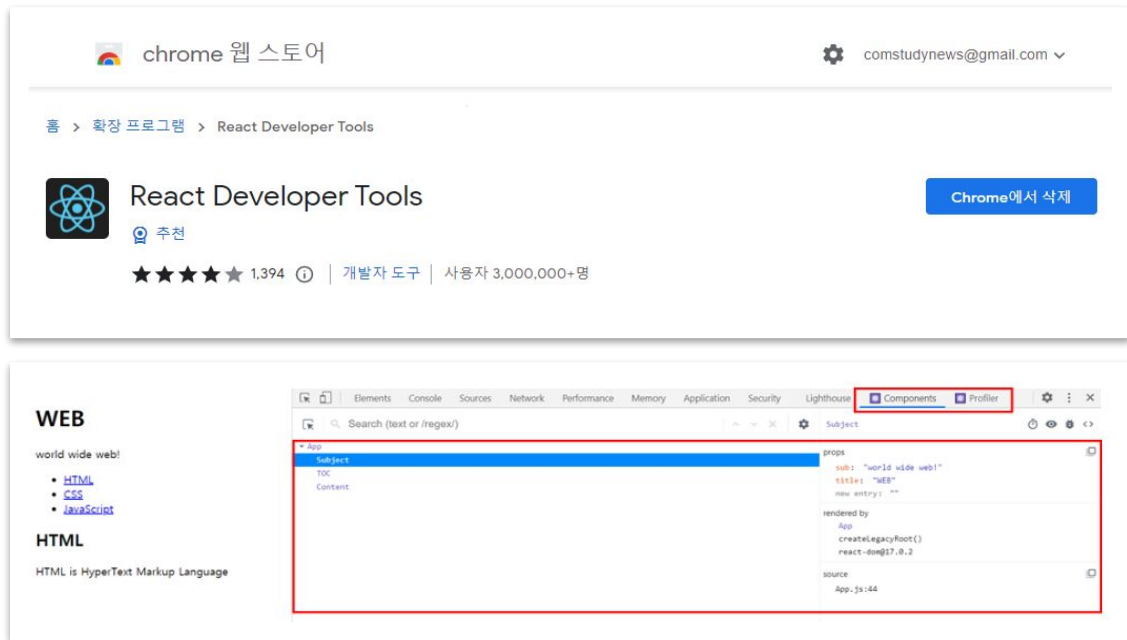
리액트 Components 실습



React.js props와 state

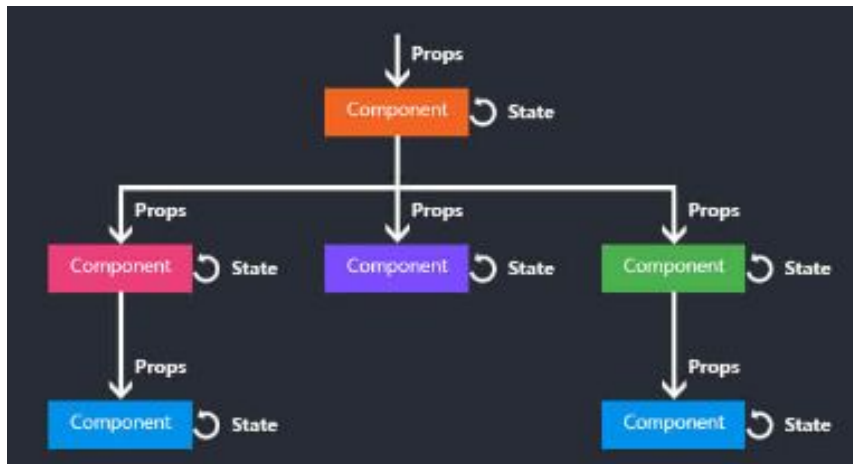
React Developer Tools

- <https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi>



state

- https://www.w3schools.com/react/react_usestate.asp



Use a button to update the state:

```
import { useState } from "react";
import ReactDOM from "react-dom/client";

function FavoriteColor() {
  const [color, setColor] = useState("red");

  return (
    <>
      <h1>My favorite color is {color}!</h1>
      <button
        type="button"
        onClick={() => setColor("blue")}
      >Blue</button>
    </>
  )
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<FavoriteColor />);
```


이벤트 state props와 render 함수

- render 함수를 사용하여 HTML을 웹 페이지에 렌더링.
- public > index.html 파일

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

과정 실습

React 훅 실습

- Node.js 서버에서 데이터를 Ajax 통신.

전화번호부

이름 : 전화번호 : 이메일 :

- 1.오탁근 / 010-1111-1111 / bread1@naver.com /
- 2.염정아 / 010-1111-2222 / bread2@naver.com /
- 3.정승호 / 010-1111-3333 / bread3@naver.com /
- 4.권석준 / 010-1111-4444 / bread4@naver.com /
- 5.추가사람1 / 010-1111-5555 / bread5@naver.com /
- 6.추가사람1 / 010-1111-5555 / bread5@naver.com /
- 7.추가사람1 / 010-1111-5555 / bread5@naver.com /

```
function App() {

  const [phoneList, setPhoneList] = useState([]);

  useEffect(() => {
    axios.get("http://localhost:5000/phone/list").then(function(response) {
      // console.log("서버로부터 응답 : " + response.data);
      setPhoneList(response.data);
    })
  })

  return (
    <>
      <div style={{
        display:"flex",
        flexDirection:'row',
        alignItems:'center',
        justifyContent:'center',
        backgroundColor:'greenyellow'
      }}><h1>전화번호부</h1></div>
      <InputComponent items={phoneList} />
      <PrintComponent items={phoneList} />
    </>
  );
}
```

React.js 이벤트

Object.assign 과 Spread 연산자

- Object.assign(target, ...sources)

```
const target = { a: 1, b: 2 };  
const source = { b: 4, c: 5 };  
  
const returnedTarget = Object.assign(target, source);  
  
console.log(target);  
  
console.log(returnedTarget);
```

```
{ a: 1, b: 4, c: 5 }  
{ a: 1, b: 4, c: 5 }
```

Object.assign 과 Spread 연산자

- 같은 속성 이름이 있다면 덮어쓰기
- 새로운 속성은 추가.
- 기존에 있던것은 그대로...

```
const target = { a: 1, b: 2 };
const sources = [
  { b: 4, c: 5 },
  { b: 6, c: 7, d: 8 },
  { b: 9, c: 10, d: 11 },
  { b: 12, c: 13, d: 14, e: 15 },
];

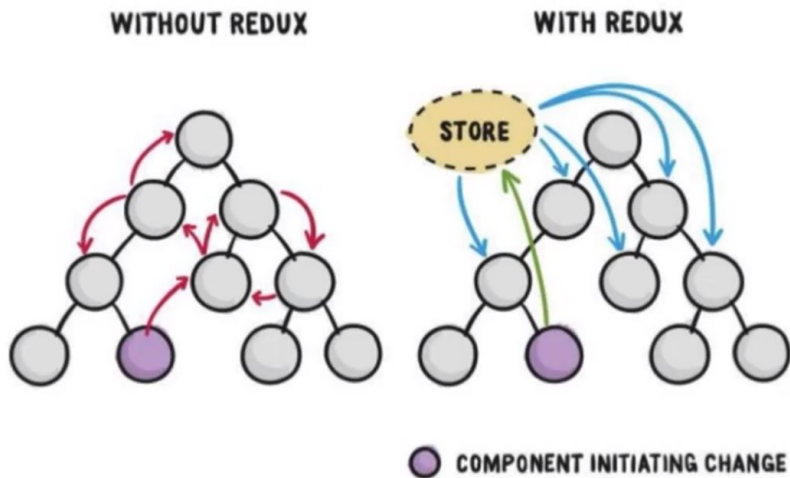
Object.assign(target, ...sources);

console.log(target);
```

```
{ a: 1, b: 12, c: 13, d: 14, e: 15 }
```

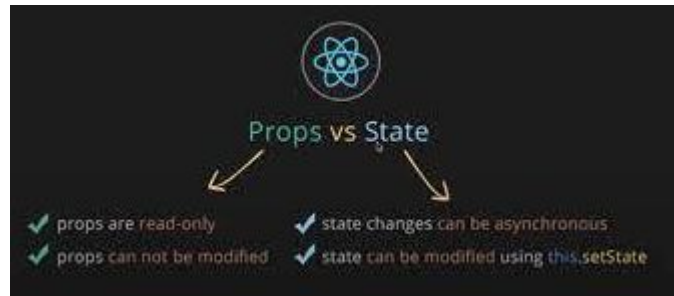
컴포넌트 이벤트 만들기

- React의 데이터 흐름은 단방향이다.
- Redux를 사용하는 목적은 프로그램 소스코드의 복잡도를 줄이기 위함.



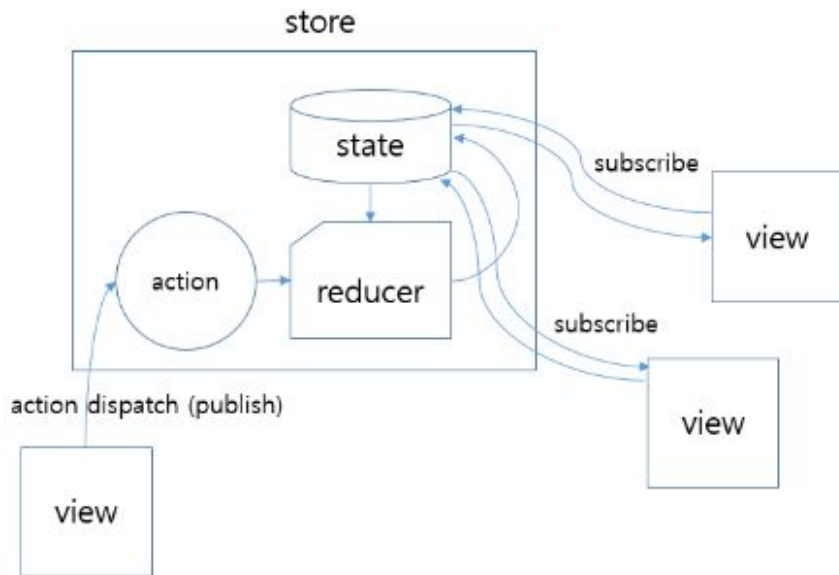
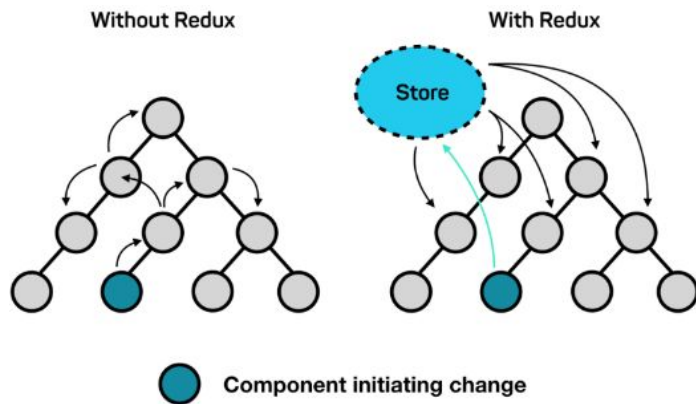
React 베이스캠프

- 컴포넌트의 상태가 변하면 변경된 상태를 참조하는 컴포넌트가 다시 그려짐.
- props는 읽기 전용.
- state는 `setState`를 이용하여 수정 가능.
 - 상위 컴포넌트가 하위 컴포넌트 수정시 props 수정.
 - 하위 컴포넌트가 상위 컴포넌트 수정시 event 이용.
- 리덕스
 - 컴포넌트 상태를 하나의 저장소인 Store를 이용.
 - 좀 더 체계적으로 상태 관리.
 - Store의 특정 상태가 변경되면, 관련된 컴포넌트에 적용 될 수 있게 처리.



React Redux

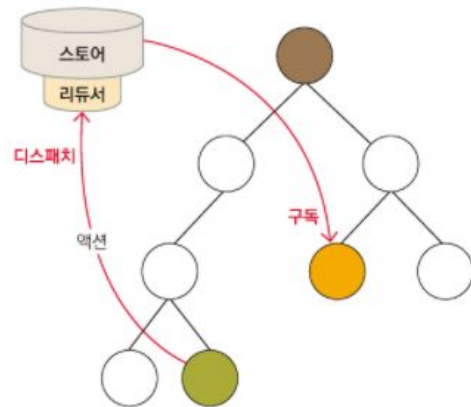
- state 종속성 탈피
- props -> props -> props 지옥 탈출



Redux

Redux란?

- 리액트에서 상태를 더 효율적으로 관리하는 데 사용하는 상태 관리 라이브러리
- 리덕스는 쉽게 설명하면 상태 관리의 로직을 컴포넌트 밖에서 처리 하는 것이다.
- 리덕스를 사용하면 스토어라는 개체 내부에 상태를 담게 된다.



리덕스를 적용한 구조

예측 가능

Redux 는 **일관되게** 작동하고 다양한 환경(클라이언트, 서버 및 기본)에서 실행되며 **테스트하기 쉬운** 애플리케이션을 작성하는 데 도움이 됩니다 .

중앙 집중식

애플리케이션의 상태와 논리를 중앙 집중화하면 **실행 취소/다시 실행** , **상태 지속성** 등과 같은 강력한 기능을 사용할 수 있습니다.

디버깅 가능

Redux DevTools를 사용하면 **애플리케이션 상태가 언제, 어디서, 왜, 어떻게 변경되었는지** 쉽게 추적 할 수 있습니다 . Redux의 아키텍처를 사용하면 변경 사항을 기록하고 "**시간 여행 디버깅**" 을 사용하여 완전한 오류 보고서를 서버에 보낼 수도 있습니다.

유연한

Redux 는 모든 **UI 계층** 에서 작동하며 사용자의 요구에 맞는 **대규모 애드온** **에코시스템**을 갖추고 있습니다.

React Redux

공식적인

React Redux는 Redux 팀에서 유지 관리하며 **Redux** 및 **React**의 최신 **API**로 최신 상태를 유지합니다 .

예측 가능

React의 컴포넌트 모델과 함께 작동하도록 설계되었습니다 . Redux에서 구성 요소에 필요한 값을 추출하는 방법을 정의하면 구성 요소가 필요에 따라 자동으로 업데이트됩니다.

캡슐화

구성 요소가 **Redux** 저장소와 상호 작용할 수 있도록 하는 API를 제공 하므로 해당 논리를 직접 작성할 필요가 없습니다.

최적화

복잡한 성능 최적화를 자동으로 구현 하여 필요한 데이터가 실제로 변경된 경우에만 구성 요소를 다시 렌더링합니다.

Redux는 4가지 행동으로 분리된다.

- Action

- 어떤 작업을 할 건지 구분 해준다.
- Object 형태로 flag를 던짐.
- 데이터도 던짐.

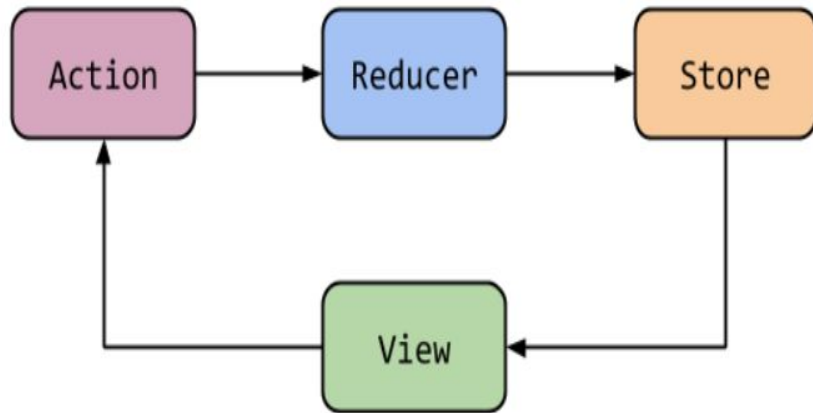
- Reducer

- Action에게 받은 object로, 모든 컴포넌트들이 공유 할 수 있는 state의 값을 변경한다.
- setState 하는 역할

- Store

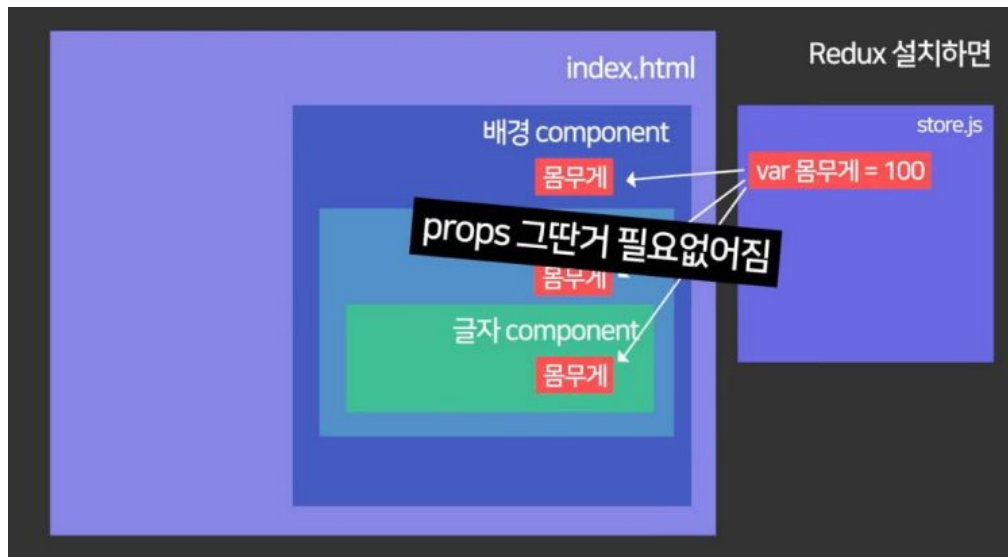
- 상태를 계속 확인 하고 변경된 사항을 View에게 알려 줌.
- 변경이 일어나면 Component는 자동 갱신 됨으로 Reducer를 Store에 적용 시켜주기만 하면 됨.

- View



React Redux

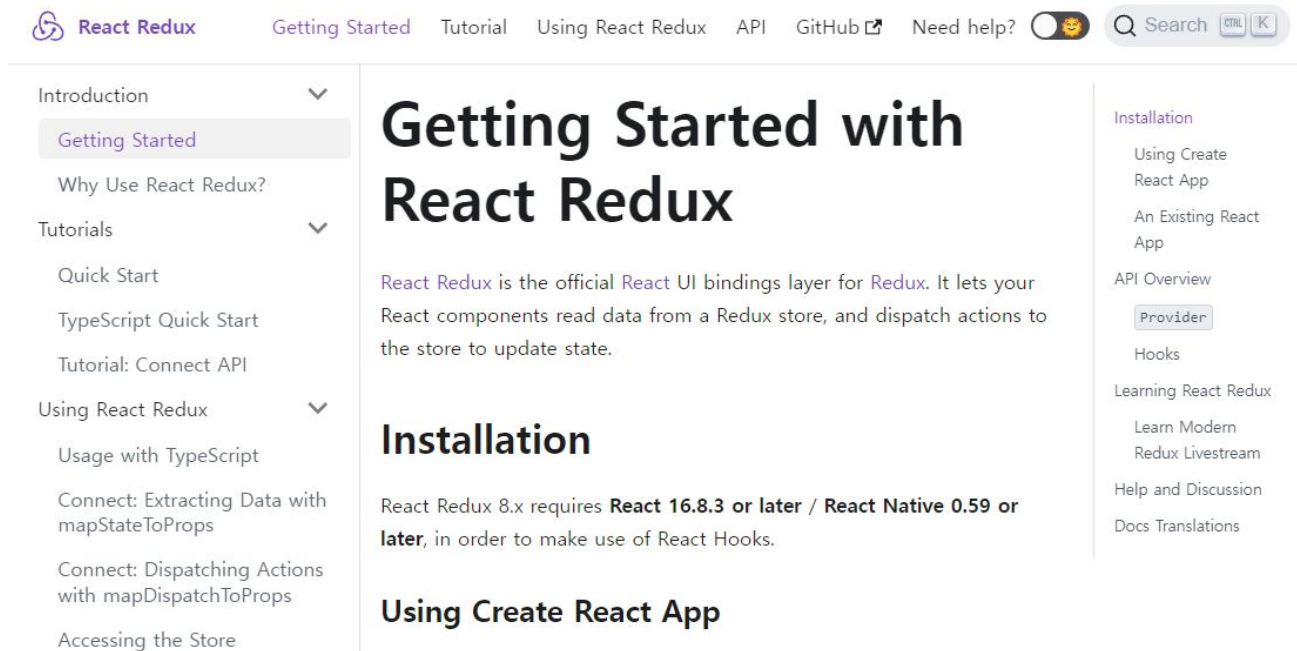
- https://drive.google.com/file/d/10e7lunHgpru8gF6ycNv_Uy4BqVZcXCKv/view?usp=sharing



<https://youtu.be/QZcYz2NrDIs>

<https://www.opentutorials.org/module/4518>

React Redux Getting Started



The screenshot shows the React Redux documentation website. The header includes the React Redux logo, navigation links for 'Getting Started', 'Tutorial', 'Using React Redux', 'API', 'GitHub', and 'Need help?', a search bar, and a dark mode toggle. The left sidebar contains a table of contents with categories like 'Introduction', 'Getting Started', 'Tutorials', 'Using React Redux', and 'Installation'. The main content area features the title 'Getting Started with React Redux', a paragraph describing React Redux as the official React UI bindings layer for Redux, and sections for 'Installation' and 'Using Create React App'. The right sidebar lists additional resources such as 'Installation', 'API Overview', 'Provider', 'Hooks', 'Learning React Redux', 'Help and Discussion', and 'Docs Translations'.

React Redux

Getting Started Tutorial Using React Redux API GitHub Need help?

Search

Introduction

Getting Started

Why Use React Redux?

Tutorials

Quick Start

TypeScript Quick Start

Tutorial: Connect API

Using React Redux

Usage with TypeScript

Connect: Extracting Data with `mapStateToProps`

Connect: Dispatching Actions with `mapDispatchToProps`

Accessing the Store

Getting Started with React Redux

React Redux is the official React UI bindings layer for Redux. It lets your React components read data from a Redux store, and dispatch actions to the store to update state.

Installation

React Redux 8.x requires **React 16.8.3 or later** / **React Native 0.59 or later**, in order to make use of React Hooks.

Using Create React App

Installation

Using Create React App

An Existing React App

API Overview

Provider

Hooks

Learning React Redux

Learn Modern Redux Livestream

Help and Discussion

Docs Translations

<https://react-redux.js.org/>

<https://react-redux.js.org/introduction/getting-started>

redux, react-redux 설치

- npm install redux --save
 - npm install react-redux --save
 - npm install @reduxjs/toolkit react-redux
-
- createStore는 디프리케이드 됨. 다음과 같이 import를 수정 하면 그대로 사용 가능.

```
import { legacy_createStore as createStore } from "redux";
```

index.js

```
import React from "react";
import ReactDOM from "react-dom/client";
import { Provider } from "react-redux";
import { createStore } from "redux";
//import store from "./store";

import App from "./App";

const currentState = { 체중: 100, 키: 180 };
function reducer(state = currentState, action) {
  if (currentState == undefined) {
    return { 체중: 100, 키: 180 };
  }
  if (action.type === "체중증가") {
    state.체중++;
  } else if (action.type === "체중감소") {
    state.체중--;
  }
  const newState = { ...state };
  return newState;
}

let store = createStore(reducer);

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <Provider store={store}>
    <App />
  </Provider>
);
```

App.js

```
import { useSelector } from "react-redux";
import WeightController from "./WeightController";

function App() {
  const 몸무게 = useSelector((state) => state.체중);
  return (
    <div className="App">
      <p>몸무게 : {몸무게}</p>
      <WeightController />
    </div>
  );
}

export default App;
```

WeightController.js

```
import { useDispatch, useSelector } from "react-redux";

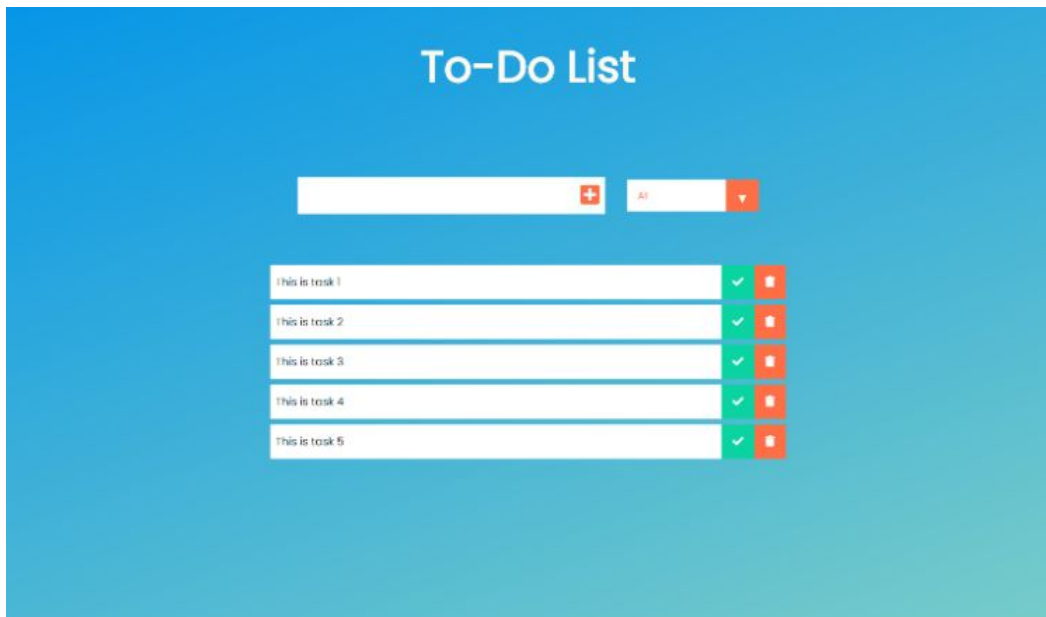
function WeightController() {
  const dispatch = useDispatch();
  return (
    <div>
      <button
        onClick={function () {
          dispatch({ type: "체중증가" });
        }}
      >
        몸무게 증가
      </button>
    </div>
  );
}

export default WeightController;
```


과정 실습

React event 실습

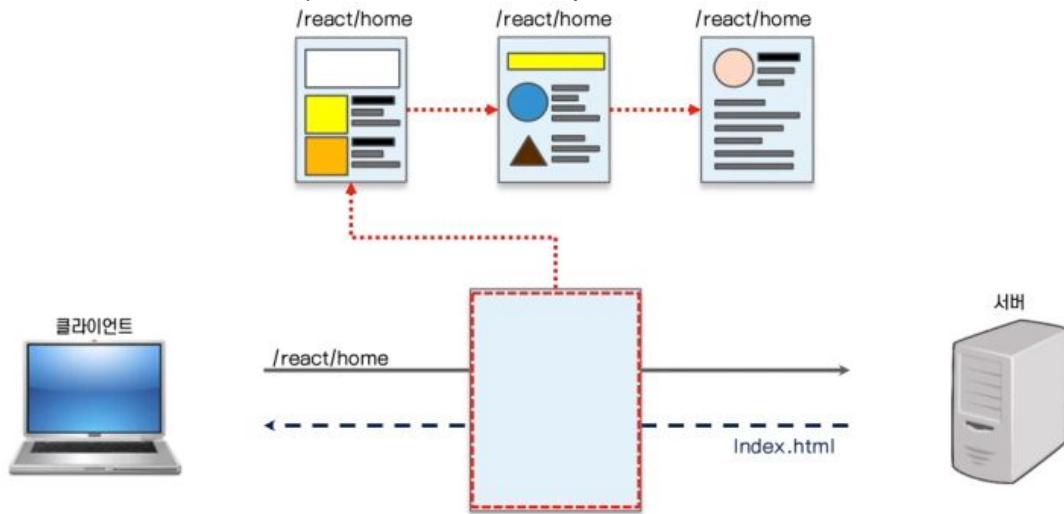
- Todo list 구현



React Router

Router 개요

- SPA는 단일 페이지 - URL 요청하면 같은 페이지 반환.
- 하나의 페이지에서 탭이나 레이어드 형태로 화면이 변환.
 - 북마크 불가능. history기능 구현이 까다로움.
- URL과 페이지를 유기적으로 구성하고, 브라우저의 history와 동기화 해야 함.



React-router 주요 컴포넌트

- 모듈 설치
 - npm install --save react-router-dom
- 중요 Router 관련 컴포넌트
 - <BrowserRouter>
 - <Routes>, <Route>
 - <Link>

길동이의 홈페이지 헤더

안녕하세요. 길동이의 홈페이지 메인

- 상품 1
- 상품 2

```
src > JS App.js > ...
1 import {BrowserRouter, Routes, Route} from 'react-router-dom';
2 import React from 'react';
3 import Header from './Header';
4 import Main from './Main';
5 import NotFound from './NotFound';
6 import Product from './Product';
7
8 const App = ()=>{
9   return (
10     <>
11       <div className="App">
12         <BrowserRouter>
13           <Header></Header>
14           <Routes>
15             <Route path="/" element={<Main />}></Route>
16             <Route path="/product/*" element={<Product />}></Route>
17             { /* 상단에 일치하는 항목이 없을 경우 보여지는 부분 */ }
18             <Route path="*" element={<NotFound />}></Route>
19           </Routes>
20         </BrowserRouter>
21       </div>
22     </>
23   )
24 }
25
26 export default App;
```

React-router 주요 컴포넌트

- Main 페이지에서 Product 페이지 링크

길동이의 홈페이지 헤더

안녕하세요. 길동이의 홈페이지 메인

- [상품 1](#)
- [상품 2](#)



길동이의 홈페이지 헤더

상품 페이지입니다.

```
src > JS Main.js > ...
1  import { Link } from "react-router-dom";
2
3  const Main = (props) => {
4    return (
5      <>
6        <h3>안녕하세요. 길동이의 홈페이지 메인</h3>
7        <ul>
8          <Link to="/product/1"><li>상품 1</li></Link>
9          <Link to="/product/2"><li>상품 2</li></Link>
10         </ul>
11      </>
12    );
13  }
14
```

```
src > JS Product.js > ...
1  import React from 'react';
2
3  const Product = (props) => {
4    return (
5      <>
6        <h3>상품 페이지입니다.</h3>
7      </>
8    );
9  }
10
11  export default Product;
```

reactrouter.com

- <https://v5.reactrouter.com/web/guides/quick-start>

The image shows a screenshot of the React Router v5 Quick Start guide. On the left is a sidebar with the React Router logo and navigation links: CORE, WEB (highlighted), and NATIVE. Below these are sections for ANNOUNCEMENTS and EXAMPLES. The main content area is titled "Quick Start" and contains text about getting started with React Router in a web app, recommending the use of Create React App. It includes a code block for installing create-react-app and making a new project. Below this is the "Installation" section, which explains how to install React Router from the public npm registry using either npm or yarn. A code block shows the command to install react-router-dom. A small overlay window titled "API" lists the available hooks (useHistory, useLocation, useParams, useRouteMatch) and the components (BrowserRouter, HashRouter) with their props. To the right of the API overlay is a section titled "<BrowserRouter>" which explains that it uses the HTML5 history API and includes a code block showing the JSX for using BrowserRouter in an application.

REACT ROUTER

CORE WEB NATIVE

ANNOUNCEMENTS

The Future of React Router

EXAMPLES

- Basic
- URL Parameters
- Nesting
- Redirects (Auth)
- Custom Link
- Preventing Transitions
- No Match (404)
- Recursive Paths
- Sidebar
- Animated Transitions
- Route Config
- Modal Gallery
- StaticRouter Context
- Query Parameters

Quick Start

To get started with React Router in a web app, you'll need a React web app. If you need to create one, we recommend you try [Create React App](#). It's a popular tool that works really well with React Router.

First, install `create-react-app` and make a new project with it.

```
npx create-react-app demo-app
cd demo-app
```

Installation

You can install React Router from the [public npm registry](#) with either `npm` or `yarn`. In this guide, we'll use `react-router-dom` in this guide.

```
yarn add react-router-dom
```

Next, copy/paste either of the following examples into `src/App.js`.

API

Hooks

- `useHistory`
- `useLocation`
- `useParams`
- `useRouteMatch`

<BrowserRouter>

- `basename`: string
- `getUserConfirmation`: func
- `forceRefresh`: bool
- `keyLength`: number
- `children`: node

<HashRouter>

- `basename`: string
- `getUserConfirmation`: func
- `hashType`: string
- `children`: node

<BrowserRouter>

A `<Router>` that uses the HTML5 history API (`pushState`, `replaceState` and the `popstate` event) to keep your UI in sync with the URL.

```
<BrowserRouter>
  basename={optionalString}
  forceRefresh={optionalBool}
  getUserConfirmation={optionalFunc}
  keyLength={optionalNumber}
>
<App />
</BrowserRouter>
```

Using React in Firebase

React Ajax - Firebase 환경 설정

- Node.js 설치 확인
 - `node -v`
 - `npm -v`
- React 프로젝트 생성
 - `npx create-react-app .`
 - 불필요한 파일 프로젝트에서 삭제
- 파이어베이스 설정
 - 시작하기 > 프로젝트 만들기
 - 웹 앱에 파이어베이스 추가 > `firebaseConfig` 값 사용.
- `firebase` 모듈 설치 (8버전) - 9버전 이상은 ES6문법 사용.
 - `npm install firebase@8`

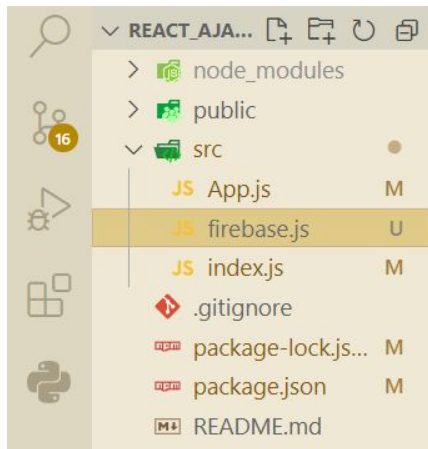


firebase.js 생성 (firebaseConfig)

```
import firebase from "firebase/app";
```

```
const firebaseConfig = {  
  apiKey: "#####",  
  authDomain: "#####",  
  projectId: "#####",  
  storageBucket: "#####",  
  messagingSenderId: "#####",  
  appId: "#####"  
};
```

```
export default firebase.initializeApp(firebaseConfig);
```



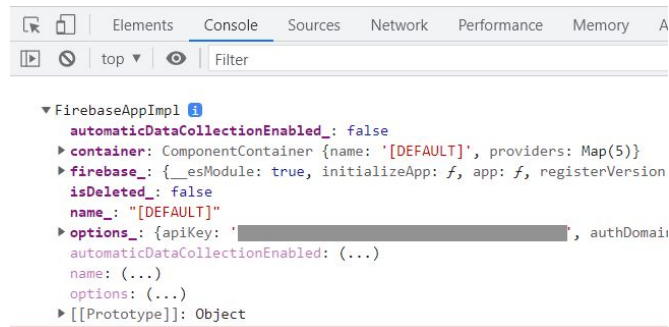
설정 테스트

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
import App from './App';
```

```
import firebase from "././firebase"  
console.log(firebase);
```

```
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(  
  <>  
    <App />  
  </>  
);
```

> npm start



React.js로 CRUD 구현

웹사이트 Create 기능 구현

Create product

[Read Products](#)

Name	<input type="text" value="I"/>
Description	<input type="text"/>
Price (\$)	<input type="text"/>
Category	<input type="text" value="Select category..."/>
<input type="button" value="Save"/>	

웹사이트 Update 기능 구현

Update product

Read Products

Name

Pillow VI



Description

This product will help you sleep at night.

Price (\$)

29.99

Category

Personal



Save Changes

웹사이트 Delete 기능 구현

Delete Product

Are you sure?

Yes

No

과정 실습

Weekly project : Web site cloning 2

- 지금까지 배운 내용을 활용 해서 원하는 사이트를 크론 코딩 합니다.
- 사이트를 똑같이 만들지 말고 자기 방식대로 만들어 봅니다.
- 기능 구현도 합니다.
- **CRUD** 기능이 모두 작동 하도록 합니다.
- 완성된 소스코드와 결과 화면 캡처를 학습 카페 과제함에 제출하세요.
- 가능한 자기 힘으로 하고 막히는 것은 주변 동료 or 조교 선생님에게 물어 봅니다.
- 프로젝트에 대해 궁금한 점은 전문강사님에게 질문 합니다.
- 행복하고 즐거운 코딩 시간!
- 참고1 : <https://www.notion.so/6edb3d3a694343e7865f866fd45036ee>
- 참고2 : <https://www.notion.so/Firestore-React-Bootstrap-23b983b73cf749cd92a474993a9b3749>
- 참고3 : <https://www.notion.so/Redux-toolkit-1f620375d2844b5791e5950a6426b635>
- 참고4 : <https://www.notion.so/Ract-js-Carousel-7f8632b32491402592effbcee8a5ce56>

리액트 미니 프로젝트 실습

- CRUD 통합 구현

Read Products

Create product

Name	Description	Price	Category	Action
Earphone	You need this one if you love music.	\$9.00	Personal	<button>Read</button> <button>Edit</button> <button>Delete</button>
Mouse	Very useful if you love your computer. Indeed.	\$11.35	Personal	<button>Read</button> <button>Edit</button> <button>Delete</button>
Trash Can	It will help you maintain cleanliness.	\$3.95	Sports	<button>Read</button> <button>Edit</button> <button>Delete</button>
Eye Glasses	It will make you read better.	\$6.00	Personal	<button>Read</button> <button>Edit</button> <button>Delete</button>
Gatorade	This is a very good drink for athletes.	\$1.99	Sports	<button>Read</button> <button>Edit</button> <button>Delete</button>
Basketball	A ball used in the NBA.	\$49.99	Sports	<button>Read</button> <button>Edit</button> <button>Delete</button>

Create product

Read Products

Name

Description

Price (\$)

Category

Select category...

Save

Update product

Read Products

Name

Description

Price (\$)

Category

Personal

Save Changes

Delete Product

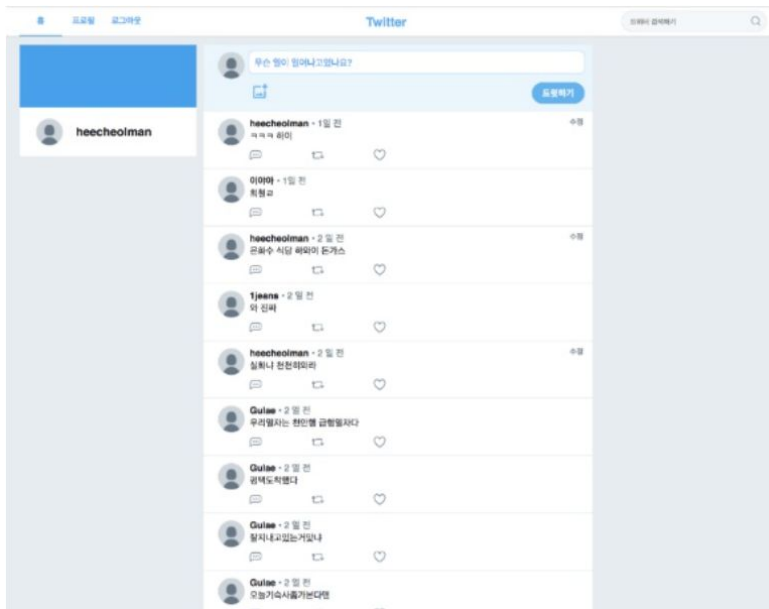
Are you sure?

Yes

No

React router 실습

- 트위터 클론 코딩 해보기



—

Do
it!

노마드 코더 니콜라스와 만드는 트위터 서비스

클론 코딩
트위터

코딩은 집짜를 만들어 보는 것!

하지만, 동적데이터가 없고, UI/UX를 완벽하게 구현한 서비스는 아닙니다



니콜라스와 함께 코딩



노마드 코더 니콜라스와 함께하는

이제 시작!

참고 서적

다음은 Backend!

גכ
ע
.