

# Hello React

## #01. 개발환경 구성하기

### 1) Node.js

#### 설치확인

OS	방법
Window	1. 명령프롬프트 실행 : <code>WinKey + R &gt; cmd</code> (엔터) 2. 명령어 수행 : <code>node --version</code>
MacOS	1. 터미널 실행 : <code>Cmd+Space &gt; terminal</code> (엔터) 2. 명령어 수행 : <code>node --version</code>

결과값이 출력되지 않을 경우 설치 필요

#### 설치하기

- <https://nodejs.org> 에서 프로그램 내려받아 설치.
- 대부분의 경우 LTS 버전 권장.
- Mac M1 버전의 경우 17 이상 버전 필요함.
- 설치 완료 후 열어두었던 명령프롬프트나 터미널을 종료하고 재시작.
- 앞서 수행한 버전확인 명령어를 통해 설치 완료 확인.

### 2) yarn

터미널(혹은 명령프롬프트)를 열고 아래의 명령어를 통해 설치 수행.

반드시 Node.js가 설치된 상태에서 진행해야 함.

```
npm install -g yarn
```

#### 설치에 실패한 경우

OS	해결방법
windows	<a href="https://classic.yarnpkg.com/latest.msi">https://classic.yarnpkg.com/latest.msi</a> 의 설치파일을 직접 내려받아 오프라인 설치 진행
Mac OS	homebrew를 사용하여 설치 수행 <code>brew install yarn</code>

### 3) Visual Studio Code

<https://code.visualstudio.com/>

Javascript 작성에 도움을 주는 Visual Studio Code Extension

이름	개발자	설명
Color Highlight	Sergii Naumov	색상 코드 값을 실제 그 색상으로 강조한다.
ESLint	Microsoft	Javascript 구문 검사 기능 터미널에서 <code>npm install -g eslint</code> 수행 필요.
Image preview	Kiss Tamás	이미지 파일을 소스코드 줄번호 옆에 미리보기 표시함
npm Intellisense	Christian Kohler	import 할 모듈의 이름이나 경로에 대한 자동완성 제공
Prettier - Code formatter	Prettier	코드의 줄바꿈, 들여쓰기등을 자동으로 정렬한다. 사용방법: <code>Ctrl+Shift+P</code> > <code>Format Document</code> 명령 선택
vscode-styled-components	Styled Components	JS 구문안에서 사용될 SCSS구문을 인식할 수 있게 함

## #02. 프로젝트 생성하기

### 1) 프로젝트 생성 및 초기화

아래의 명령으로 프로젝트를 생성한다.

프로젝트 이름은 영어 소문자만 사용 가능함.

```
yarn create react-app 프로젝트이름
```

여기서는 01-hello-react라는 이름으로 생성함

```
yarn create react-app 01-hello-react
```

### 2) yarn berry 버전으로 변경

프로젝트 폴더 안에서 명령프롬프트 실행 후 아래의 두 명령을 순차적으로 실행

```
yarn set version berry  
yarn install
```

### 3) 설정파일 수정

`.yarnrc.yml`

1라인의 `node-modules` 값을 `pnp`로 변경

`package.json`

## eslintConfig 설정부분 삭제

### 변경사항을 반영하기 위한 명령어 수행

```
yarn install
```

### 4) git 관련 파일 삭제

명령프롬프트에서 아래 명령 수행

```
rmdir /q /s .git  
del .gitignore
```

만약 git을 사용할 경우 .gitignore에 아래 내용 추가

```
.yarn/*  
!.yarn/cache  
!.yarn/patches  
!.yarn/plugins  
!.yarn/releases  
!.yarn/sdks  
!.yarn/versions
```

### 5) 프로젝트 생성 후 기초작업

아래의 작업을 진행하고 소스파일을 저장한다.

브라우저나 프로젝트를 재가동할 필요 없이 현재 브라우저 상에 결과가 즉시 반영된다.

#### 프로젝트 초기화

대상	작업 내용
src폴더	App.css, App.test.js, index.css, logo.svg, setupTests.js, reportWebVitals.js 삭제
App.js	App.css와 logo.svg에 대한 참조(import) 구문 제거
index.js	index.css와 reportWebVitals.js에 대한 참조(import) 구문 제거 맨 마지막 행에 있는 reportWebVitals() 부분 삭제 <React.StrictMode> 태그 삭제

#### App.js 파일 수정

```
import React from "react";
```

```
const App = () => {  
  return (  
    <div>  
      <h1>App</h1>  
    </div>  
  );  
};  
  
export default App;
```

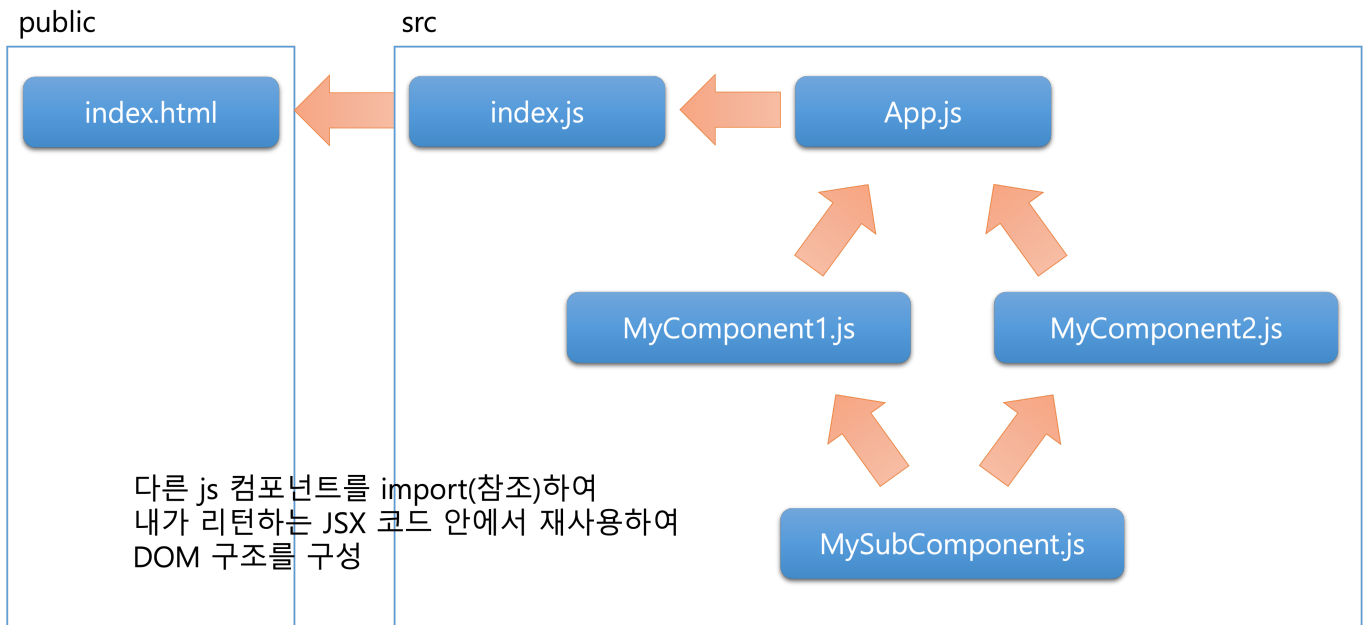
## 6) 프로젝트 실행하기

프로젝트를 VSCode로 열고, **Ctrl** + **~**를 눌러 터미널 실행.

(터미널상에서 프로젝트가 생성된 디렉토리 안에서 명령을 수행해도 동일함.)

```
yarn start
```

## #03. hello-react 소스코드 구조



## 03. 컴포넌트 종류

### 1) 컴포넌트

재사용 가능한 HTML 조각단위

### 2) 클래스 컴포넌트

- React 오리지널 컴포넌트
- class 형태로 정의하는 컴포넌트.
- 함수형 컴포넌트보다 구문이 복잡하고 길다.

- 함수형 컴포넌트보다 구문이 좀 더 명확하다.
- 클래스 안에 통칭 LifeCycle이라는 미리 정해진 함수들을 통해 화면의 갱신 시점이나 변수값의 변경 시점에 화면을 어떻게 처리할지 정의할 수 있다.

### 3) 함수형 컴포넌트 (권장)

- 클래스 컴포넌트보다 구문이 간결하지만 그만큼 더 난해할 수 있다.
- 리액트 버전 16이후에 LifeCycle을 처리할 수 있는 hook이라는 기능이 도입되면서 React에서 공식적으로 권장하는 컴포넌트 작성 방법

수업에서는 함수형 컴포넌트를 기본으로 채택하여 진행

## 04. 실습 완료 후

### 1) 실습 결과물을 다른 컴퓨터나 외부 저장장치로 옮기기

실습 결과물을 다른 컴퓨터로 옮기고자 할 때 다음의 항목들은 포함시킬 필요가 없다.

```
.yarn 폴더 내의 unplugged 폴더  
.yarn 폴더 내의 install-state.gz 파일  
node_modules 폴더
```

위의 항목들을 삭제한 후 프로젝트 폴더를 압축하여 다른 PC로 전달한다. (USB 복사 등)

### 2) 다른 컴퓨터에서 실습 결과물 확인하기

실습 결과물을 압축 해제하고 프로젝트 폴더 위치에서 아래 명령을 수행한 후 프로그램을 가동해야 한다.

```
$ yarn install
```