

React

리액트 작동원리와 기초

리액트

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

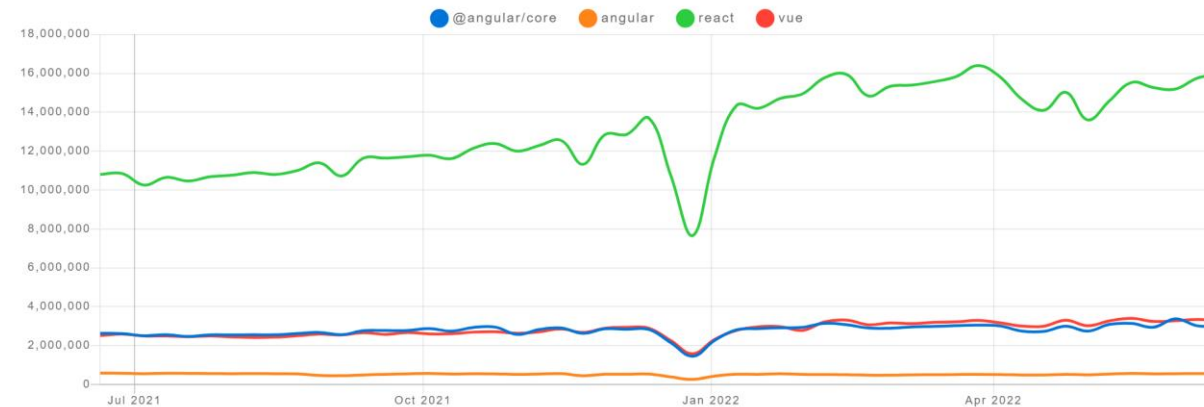
 npm trends

@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 ▾

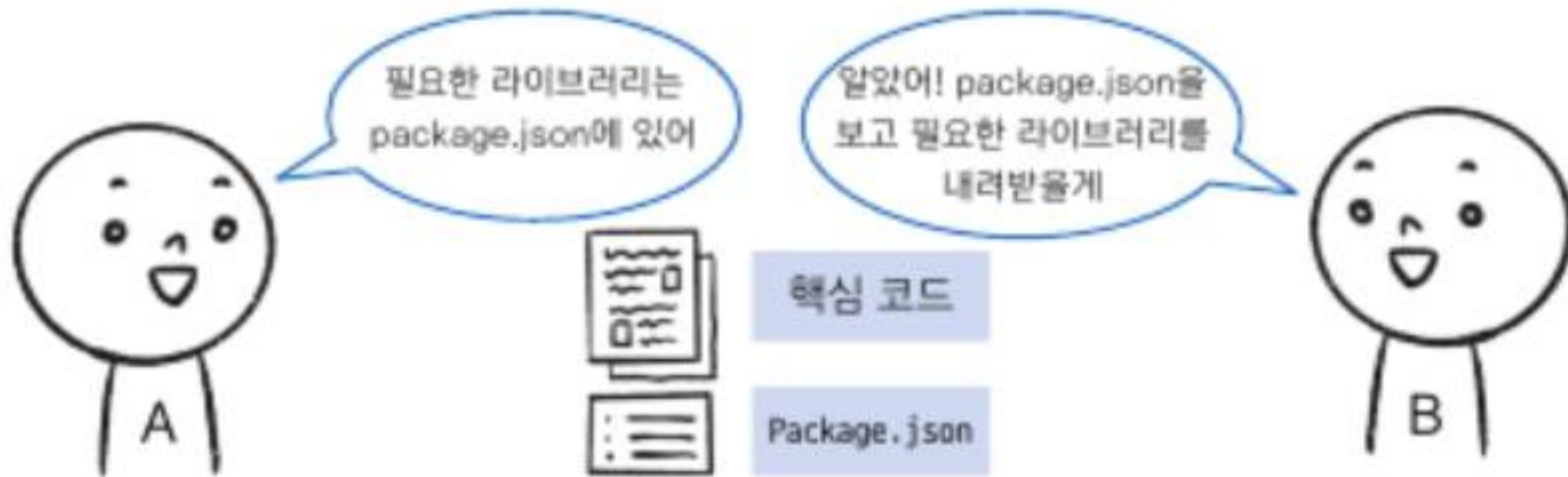


리액트 개념 및 장점

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

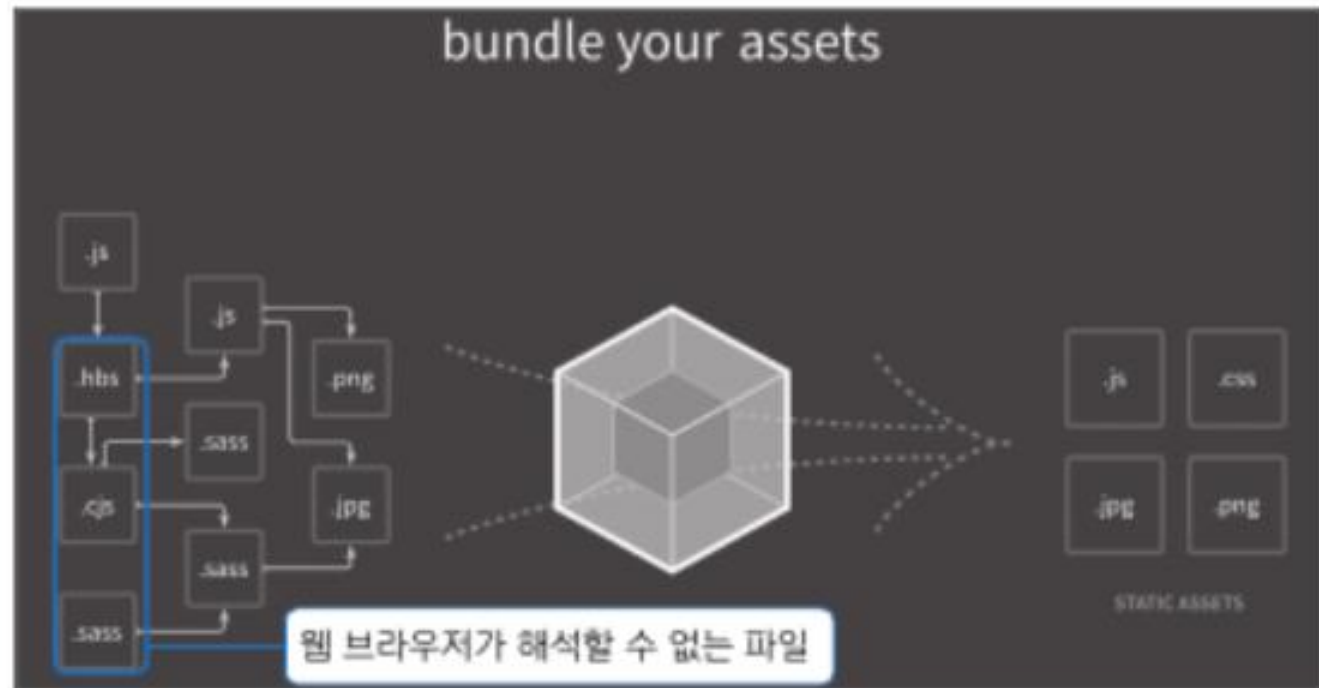
실습에 필요한 필수 도구

- 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.

리엑트는 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

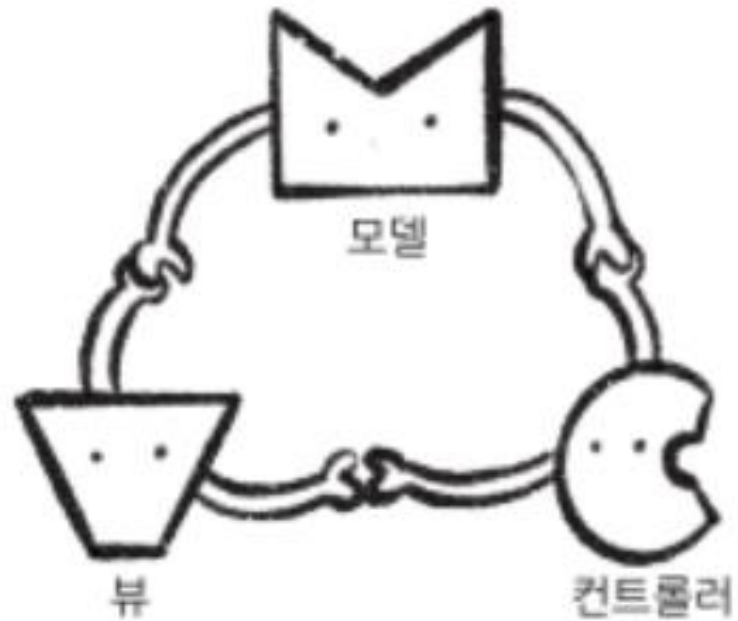
- JSX = JavaScript + XML
- 하나의 파일에 자바스크립트와 HTML을 동시에 작성

실습1 : JSX 사용

- App.js

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

컴포넌트의 개념

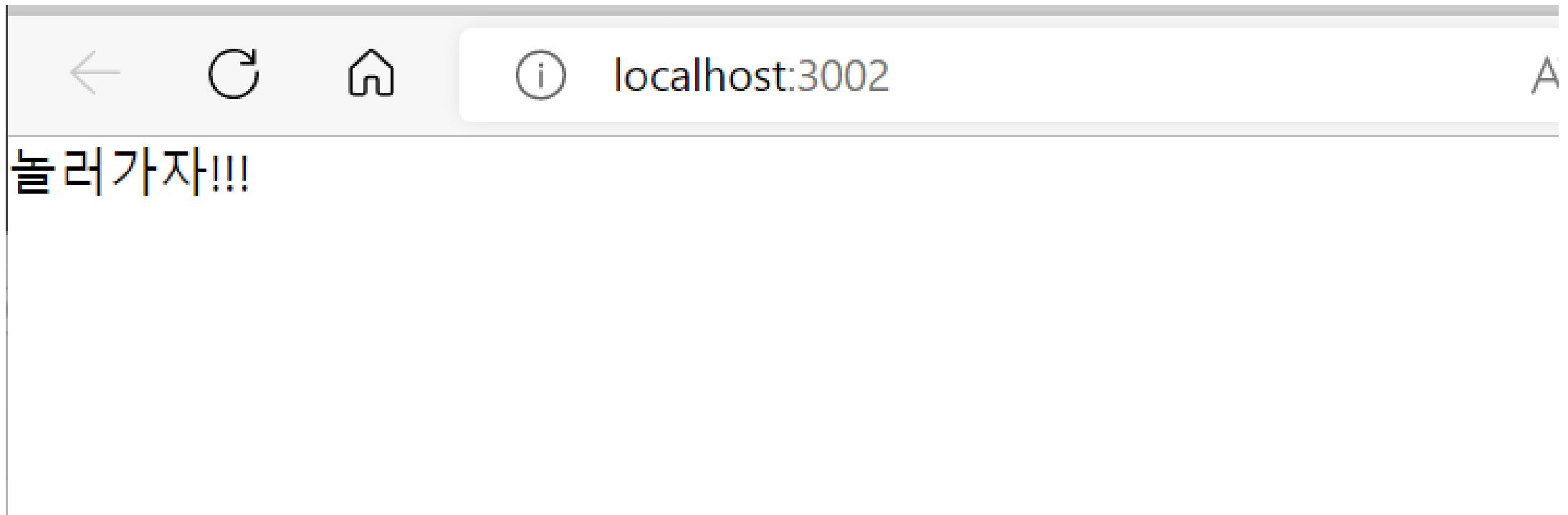


우리는 서로 꼭 붙어 있어야 정상적으로 작동해요



우리는 규격만 맞으면 서로 연결
할 수 있어요

실습2 : TodaysPlan 컴포넌트 만들기



컴포넌트 만들시 주의사항

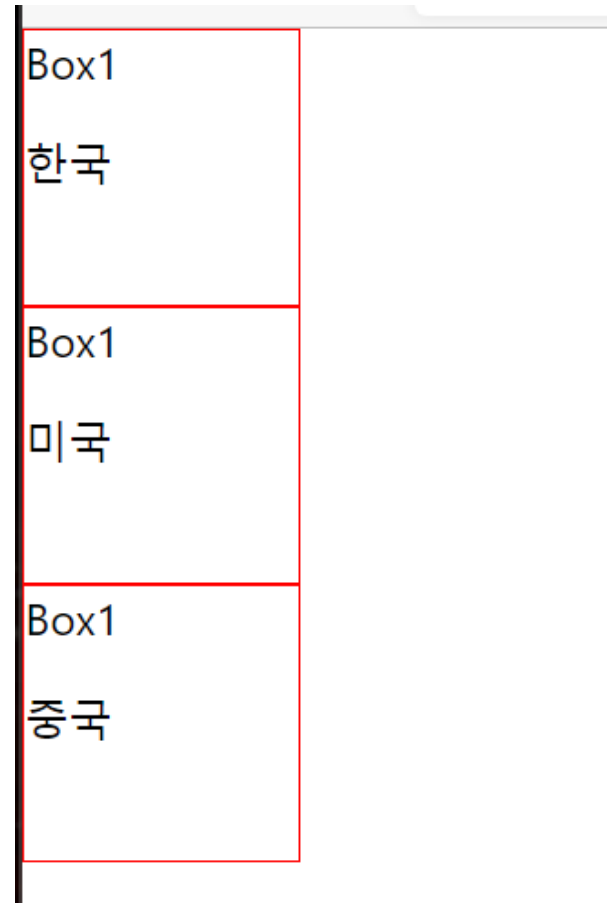
- 컴포넌트 이름은 반드시 대문자로 시작
- 컴포넌트와 일반 HTML 태그를 구별하는 방법은 태그가 대문자로 시작하는지 여부
- 소문자로 시작하면 일반 HTML 태그로 분류
- 반드시 첫 글자는 대문자로 함

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