

React

리액트 작동원리와 기초

리액트

- 프론트엔드 프레임워크 중 하나
- 리액트(react), 앵귤러(angular), 뷰제이에스(vue.js)

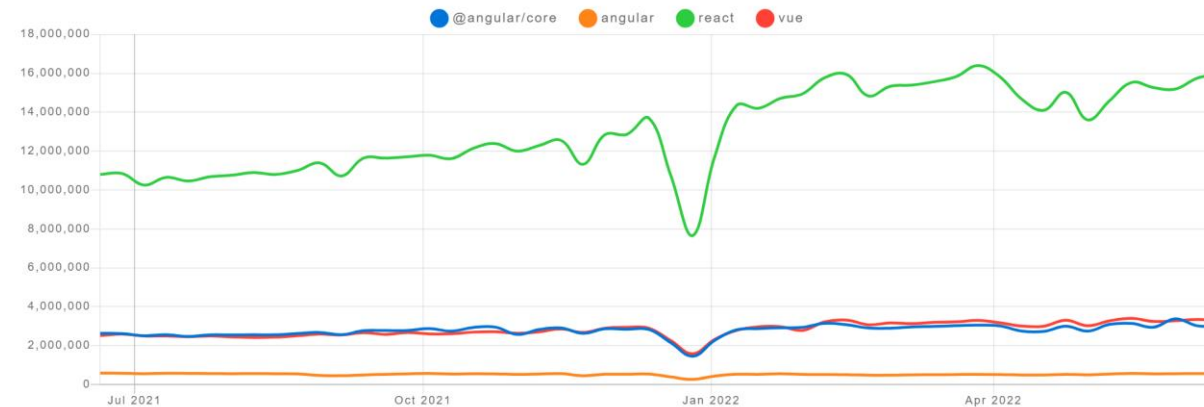
 npm trends

@angular/core vs angular vs react vs vue

Enter an npm package...

@angular/core x angular x react x vue x + ember-source

Downloads in past 1 Year ▾

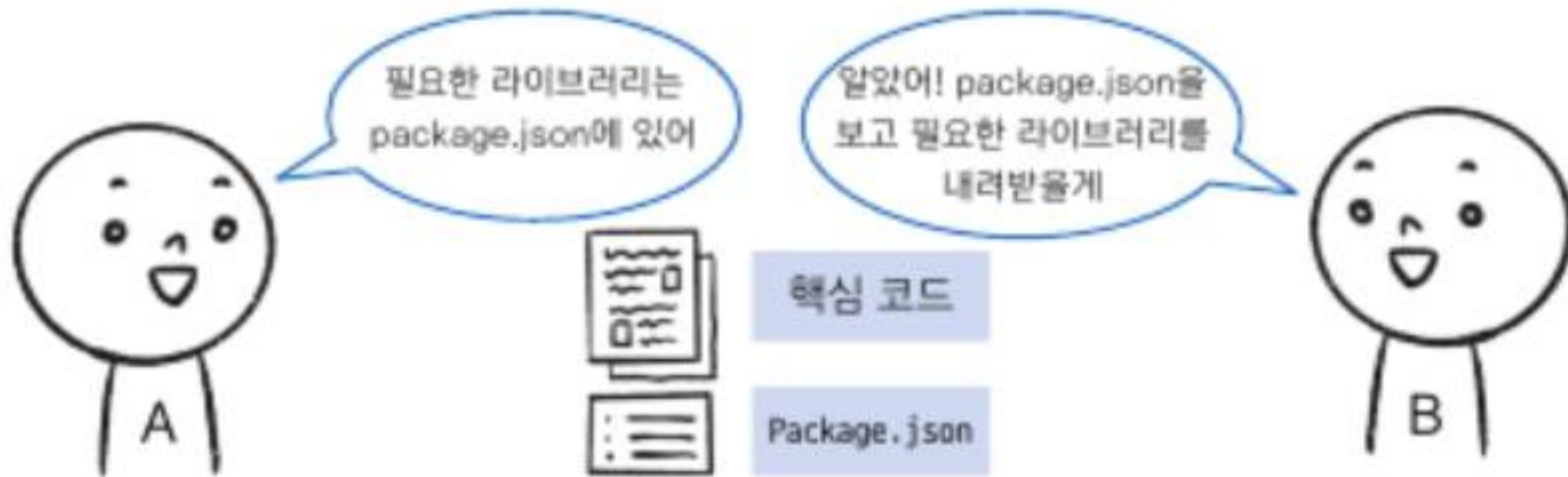


리액트 개념 및 장점

- 페이스북 개발에 사용한 기술
- 공개 소프트웨어
- 화면 출력에 특화된 프레임워크
- 컴포넌트를 조립하여 화면 구성
- 게임 엔진 원리를 도입하여 화면 출력 속도가 빠름(Virtual DOM)

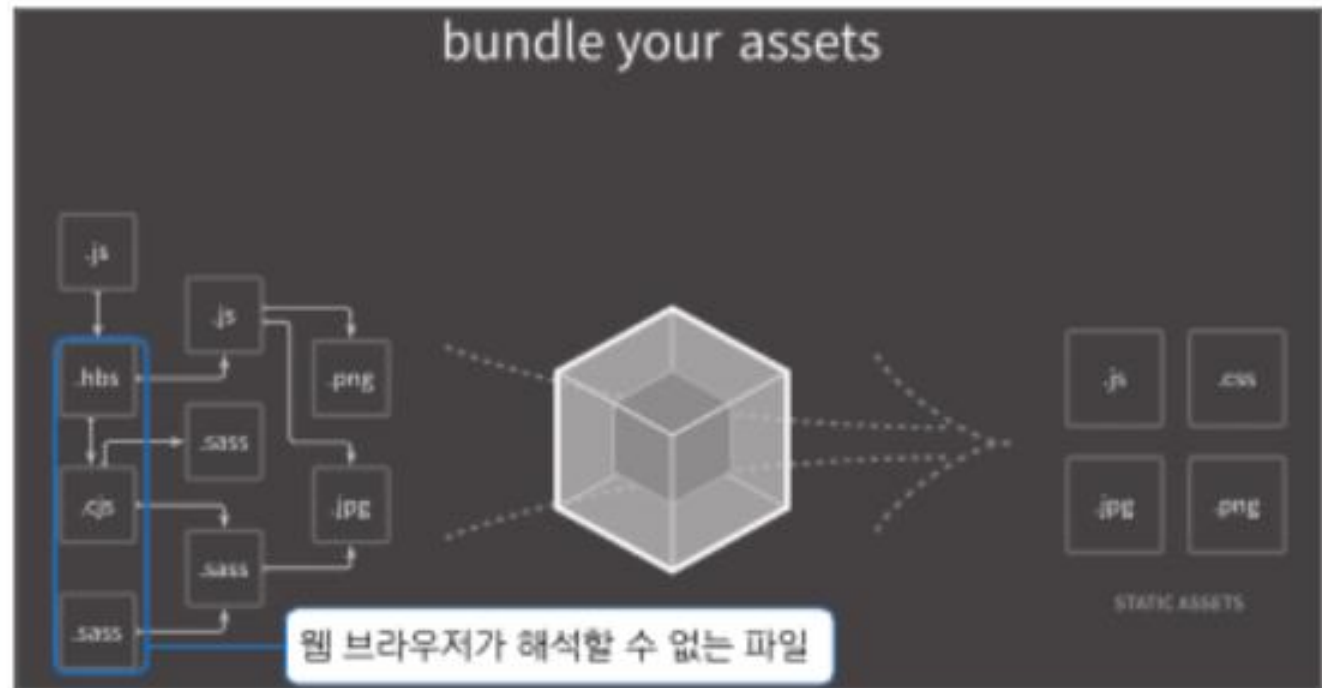
실습에 필요한 필수 도구

- 노드 패키지 매니저(npm) : 자바스크립트 라이브러리 관리 프로그램



실습에 필요한 필수 도구

- 웹팩(Web Pack) : 프로젝트에 사용된 파일을 분석하여 기존 웹 문서 파일로 변환하는 도구
- 간단한 노드 기반의 웹 서버 구동



웹팩의 작동 구조

리액트 개발 환경 설치

- Node.js (필수) : 확장성 있는 네트워크 애플리케이션 개발에 사용되는 소프트웨어 플랫폼이다. 작성 언어로 자바스크립트를 활용
- 에디터 (VS Code 등)
- 웹 브라우저 (파이어폭스 등)

Nodejs 설치하기

- <https://nodejs.org/en/>
- recommended for most users 설치
- <https://github.com/coreybutler/nvm-windows/releases>
- Nvm-setup.zip 다운로드
- 설치 후 명령프롬프트에서(관리자 권한으로 실행)
- >nvm -v
- > nvm install 16.15.1
- > nvm use 16.15.1
- > node -v
- **npm이란?** : Node Package Manager의 약자이다. NodeJS로 개발된 모듈들을 설치하고 관리해주는 패키지 매니저이다. 우리가 HTML CSS 수업에서 부트스트랩이 필요하면 가져다 썼던 것 처럼 node.js로 만들어진 유용한 모듈들을 가져다 쓰고 관리할 수 있다.

프로젝트 시작하기

- 프로젝트를 설치하고싶은 경로로 먼저 이동하기.
- `npx create-react-app` 폴더명
- Happy hacking 메시지 나오면 설치 완료
- `code .`으로 프로젝트 열기
- `npm start` 으로 프로그램 실행

SPA란?

Single Page Application의 준말 즉 페이지가 하나라는 뜻이다. 하나의 페이지 안에서 자바스크립트가 필요한 페이지나 정보만 동적으로 그려준다. 즉, 새 페이지로 넘어갈 때마다 서버에서 HTML CSS JS 소스를 가져올 필요가 없어 새로고침이 필요없다. 따라서 사용자에게 물흐르듯 끊김없는 웹 앱 경험을 선사해줄 수 있다.

단점은 처음에 필요한 리소스를 다 가져와야 해서 초기 구동 속도가 느리다.

리액트는 이런 SPA의 형식이다.

Visual Studio Code 플러그인 설치

- reactjs code snippets

리액트 파일 구조

- node_modules : 해당프로젝트에 대한 라이브러리가 저장
- public : 정적 파일을 넣는 디렉토리(html 파일, img등)
- Src폴더에 코드의 대부분이 들어 감(index.js, 리액트 컴포넌트 같은 js파일, css파일 등)
- Components : pages에서 사용할 컴포넌트들이 들어가 있다.
- pages : app.js 파일에서 react-router-dom을 사용해서 나눈 라우팅의 컴포넌트가 여기에 정리
- src/app.js 파일로 시작

컴포넌트

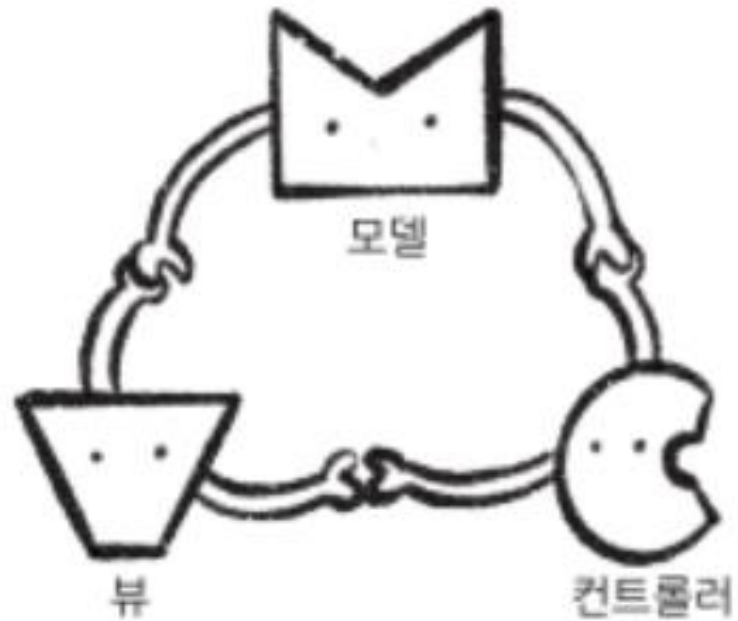
- JSX = JavaScript + XML
- 하나의 파일에 자바스크립트와 HTML을 동시에 작성

실습1 : JSX 사용

- App.js

```
<div>  
    
  <div>Hello World!</div>  
</div>
```

컴포넌트의 개념

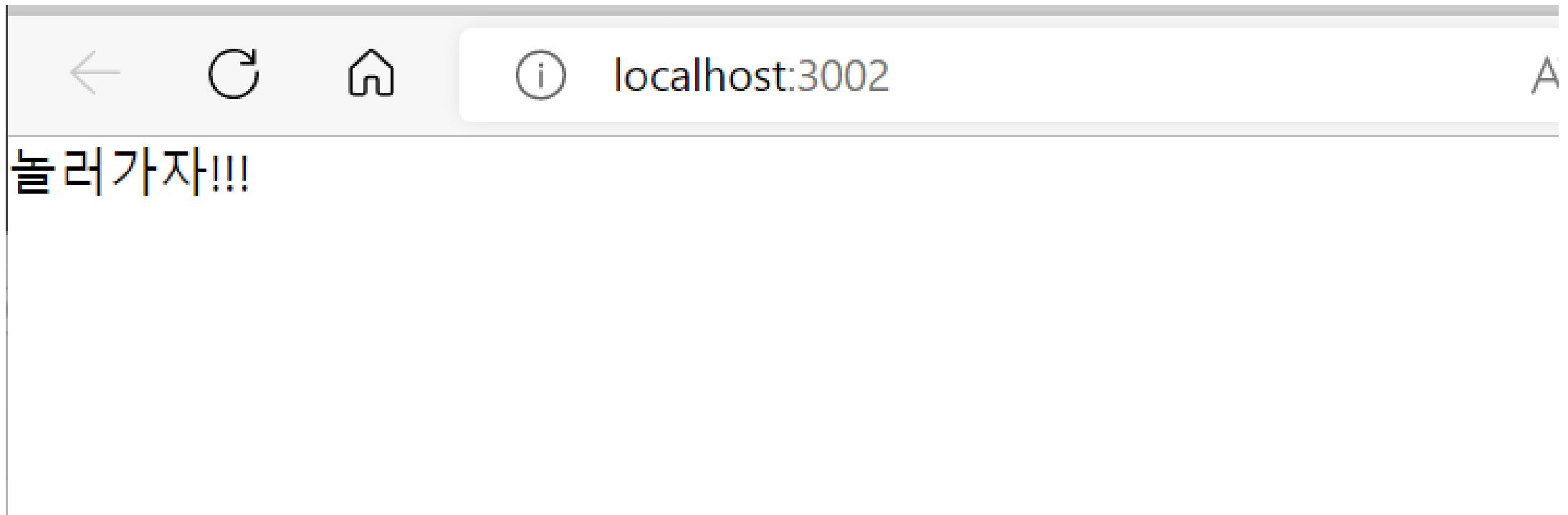


우리는 서로 꼭 붙어 있어야 정상적으로 작동해요



우리는 규격만 맞으면 서로 연결
할 수 있어요

실습2 : TodaysPlan 컴포넌트 만들기



컴포넌트 만들시 주의사항

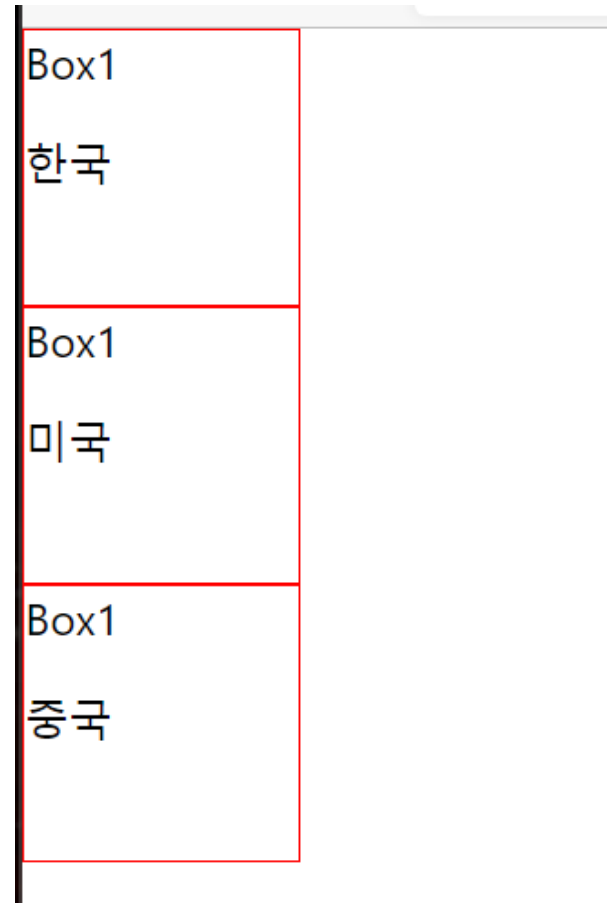
- 컴포넌트 이름은 반드시 대문자로 시작
- 컴포넌트와 일반 HTML 태그를 구별하는 방법은 태그가 대문자로 시작하는지 여부
- 소문자로 시작하면 일반 HTML 태그로 분류
- 반드시 첫 글자는 대문자로 함

실습3 box 만들기(컴포넌트 재활용하기)



Box1

한국



실습4 box값 변경

- props : 함수의 매개변수와 같은 존재

실습5 버튼클릭시 숫자 증가하기

- state 사용
- 특성 : 비동기

에어비앤비 디자인 시스템 따라하기

- 스토리북(Storybook) : UI컴포넌트 개발 도구
- 데모용 코드를 작성하는 데 도움을 주고 공통적으로 사용될 컴포넌트를 팀원들과 편리하게 공유하는 도구로 활용
- 구성단위는 스토리(Story)
- 하나의 UI컴포넌트는 보통 하나 이상의 Story를 가짐

장점

복잡한 로직 없이 독립적인 환경에서 컴포넌트를 개발할 수 있습니다.

재사용을 위한 컴포넌트들을 story에서 조합해 테스트할 수 있습니다.

컴포넌트들을 문서화 할 수도 있고 디자인 시스템에 적용해 피그마의 컴포넌트들과 동기화할 수 있습니다.

실습 프로젝트 생성

- CRA(Create React App)
- `npx create-react-app` 프로젝트 디렉터리
- `cd` 프로젝트 디렉터리
- `npx -p storybook sb init`
- `npm run storybook`

폴더구조

- main.js : stories를 위한 config 설정
- preview.js : 모든 story들에 글로벌하게 적용될 포맷 세팅

폴더구조

- main.js : stories를 위한 config 설정
- preview.js : 모든 story들에 글로벌하게 적용될 포맷 세팅

스토리 기본구조

Export default {

Title : 스토리북에 올릴 component 폴더 계층 구조,

Component : 스토리를 만들 컴포넌트 이름

}

Export const 스토리이름 = () => 해당스토리에서 테스트할 인자가 담긴 컴포넌트

실습6 text 컴포넌트 작성 및 story 연결

실습7 Input 컴포넌트 작성 및 story 연결

컨텍스트

- 어떤 컴포넌트 트리 내에서 최상위 컴포넌트 부터 최말단 컴포넌트에 걸쳐 전역(global)으로 관리해야 할 데이터가 필요
- React Context는 전역 데이터를 좀 더 단순하지만 체계적인 방식으로 접근할 수 있도록 도와줌

컨텍스트

- consumer로 Context 접근
- useContext로 Context 접근
- contextType으로 Context 접근

리덕스(redux)

- JavaScript 상태관리 라이브러리

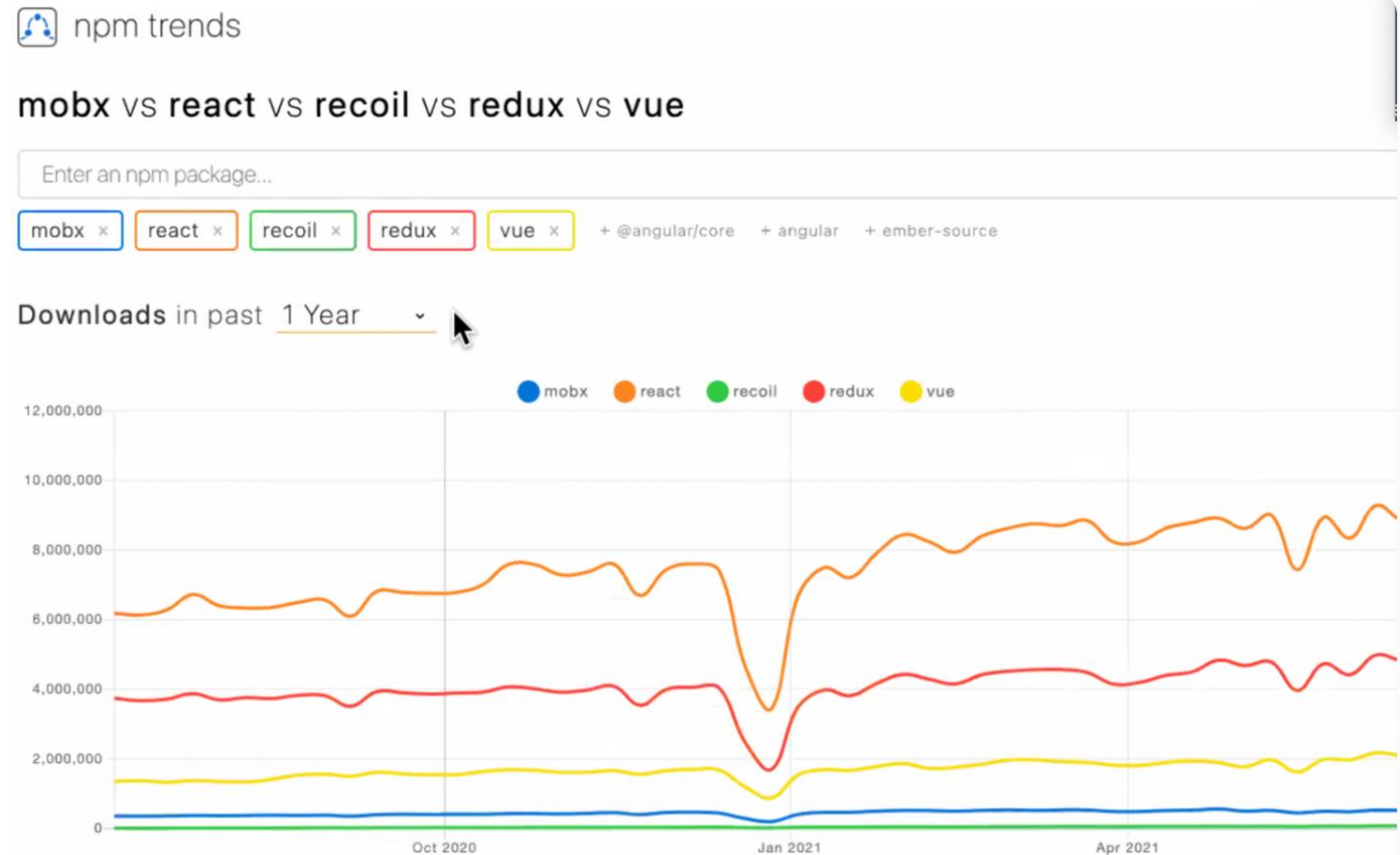
- Node.js 모듈

- 상태관리 툴 종류

- React Context

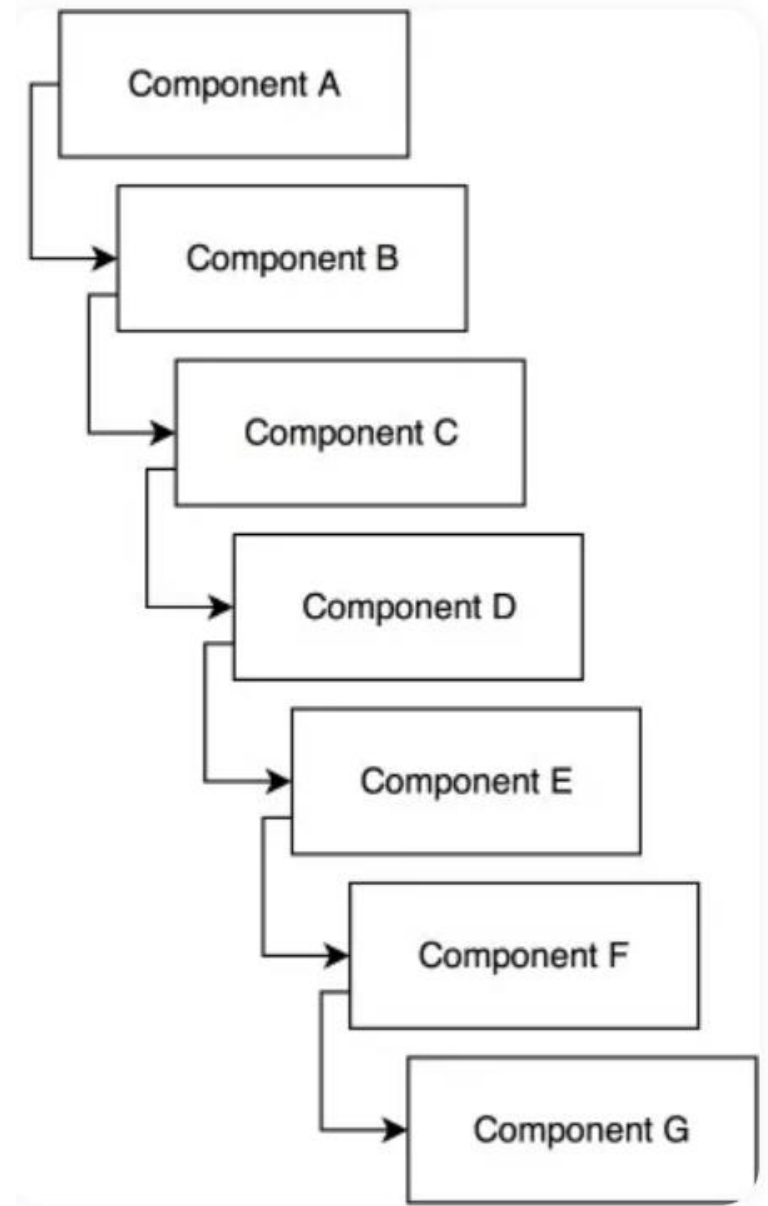
- Redux

- Mobx



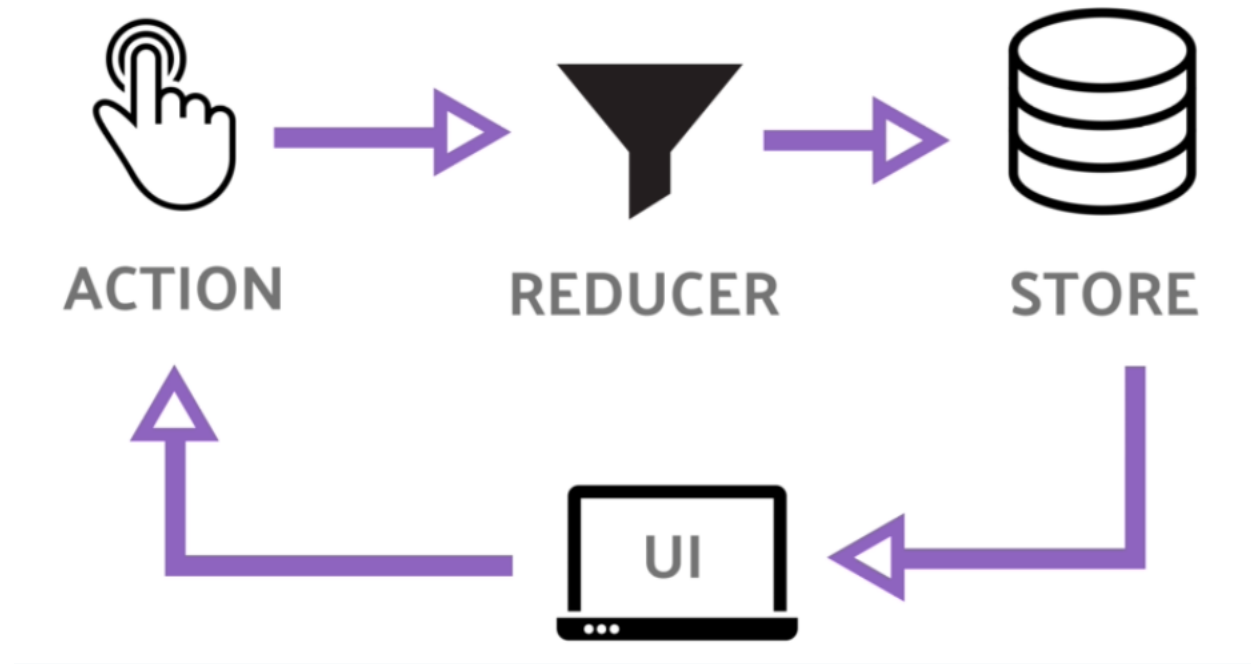
상태관리의 필요성

- 상태관리
 - React에서 State는 component 안에서 관리되는 것
- Component간의 정보 공유
 - 자식 컴포넌트들 간의 직접 데이터 전달은 불가능
 - 자식 컴포넌트들 간의 데이터를 주고 받을 때는 상태를 관리하는 부모 컴포넌트를 통해서 주고 받는다.
 - 자식이 많아진다면 상태 관리가 매우 복잡해진다.
 - 상태를 관리하는 상위 컴포넌트에서 계속 내려 받아야한다.
 - => Props drilling 이슈



상태관리의 필요성

- 전역 상태 저장소가 있고, 어디서든 해당 저장소에 접근할 수 있다면 이러한 문제는 해결



리덕스

- **Provider**란?

- Provider은 react-redux라이브러리 안에 있는 컴포넌트입니다.
- 리액트 앱에 스토어를 쉽게 연결하기 위한 컴포넌트 입니다.

- **combineReducer**란?

- redux모듈이 제공하는 함수이다.
- 만든 모든 리듀서들을 통합하여 하나의 리듀서로 쓰기 위한 함수이다

- **useSelector**란?

- redux의 state조회 (즉, 스토어에 있는 데이터들 조회)

- **useDispatch**란?

- 생성한 action 실행

리덕스

- `npm install redux react-redux`
- `react-redux`: 리액트 환경에 맞는 리덕스를 사용할 수 있게 해줌

실습

연락처

이름

이름을 입력해주세요

이름을 입력해주세요

찾기

전화번호

전화번호를 입력해 주세요

추가

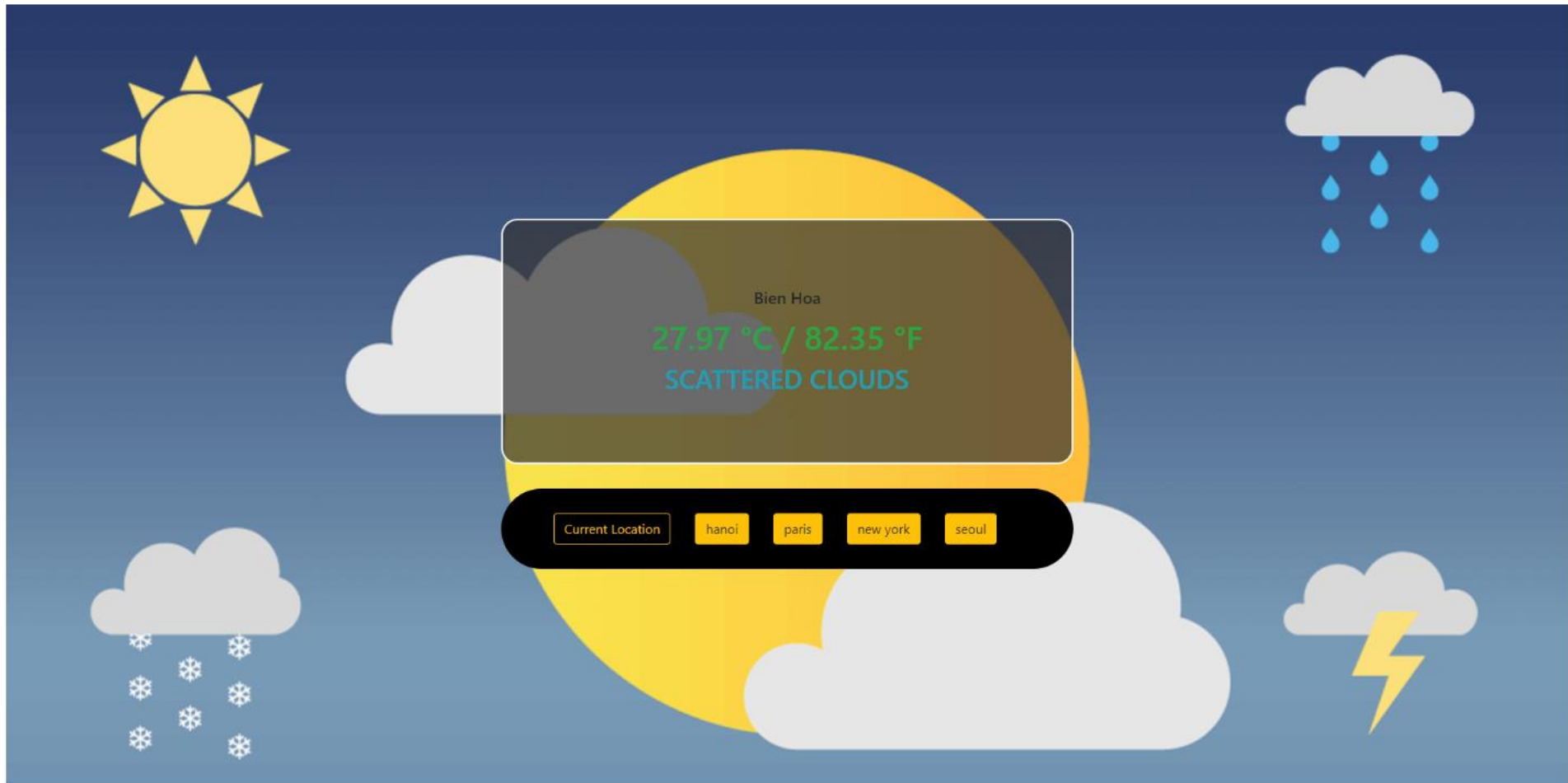
구현 과정

1. 왼쪽에는 연락처 등록하는 폼이, 오른쪽에는 연락처 리스트와 search창이 있다.
2. 리스트에 유저 이름과 전화번호를 추가할 수 있다.
3. 리스트에 아이템이 몇개 있는지 보인다.
4. 사용자가 유저를 이름검색으로 찾을 수 있다.

구현 하기

- Project 만들기
- `npx create-react-app projcontact`

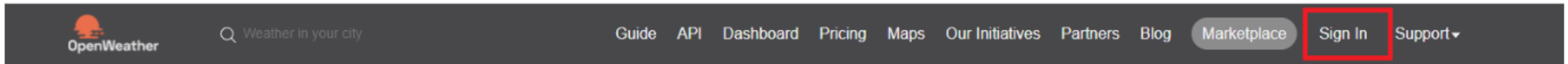
날씨 앱 만들기



회원 가입하기

- <https://openweathermap.org/api>

1. sign in 누르고 계정만들기 클릭



1. sign in 클릭

A screenshot of the 'Sign In To Your Account' form. The form has a title 'Sign In To Your Account' at the top. Below the title, there are two input fields: 'Enter email' and 'Password'. Below these fields is a checkbox labeled 'Remember me'. At the bottom of the form is a red 'Submit' button. Below the form, there are two links: 'Not registered? Create an Account.' and 'Lost your password? Click here to recover.'. The 'Create an Account.' link is highlighted with a red rectangular box.

2. 클릭

회원 가입하기

2.계정정보 입력하고 가입하기

Create New Account

bitna kim

codingnuna@gmail.com

.....

.....

We will use information you provided for management and administration purposes, and for keeping you informed by mail, telephone, email and SMS of other products and services from us and our partners. You can proactively manage your preferences or opt-out of communications with us at any time using Privacy Centre. You have the right to access your data held by us or to request your data to be deleted. For full details please see the OpenWeather [Privacy Policy](#).

- ☒ I am 16 years old and over
- ☒ I agree with [Privacy Policy](#), [Terms and conditions of sale](#) and [Websites terms and conditions of use](#)

정보 입력하고

체크

I consent to receive communications from OpenWeather Group of Companies and their partners:

- ☐ System news (API usage alert, system update, temporary system shutdown, etc)
- ☐ Product news (change to price, new product features, etc)
- ☐ Corporate news (our life, the launch of a new service, etc)

☒ I'm not a robot

reCAPTCHA
Privacy - Terms

체크

클릭

Create Account

회원 가입하기

2. 이메일 확인하기

3. API 키 확인하기

er in your city

GuideAPIDashboardPricingMapsOur InitiativesPartnersBlogMarketplacebitna...Support

Weather API



Home / Weather API

클릭

Please, [sign up](#) to use our fast and easy-to-work weather APIs for free. In case your requirements go beyond our freemium account conditions, you may check the entire list of our [subscription plans](#). You can read the [How to Start](#) guide and enjoy using our powerful weather APIs right now.

New ProductsServicesAPI keysBilling plansPaymentsBlock logsMy ordersMy profileAsk a question

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Status	Actions
3f9ded4f0dc79e25d98235ad8779b228	Default	Active	 

Create key

API key name

Generate

구현 하기

1. 앱이 실행되자마자 현재위치 기반의 날씨가 보인다.
2. 날씨정보에는 도시, 섭씨, 화씨, 날씨상태
3. 5개의 버튼이 있다.(1개는 현재위치, 4개는 다른도시)
4. 도시버튼을 클릭할 때마다 도시별 날씨가 나온다.
5. 현재위치 버튼을 누르면 다시 현재위치 기반의 날씨가 나온다.
6. 데이터를 들고오는 동안 로딩 스피너가 돈다.

현재 위치 가져오기

- Geolocation API
https://www.w3schools.com/html/html5_geolocation.asp
- 검색 get current location javascript
[HTML Geolocation API \(w3schools.com\)](https://www.w3schools.com/html/html5_geolocation.asp)
- 검색 react-spinner
[react-spinners - npm \(npmjs.com\)](https://www.npmjs.com/package/react-spinners)
- 검색 react-bootstrap
[React-Bootstrap · React-Bootstrap Documentation](https://react-bootstrap.github.io/)

동기와 비동기

- 동기

- 작업(task)들이 순차적으로 이루어 지는 것
- 다른 작업들을 blocking

- 비동기

- 작업(task)들이 순차적으로 이루어지지 않음
- 다른 작업들은 non-blocking
- Javascript를 사용한 비동기 통신 방식을 Ajax(Asynchronous Javascript and XML)
- Http요청(GET, POST), 이벤트 핸들러(click, over ...), setTimeout, setInterval

비동기

Callback

- 콜백 함수란 인자로 들어오는 함수를 칭함.
- 콜백은 비동기 통신을 할 수 있는 한 패턴이다.
- 문제점: 콜백 헬로 인한 `에러처리 불가`, `가독성 hell`

Promise

- ES6에서 나온 비동기 패턴.
- 비동기 통신 **결과**와 **상태**를 저장하는 객체.
- 후속처리 메서드로 `then(성공)`, `catch(에러)`, `finally(무조건)` 가 있다.

async/await

- Promise의 복잡성으로 인해 ES8에서 나온 비동기 패턴.
- Promise를 기반으로 하며, 완전히 같지는 않으나 사용하기 편하다.

stateful, stateless

컴포넌트를 나누는 또 다른 방식이다.

- stateful 컴포넌트 : 모든 state를 들고있는 컴포넌트
- stateless 컴포넌트: state를 안들고 있고 props 와 같이 받은 데이터만 보여주는 컴포넌트
- 리엑트는 단방향 통신이다 즉 부모에서 자식으로만 데이터를 전달 할 수 있다.

이를통해 데이터 흐름을 추적하기는 더 쉽지만, 같은 형제끼리 데이터를 주고받기는 힘들다. 그래서 데이터는 주로 부모가 들고있고 자식에게 전달해주는 형식이 된다. state를 들고있는 부모 컴포넌트는 stateful 컴포넌트, 부모가주는 데이터만 받는 컴포넌트를 stateless 컴포넌트라고 한다. 이렇게 컴포넌트를 구성하면 장점은 state가 없는 stateless컴포넌트는 재사용이 쉽고 데이터를 걱정하지 않아도 된다.

- 모든 중요한 데이터들은 stateful컴포넌트에 있기 때문에 유지보수가 쉽다 (stateful컴포넌트 하나만 주시하고 관리해주면 되니까)

그래서 최대한 모든 컴포넌트를 stateless로 만들고 몇개의 stateful컴포넌트에서 데이터를 관리하는 구조가 이상적이라고 리엑트는 말하고 있다.

React - One way Data Flow

- Data flows down
 - Props is immutable
 - States is mutable
- Event flows up
- Render by setState

