React

# NPX VS NPM

## NPX

NPX 설치할 js파일 🡺 js파일을 설치한 후 실행한 뒤 제거한다.

장점 - 컴퓨터 메모리를 아낄 수 있다 + 항상 최신 버전 사용가능

ex) NPX crate-react-app .

## NPM

NPM 설치할 모듈 🡺 모듈을 컴퓨터에 설치한다.

장점 – 설치해두면 네트워크 없이도 개발 가능 + 컴퓨터 전역적으로 설치 및 사용가능

ex) > NPM install -g create-react-app

> create-react-app .

공통점 🡺 두 명령어 모두 Node.js로 만들어진 모듈들을 사용하기 위한 앱스토어같은 것

# 중요 개념

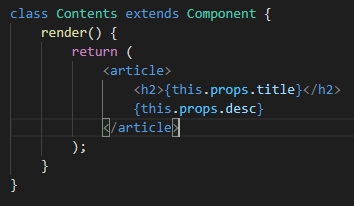
## Component

장점: 쉬운 정리정돈 & 유지보수 용이 & 재사용 가능

특징

함수나 클래스로 만들 수 있음

ex)



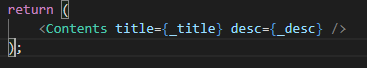
JSX 문법

[그림] Component 예제

### React Component는 Component클래스를 상속받아서 만들어진다.

### render()메서드를 오버라이딩 해야 한다.

사용자 정의 태그처럼 사용 가능 ex) <Subject/>



props는 외부에서 데이터 전달받음, state는 내부에서만 접근가능

### props

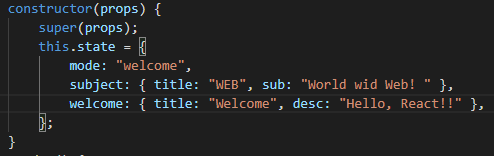
외부에서 데이터 보내기 🡺 <Contents title={\_title} desc={\_desc} />

title과 desc라는 props를 Contents 컴포넌트로 전달

내부 class에서는 props변수를 통해 참조(그림1 참고) 🡺 this.props.title(props명)

### state

외부에서는 접근이 불가능. 내부에서만(해당 클래스) 접근가능(get, set)하다.



초기설정 🡺 생성자에서 해야 함

this.state={}를 통해 가능

그냥 하자!! 안 하면 props 사용 불가능

[그림] state초기설정 방법

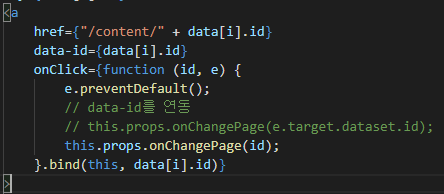
state값을 변경할 때는 반드시 this.setState()를 사용해야 한다.

this.state.mode=”welcome” 🡺 React가 변경된 것을 제대로 인식하지 못한다.



[그림] state 값 변경방법

Component에서 props를 통해 event를 입력받고 사용해보기



익명 이벤트 함수

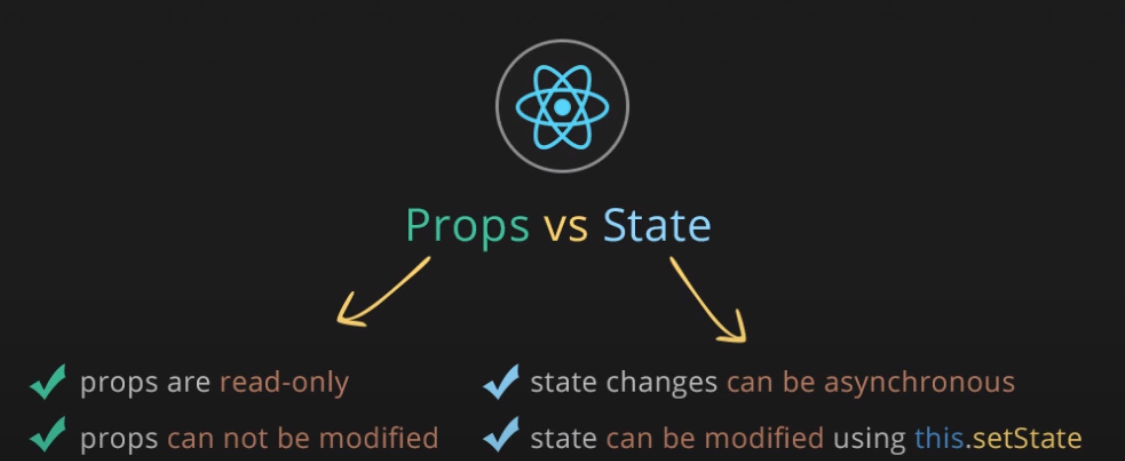
원래 이 익명 이벤트 함수에서 this는 null값

하지만 bind함수를 통해 this는 현재 컴포넌트가 된다.

또한 bind함수로 인해 data[i].id가

익명이벤트 함수의 첫 번째 인자로 들어오게 된다.

props를 통해 입력받은 함수 호출



[그림] props와 state 그림으로 보기

# 중요 모듈 및 명령어 정리

## create-react-app 폴더경로

react 개발환경을 구축해주는 모듈.

## npm run build(npx로는 불가능)

build폴더 안에 React프로젝트를 deploy를 위한 형태로 변환해준다.

## npx serve -s build

build폴더를 Document Root로 하여 웹서버가 구동하게 함 🡺 결론: 그냥 build를 실행

사용이유 - 같은 프로젝트를 실행했을 때 성능차이

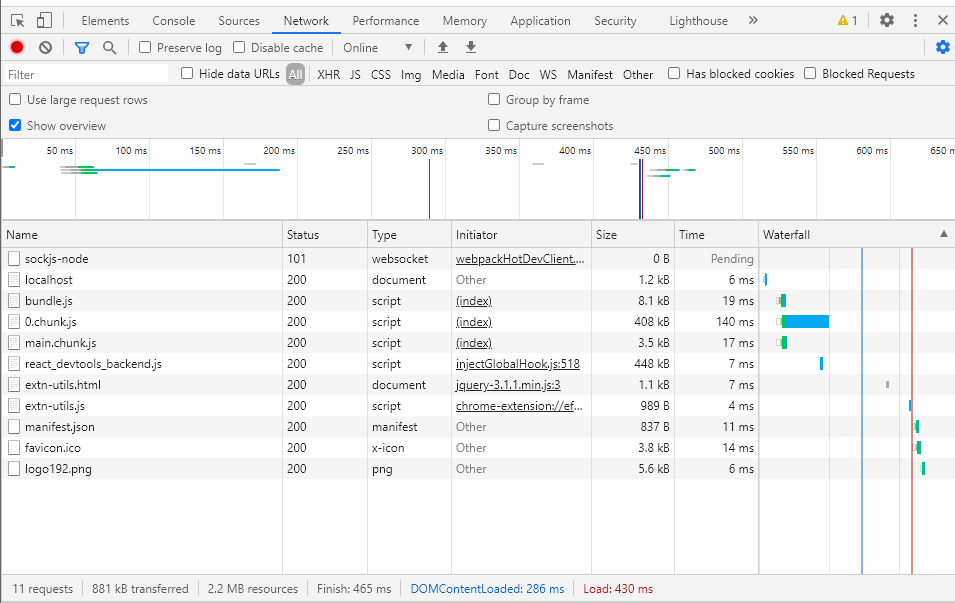
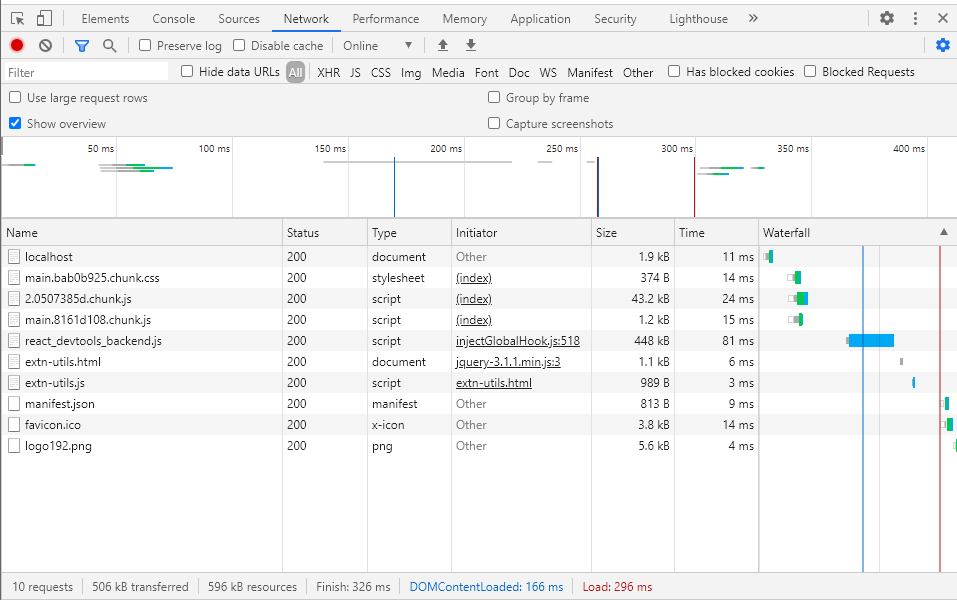
–

Figure 2 npx serve -s build로 실행 시 – 596kB

Figure npm start로 실행 시 - 2.2MB