첫 프로젝트

**학습목표**

리액트 프로젝트 생성

테스트용 데이터 만들기 함수 구현

**목차**

[1. 첫 프로젝트 2](#_Toc145225972)

[1) 리액트 프로젝트 생성 2](#_Toc145225973)

[2) 프로젝트 열기 3](#_Toc145225974)

[2. 프로젝트 구조 4](#_Toc145225975)

[1) package.json 파일 4](#_Toc145225976)

[2) 웹팩 (webpack) 6](#_Toc145225977)

[3) index.html 파일 9](#_Toc145225978)

[4) 소스코드 수정해 보기 10](#_Toc145225979)

[5) 패키지 설치 명령 11](#_Toc145225980)

# 첫 프로젝트

## 리액트 프로젝트 생성

### 리액트 프로젝트 생성 명령

|  |
| --- |
| npx create-react-app 프로젝트이름 --template typescript |

터미널 창에서 위 명령을 실행하면, 현재 디렉토리에 프로젝트 폴더가 생성되고 프로젝트 파일이 생성된다.

npx는 javascript로 구현된 명령을 다운로드하여 실행해 주는 도구이다.

위 명령에 의해서 create-react-app 패키지 최신 버전이 다운로드 되고 실행된다.

따라서 위 명령을 처음 실행하면, create-react-app 패키지를 설치할 것인지 묻는다.

### ex01 프로젝트 생성

|  |
| --- |
| npx create-react-app ex01 --template typescript |

|  |
| --- |
|  |

y 엔터

|  |
| --- |
|  |

ex01 프로젝트 생성 완료

## 프로젝트 열기

생성된 ex01 폴더를 Visual Studio Code에서 열자

메뉴: File - Open Folder

# 프로젝트 구조

## package.json 파일

A screenshot of a computer

Description automatically generated

모든 자바스크립트 프로젝트의 기본 설정 파일은 package.json 이다.

자바스크립트 프로젝트의 루트 폴더에 이 파일이 있어야 한다.

이 파일의 내용은 다음과 같다.

### name

프로젝트 이름

### version

프로젝트 버전

### dependencies

프로젝트가 사용하는 패키지(라이브러리) 이름과 버전 목록.

Spring Boot 프로젝트 pom.xml 파일의 dependency 태그 목록에 해당한다.

dependency 패키지들은 node\_modules 폴더 아래에 다운로드된다.

따라서 node\_modules 폴더의 크기는 상당하다.

프로젝트 폴더를 압축할 때, 먼저 node\_modules 폴더를 삭제해야 한다.

다음 명령에 의해서 dependency 패키지들을 다시 전부 다운로드하여 설치할 수 있기 때문이다.

|  |
| --- |
| npm install |

위 명령을 아래처럼 간단히 입력해도 된다.

|  |
| --- |
| npm i |

### scripts

프로젝트를 빌드하거나 실행하기 위한 명령 목록이다.

npm start : 개발하기 위해 프로젝트 실행하기 위한 명령

npm run build : 프로젝트를 빌드하여 배포 파일 생성하기 위한 명령

## 웹팩 (webpack)

자바 스크립트 프로젝트의 가장 대표적인 빌드 도구가 웹팩이다.

웹팩은 프로젝트의 소스코드 파일들을 기본적인 자바스크립트 문법으로 트랜스파일하고,

트랜스파일된 파일들을 묶어서 빌드 결과 파일 즉 배포 파일을 만들어 준다.

리액트로 개발된 프론트엔드 앱은 웹 브라우저에서 실행되어야 하므로,

프로젝트 빌드 결과 만들어진 배포 파일이 웹브라우저에 다운로드되어 실행된다.

이 빌드 과정을 번들(bundle) 이라고도 한다.

번들은 묶음, 묶음을 만든다는 뜻인데,

많은 소스코드 파일들을 묶어서 소수의 배포 파일을 만드는 것을 의미한다.

그래서 웹팩을 번들러(bundler), 프로젝트 빌드 결과물을 번들(bundle) 이라고도 부른다.

Shape

Description automatically generated with medium confidence

### 빌드 해보기

다음 명령을 실행하면 웹팩에 의해 프로젝트가 빌드된다

|  |
| --- |
| npm run build |

위 명령을 실행하면 프로젝트 루트 디렉토리 아래 build 디렉토리에 빌드 파일들이 생성된다.

프로젝트 개발이 끝나고 배포할 파일을 만들 때 위 명령을 실행하면 된다.

|  |
| --- |
|  |

노란색 부분: 빌드 결과 만들어진 배포 파일들

빨간색 부분: 배포 파일들을 node.js에서 실행하기 위한 명령

### 개발 모드로 프로젝트 실행하기

개발 모드로 프로젝트를 실행하려면, 빌드할 필요 없이 그냥 다음 명령만 실행하면 충분하다.

다음 명령에 의해서 소스코드 파일들이 번들되어 바로 실행된다.

|  |
| --- |
| npm start |

위 명령에 의해서 구동되는 것은 웹팩 웹서버이다.

A picture containing diagram

Description automatically generated

Graphical user interface, application

Description automatically generated

### 강제 종료

npm start 명령은 저절로 종료되지 않고 계속 실행된다.

따라서 실행을 종료하려면, 터미널에서 Ctrl+C 키를 눌러 강제 종료하거나, 터미널 창을 닫아야 한다.

그리고 npm start 명령이 실행 중인 터미널 창에서 다른 명령을 실행할 수 없기 때문에,

터미널 창을 하나 더 열어야 한다.

|  |
| --- |
|  |

npm start 명령이 실행중인 터미널 창

빨간색 버튼을 클릭하면, 새 터미널 창이 열린다.

파란색 버튼을 클릭하면 현재 터미널 창이 닫히고 여기서 실행중인 명령도 강제 종료된다.

## index.html 파일

public/index.html

A screenshot of a computer

Description automatically generated

번들된 소스 코드들은 빨간색으로 칠한 div 태그에 삽입되어 실행된다.

자바의 main 메소드에 해당하는 것이 src/index.tsx 파일이고,

src/index.tsx 파일에서 src/App.tsx 파일을 호출한다.

## 소스코드 수정해 보기

### src/App.tsx 파일 수정

|  |
| --- |
| import React from 'react';  function App() {    console.log('App() 함수 호출됨');    return (      <div>        <h1>안녕하세요 리액트</h1>      </div>    );  }  export default App; |

개발 모드로 프로젝트를 실행하면, 수정된 소스코드가 즉시 실행에 반영된다.

이 기능을 핫 리로딩(Hot Reloading) 또는 핫 모듈 교체(Hot Module Replacement) 이라고 한다.

아주 가끔 핫 리로딩 기능이 작동하지 않을 수 있다고 한다.

그럴 때는 웹페이지를 새로고침 하면 된다고.

Graphical user interface, text, application

Description automatically generated

console.log('App() 함수 호출됨');

이 메소드로 값을 출력하는 것은 웹브라우저 콘솔 창에 출력된다.

Graphical user interface, text, application, email

Description automatically generated

이렇게 콘솔 창에 어떤 값을 출력하는 것은 디버딩할 때 유용하다.

## 패키지 설치 명령

### 실행에 필요한 패키지 설치 명령

실행에 필요한 패키지(라이브러리)를 설치하는 명령은 다음과 같다.

|  |
| --- |
| npm install 패키지명 |

위 명령을 아래처럼 간단히 입력해도 된다.

|  |
| --- |
| npm i 패키지명 |

실행에 필요한 패키지들은 package.json 파일의 depedencies 항목에 등록된다.

### 개발에만 필요한 패키지 설치 명령

실행할 때에는 필요 없고 개발할 때만 필요한 패키지를 설치하는 명령은 다음과 같다.

|  |
| --- |
| npm install --save-dev 패키지명 |

위 명령을 아래처럼 간단히 입력해도 된다.

|  |
| --- |
| npm install -D 패키지명 |

|  |
| --- |
| npm i -D 패키지명 |

개발할 때만 필요한 패키지들은 package.json 파일의 devDepedencies 항목에 등록된다.

### node\_modules 디렉토리

npm 명령에 의해서 다운로드 되고 설치된 패키지들은 node\_modules 디렉토리 아래 저장된다.

다음 명령을 실행하면 package.json 파일에 등록된 패키지들이 모두 다시 다운로드되므로,

프로젝트 폴더를 압축할 때 node\_modules 디렉토리는 삭제하는 것이 좋다.

### 프로젝트에 필요한 패키지들 전체 설치 명령

|  |
| --- |
| npm install |

위 명령을 실행하면 package.json 파일에 등록된 패키지들이 모두 다운로드되어 프로젝트에 설치된다.