

## 쇼피의 처음 만나 리액트



## 제 16장 맛보기



## Q & A

- 언제라도 질문하세요  
모르면 외우면 되지만 코딩 안되면 꼭 질문
  1. 대면수업에서 질문
  2. 한성e-class 질의응답게시판
- 강사의 1번 선생님은 여러분의 질문





# 아으로 나는 할 수 있다 (16장 맛보기 목표)

= 16장 맛보기이고 16장 자세한 내용은 15장 강의 후 학습함

1. 나는 **CRA**로 미니프로젝트 : 미니블로그 만들어서 실행할 수 있다
2. 나는 16.13 Production 빌드할 수 있다



## 미니블로그 만드는 과정

1. 16.2 CRA로 프로젝트 생성
2. 16.3 필요한 패키지 설치
3. 미니블로그 코드 복사
4. 16.12 미니블로그 실행
5. 16.13 Production 빌드



# 학생> 교수님~ 16장 맛보기는 왜 벌써 하나요?

교수> 학습할 때는 숲과 나무를 계속 같이 보면 이해하기 쉽습니다.

3장에서 15장까지는 나무를 만드는 과정이고

16장이 나무로 이루어진 숲입니다.

나무만 만들다 보면 왜 이걸 만드는 지 헤매기 쉽습니다.

16장 미니 프로젝트를 보면 3장~15장이 미니 프로젝트에 어떻게 활용되는 지 알게 됩니다.

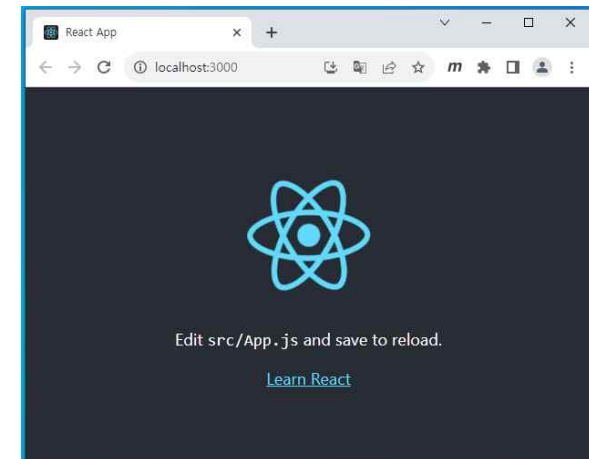
앞으로 강의시 나무와 숲을 같이 보면서 진행합니다



## 16.2 CRA로 프로젝트 생성

1. 16.2 CRA로 프로젝트 생성
2. 16.3 필요한 패키지 설치
3. 미니블로그 코드 복사
4. 16.12 미니블로그 실행
5. 16.13 Production 빌드

- 교재 466쪽 처럼 VScode 터미널에서  
C:\ws\_react 폴더에서 `$npx create-react-app mini-blog` 실행
- 교재 467쪽처럼 Vscod 터미널에서  
`$ cd mini-blog`  
`$ npm start`  
React App 시작됨





## 16.3 필요한 패키지 설치

1. 16.2 CRA로 프로젝트 생성
2. 16.3 필요한 패키지 설치
3. 미니블로그 코드 복사
4. 16.12 미니블로그 실행
5. 16.13 Production 빌드

- 교재 470쪽 처럼 VScode 터미널에서

C:\ws\_react\mini-blog 폴더에서

\$npm install --save react-router-dom@6.26.2 styled-components@6.1.13 실행  
(여기서 save앞에 - 가 2개임)

- 교재 471쪽 처럼 package.json파일에서 설치 패키지 확인

```
package.json X
mini-blog > {} package.json > ...
1  {
2    "name": "mini-blog",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^13.4",
7      "@testing-library/react": "^13.4",
8      "@testing-library/user-event": "^13.2",
9      "react": "^18.3.1",
10     "react-dom": "^18.3.1",
11     "react-router-dom": "^6.26.2",
12     "react-scripts": "5.0.1",
13     "styled-components": "^6.1.13",
14     "web-vitals": "^2.1.4"
15   },
```



# 미니블로그 코드 복사

1. 16.2 CRA로 프로젝트 생성
2. 16.3 필요한 패키지 설치
3. 미니블로그 코드 복사
4. 16.12 미니블로그 실행
5. 16.13 Production 빌드

- 한성e-class 공지사항 게시판에서 mini-blog 다운로드
- C:\ws\_react\mini-blog\src 폴더 삭제
- 공지사항 게시판의 src 폴더를 c:\ws\_react\mini-blog 로 복사

크 (C:) > ws\_react > mini-blog

이름

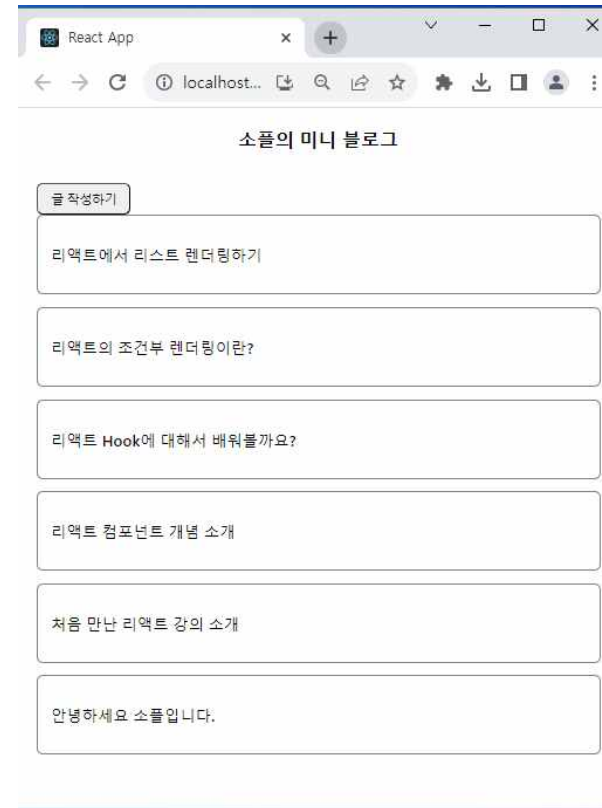
node\_modules  
public  
src  
.gitignore  
package.json  
package-lock.json  
README.md





# 16.12 미니블로그 실행

1. 16.2 CRA로 프로젝트 생성
  2. 16.3 필요한 패키지 설치
  3. 미니블로그 코드 복사
  4. **16.12 미니블로그 실행**
  5. 16.13 Production 빌드
- 교재 504쪽 처럼 VScode 터미널에서  
C:\ws\_react\mini-blog 폴더에서  
\$npm start 실행





## 16.13 Production 빌드 (1 / 4)

1. 16.2 CRA로 프로젝트 생성
2. 16.3 필요한 패키지 설치
3. 미니블로그 코드 복사
4. 16.12 미니블로그 실행
5. 16.13 Production 빌드

```
mini-blog
├── build
│   └── node_modules
├── public
├── src
├── .gitignore
├── package-lock.json
├── package.json
└── README.md
```

- Vscode 터미널에서 `$npm start` 종료 ( `ctrl-C` 입력)
- 교재 509쪽 처럼 VScode 터미널에서  
C:\ws\_react\mini-blog 폴더에서  
`$npm run build` 실행
- Build 폴더 생성 확인
- `$ serve -s build` 실행 (여기서 `-s` 옵션은 서빙 폴더)  
명령어가 없다고 나오면 `serve` 설치  
`$ npm install -g serve`



## 16.13 Production 빌드 (2/4)

학생> 보안 오류 나는데요

교수>

PS (Power Shell)을 Command Prompt로 변경

PS 삭제

C:\ws\_react\mini-blog 폴더로 변경

\$ serve -s build 실행

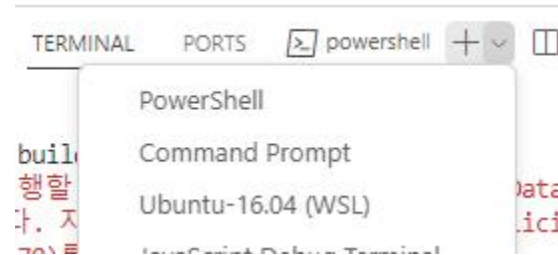
크롬 URL창에서 localhost:3000 입력

종료는 ctrl-C 입력

\$ npm start와 동일한 결과 확인

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\ws_react\mini-blog> serve -s build
serve : 이 시스템에서 스크립트를 실행할 수 없으므로 C:\ws_react\mini-blog\serve.ps1 파일을 로드할 수 없습니다. 자세한 내용은 al
o.microsoft.com/fwlink/?LinkID=135170를 참조하십시오
위치 줄:1 문자:1
+ serve -s build
+ ~~~~~
+ CategoryInfo          : 보안 오류: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```





## 16.13 Production 빌드 (3/4)

학생> production 빌드는 왜 하나요?

교수> 개발 단계에서는 **CRA** 방식으로 **jsx** 코딩을 하고  
고객에게 웹서비스를 할 때는 **build**폴더 내용을 웹서버에 업로드합니다  
이 단계를 **deploy**(배포)단계라고 합니다

Build 폴더 내용을 보면 **JSX**코드는 없고

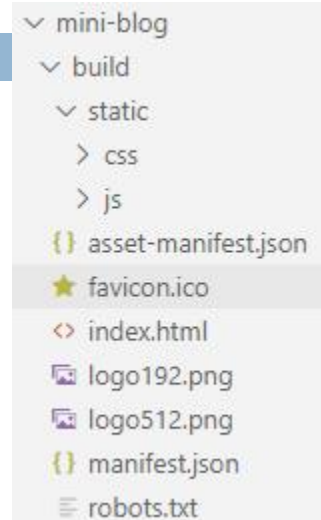
HTML파일은 **index.html**파일 1개 (SPA)

JS파일과 CSS파일로 구성되어 있습니다.

특히 **CRA**에서는 html파일이 없었는데 **\$npm run build**하면서 생겼습니다.

Index.html을 Vscode로 보면 보기 어렵기에

F12로 보면 tag도 root div 태그 1개만 있는 것을 확인



```
(index) X
1  <!doctype html>
  <html lang="en">
    <head>
      <meta charset="utf-8"/>
      <link rel="icon" href="/favicon.ico"/>
      <meta name="viewport" content="width=device-
      <meta name="theme-color" content="#000000"/>
      <meta name="description" content="Web site c
      <link rel="apple-touch-icon" href="/logo192.
      <link rel="manifest" href="/manifest.json"/>
      <title>React App</title>
      <script defer="defer" src="/static/js/main.e
      <link href="/static/css/main.e6c13ad2.css" r
    </head>
    <body>
      <noscript>You need to enable JavaScript to r
      <div id="root"></div>
    </body>
  </html>
2
```

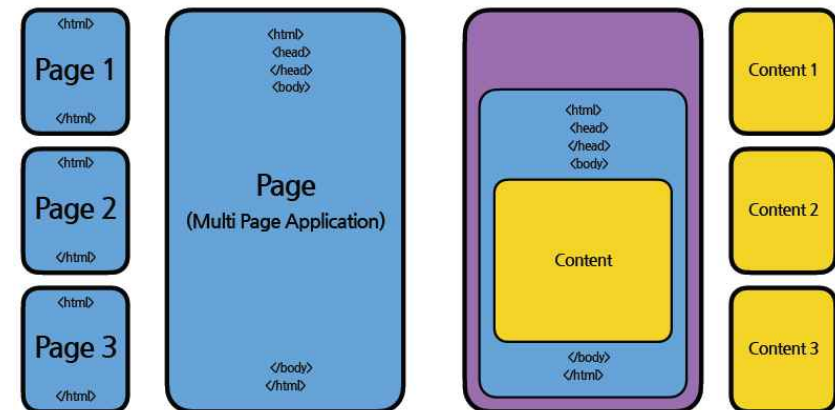
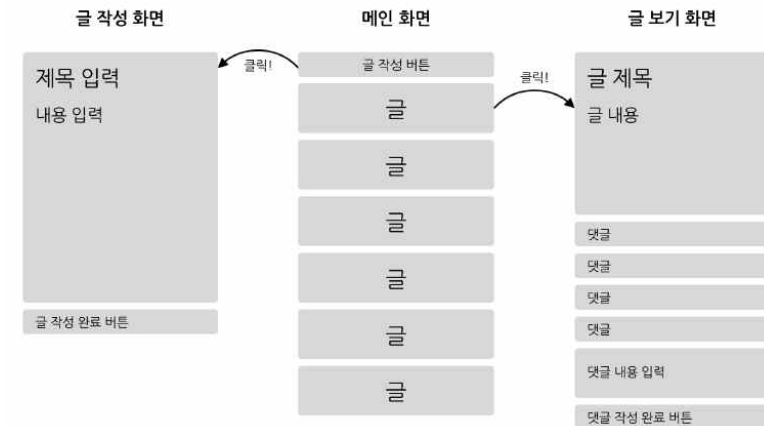


## 16.13 Production 빌드 (4/4)

- Mini-blog는 3개의 화면으로 구성됨
- 교재 31쪽의 MPA와 SPA를 비교하면 Mini-blog를 MPA방식으로 구현하면 html파일 최소 3개 SPA방식은 html파일 1개
- 특히 SPA방식은 tag도 root div 태그 1개만 사용

학생> 웹기초에서는 MPA와 여러 개의 태그를 사용했는데 react 웹은 SPA방식이라 상당히 다르네요

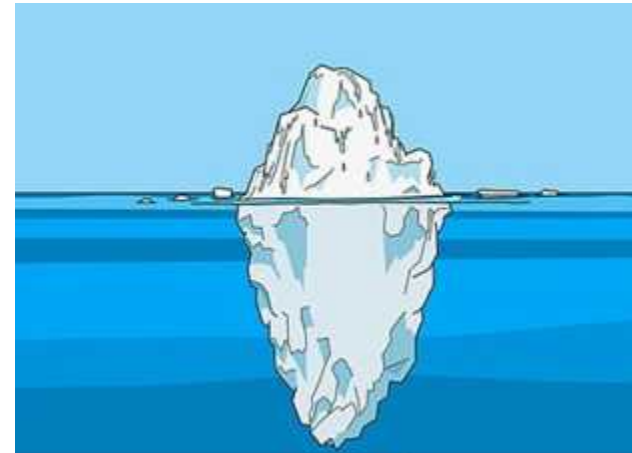
교수> 네. 앞으로 16장에서 여러 개의 화면을 어떻게 SPA방식으로 구현하는지 자세히 설명하겠습니다.





# 빙산의 일각

- CRA방식의 경우 JSX파일들을 Babel, Webpack등을 사용하여 크롬에 전달되고 있습니다
- 크롬 화면에 보이는 것은 빙산의 일각이기에 앞으로 강의 하면서 빙산의 보이지 않는 부분도 설명하겠습니다.
- 여러분의 질문이 빙산의 전체를 파악하는데 중요한 출발점입니다. 많이 질문해주세요





# 이제 나는 할 수 있다

1. 나는 CRA로 16장 미니프로젝트 : 미니블로그 만들어서 실행했다
2. 나는 16.13 Production 빌드했다
3. 나는 16장 숲을 만들었기에 앞으로 3장~15장 나무들을 만들면서 16장 숲에 어떻게 활용되는지 계속 살펴보겠다.