

VS / node / react 설치

1 VS 코드 설치 (설치 전 자리에 있는지 확인)

<https://code.visualstudio.com/docs/?dv=win64user>

Documentation for Visual Studio Code

code.visualstudio.com/docs/?dv=win64user

Visual Studio Code Docs Updates Blog API Extensions FAQ GitHub Copilot

접속하자마자 vs 코드 파일 다운로드 받아짐

Download

안된다면 버튼 클릭

Overview
SETUP
GETTING STARTED
USER GUIDE
SOURCE CONTROL
TERMINAL
GITHUB COPILOT
LANGUAGES
NODEJS / JAVASCRIPT
TYPESCRIPT
PYTHON
JAVA
C++
C#

Thanks for downloading VS Code!

Download not starting? Try this [direct download link](#).

Visual Studio Code documentation

Get familiar with Visual Studio Code and learn how to code faster with AI.

Getting started

Set up Visual Studio Code

Follow the setup guide to install and configure

Getting started

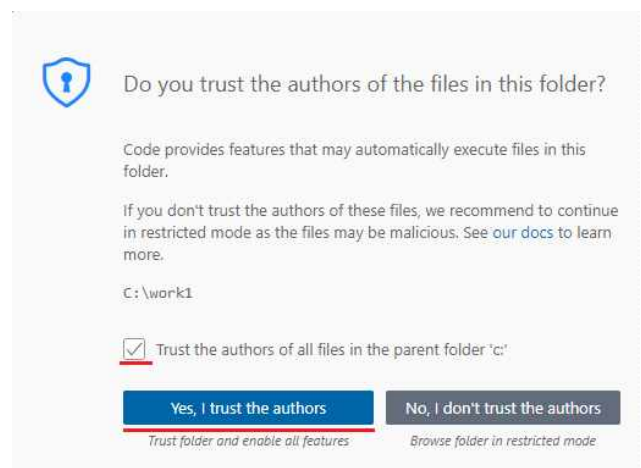
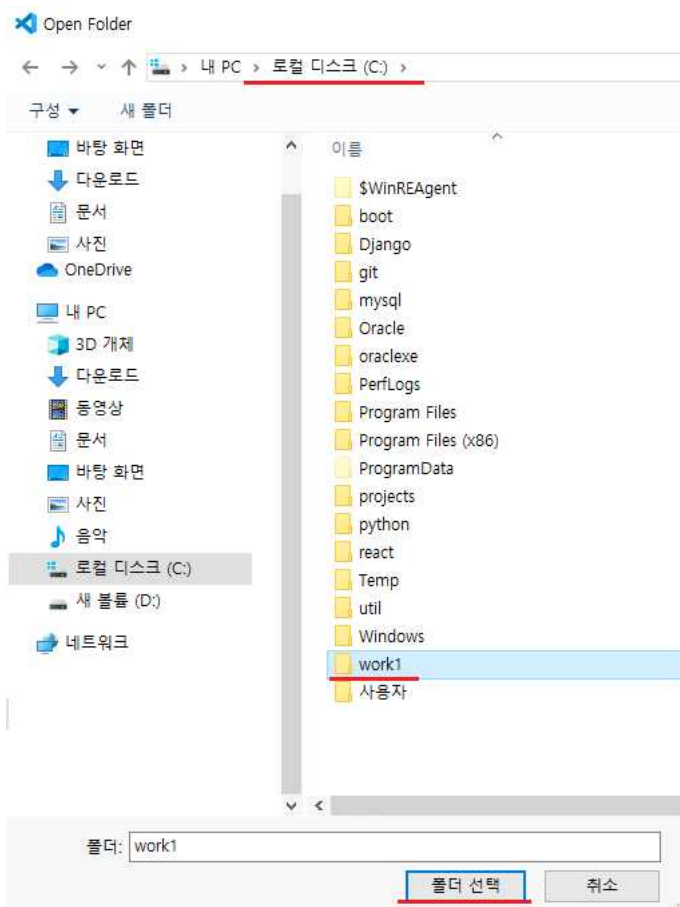
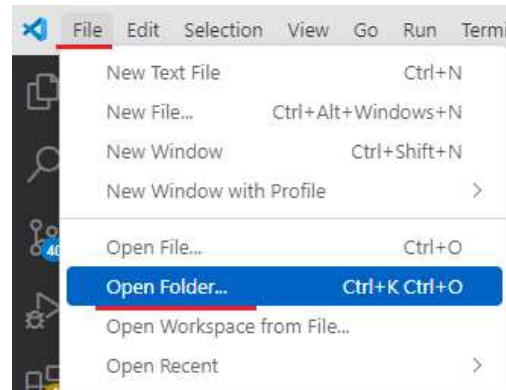
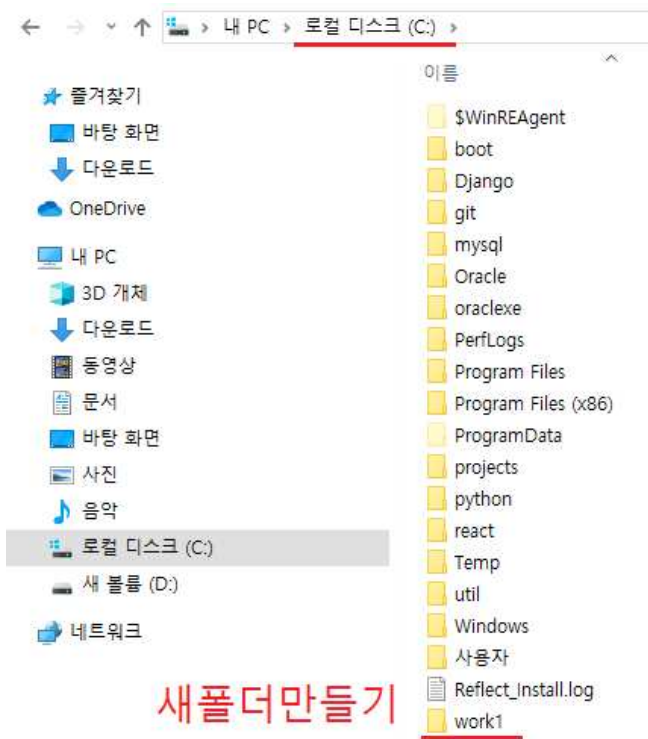
Discover the key features of VS Code with the

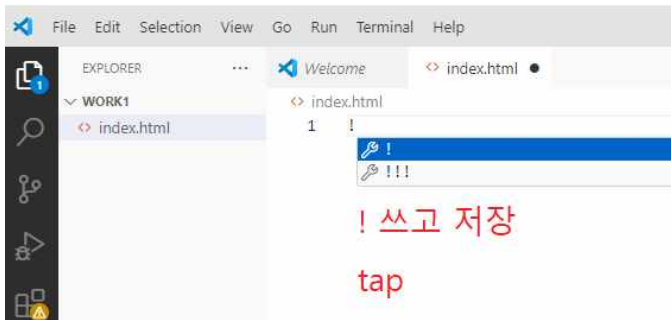
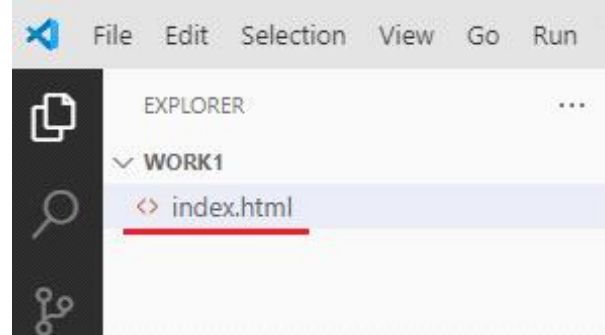
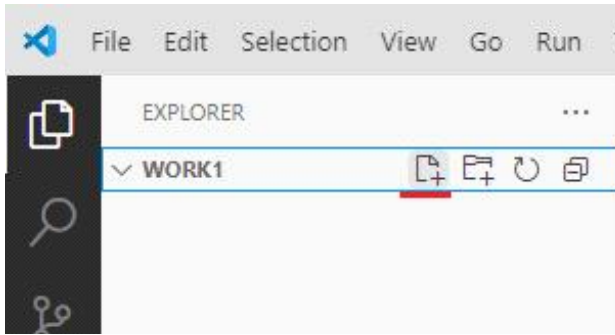
1 - 3 비주얼스튜디오 확장 프로그램 깔기

| | |
|--|--|
| 비주얼 스튜디오에서 |  |
| Open in browser | Alt + B |
| Live Server | Live Server → 실시간으로 브라우저에서 적용된 내용을 확인 (Alt + B) |
| auto rename tag | 태그의 시작/종료를 동시에 수정 |
| prettier | 코드 자동 정렬 (Ctrl + Shift + B) |
| css Peek | HTML에서 class에 적용된 CSS 속성으로 바로 이동 (<Ctrl> + <마우스 클릭>) |
| Auto import | 파일명을 입력하면 자동 import |
| Auto import -es6 | |
| Indent Rainbow | 들여쓰기를 색깔로 구분해서 가독성을 높임 |
| Reactjs code snippets | rsc로 함수 만들어줌 (자동완성) |
| ES7 + React/Redux/React-Native code snippets | rfce - 함수만들어줌(자동완성) |

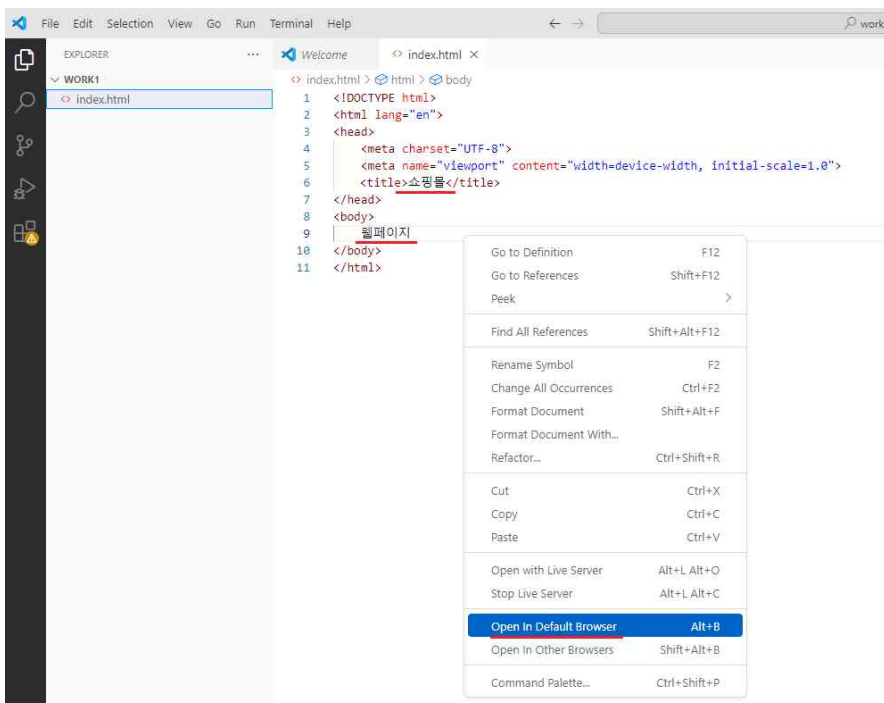
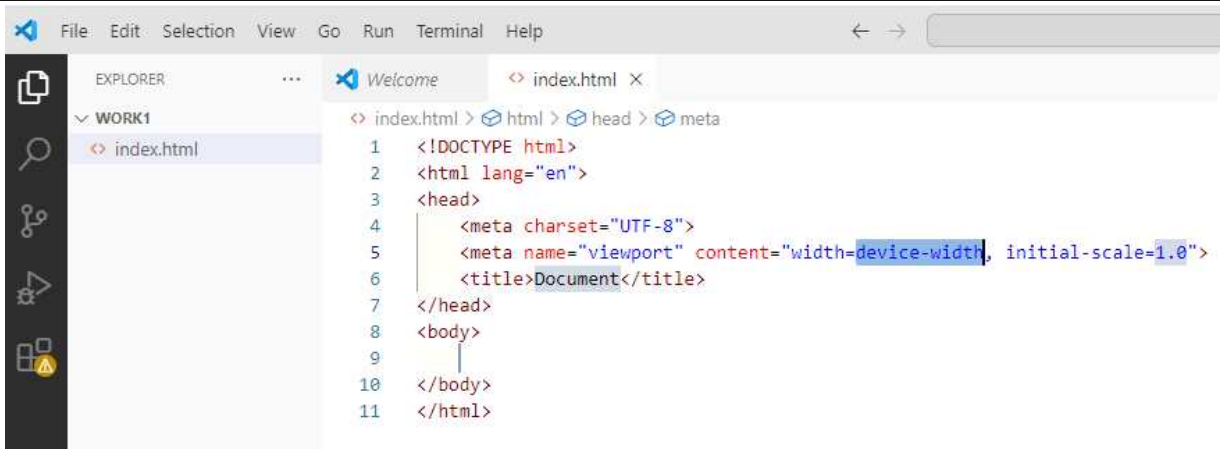
1 - 4 비주얼스튜디오 - 설정 (생략)

작업폴더만들고 index.html 파일만들기





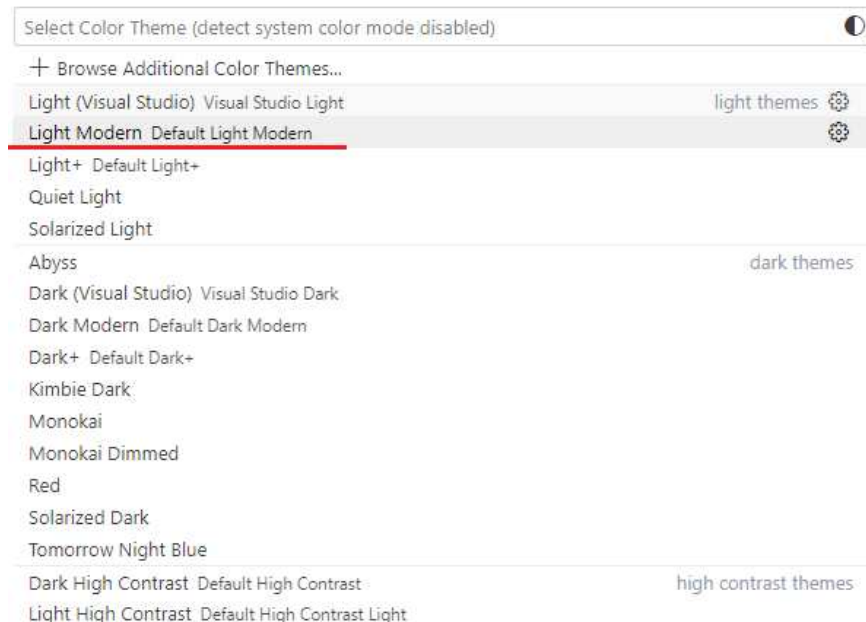
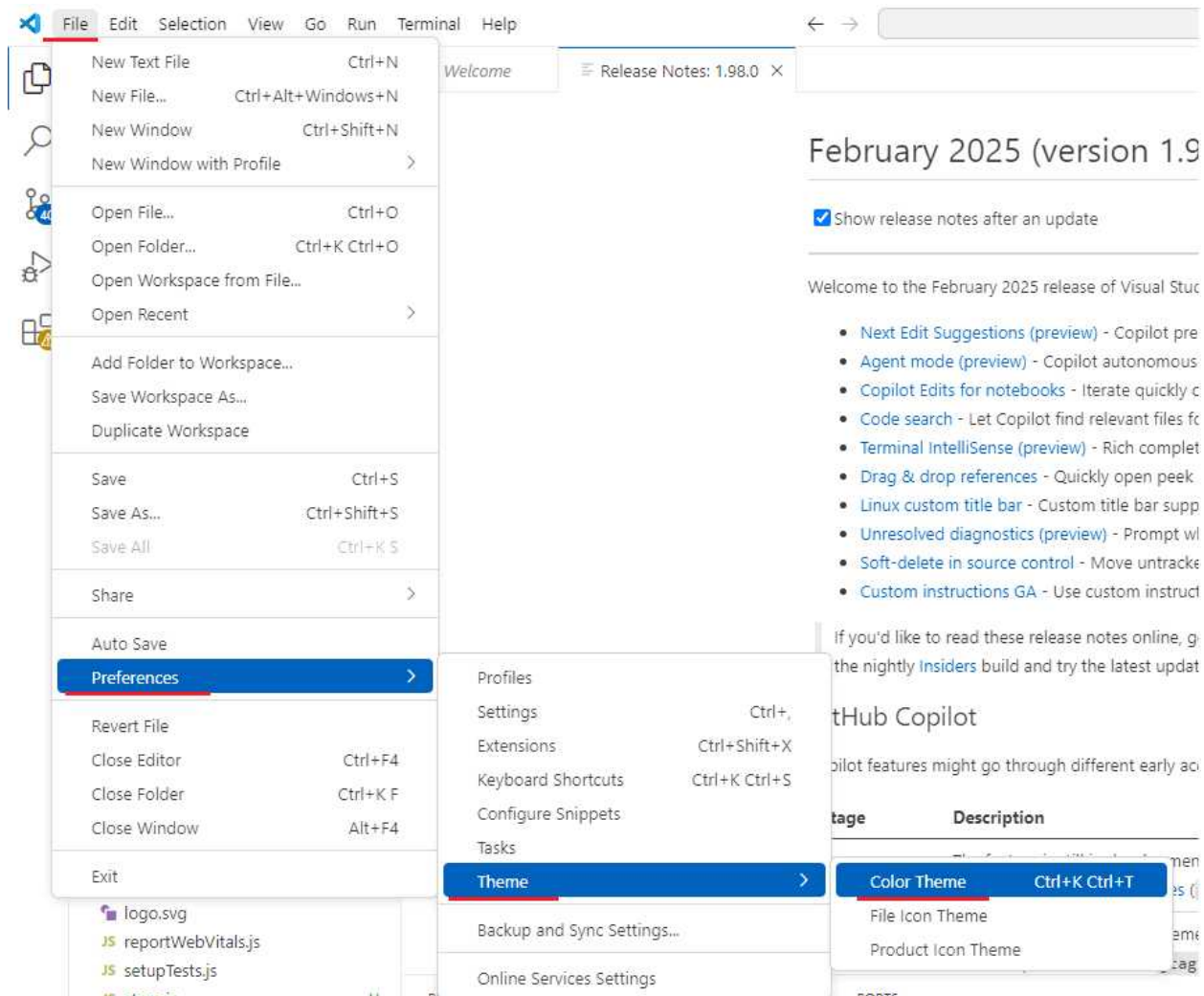
1. 파일저장하기 index.html
2. ! + Tab 키



웹페이지

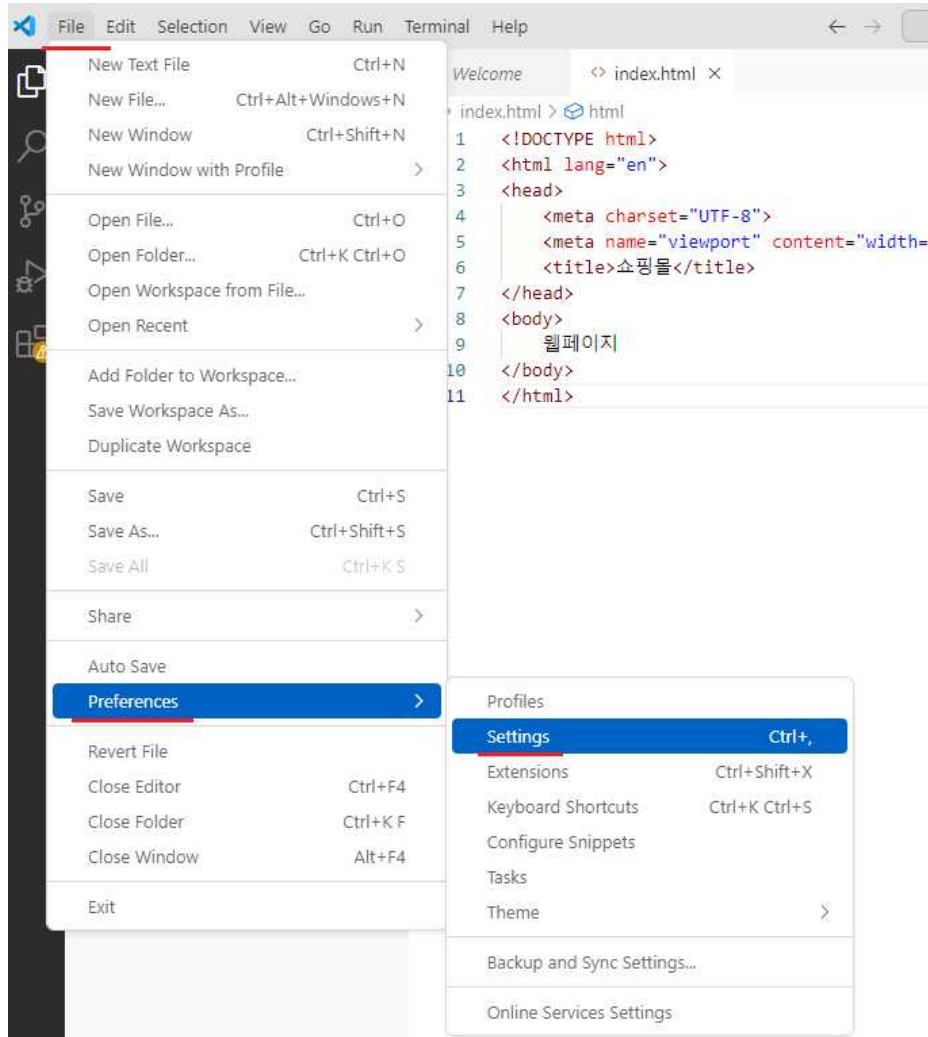
비주얼스튜디오 - 밝은 테마 사용하기

File > Preferences > Theme > Color Theme

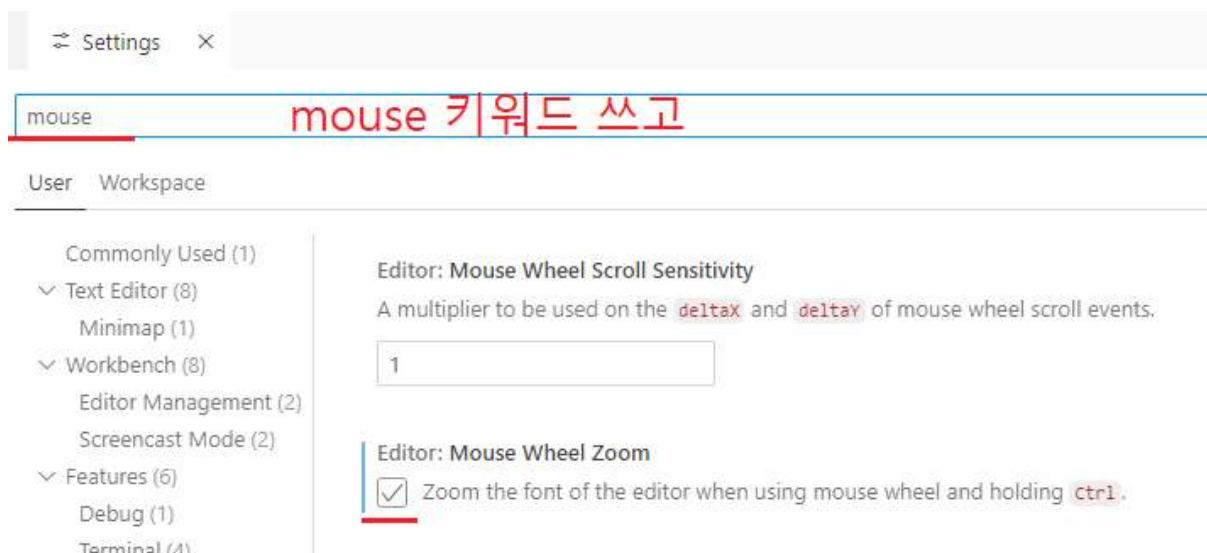


비주얼스튜디오 - 마우스휠로 폰트 키우기

File > Preferences > Settings

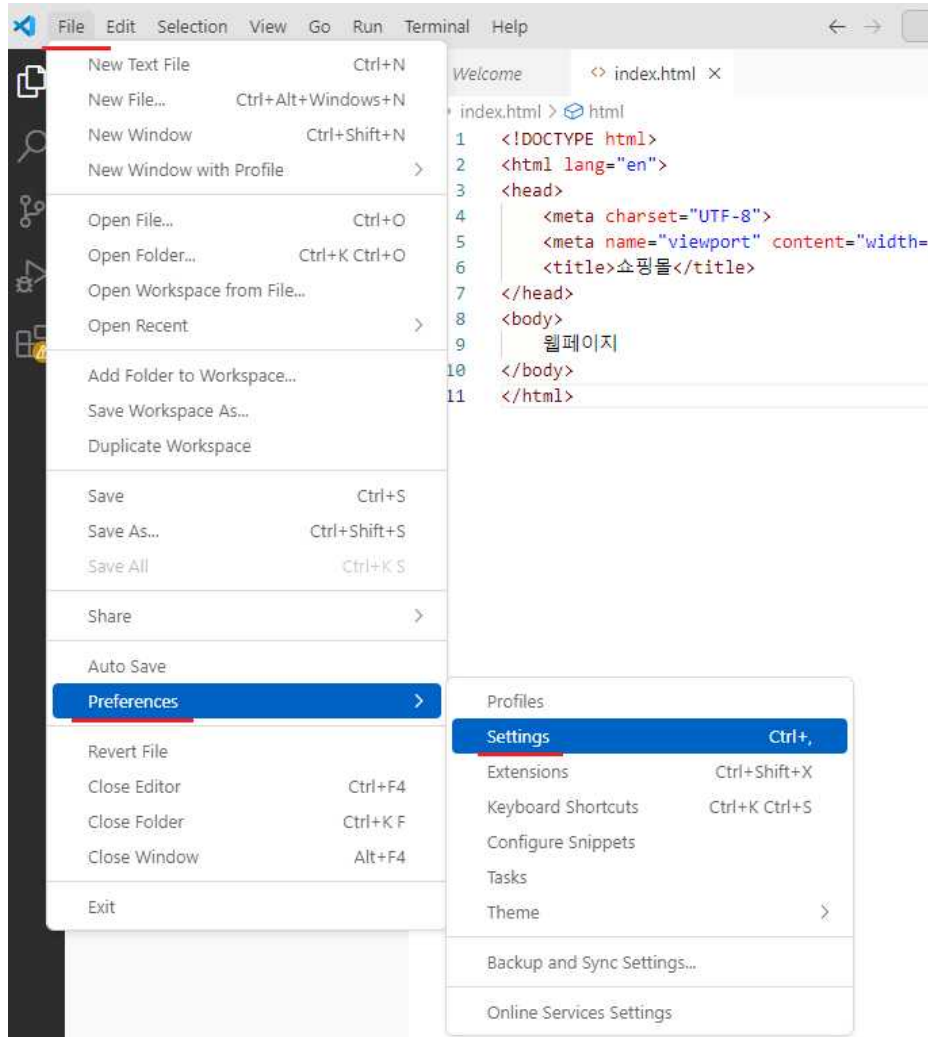


settings > mouse키워드 써주고 > Mouse Wheel Zoom 체크하기

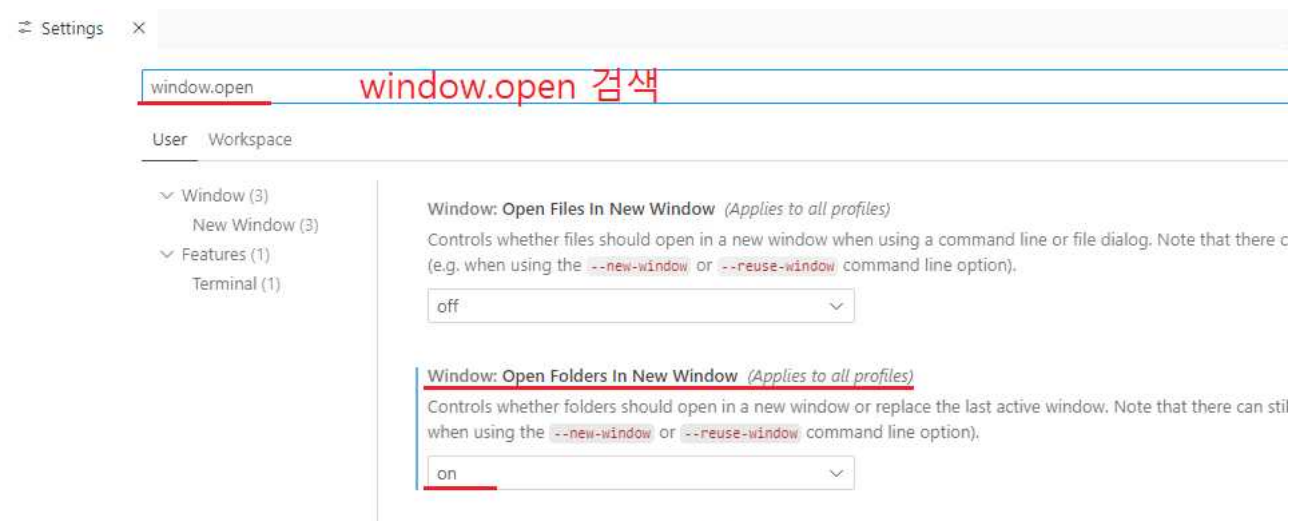


VSCode에서 파일을 항상 새 창에서 열기, 프로젝트 두 개이상 동시에 작업 설정

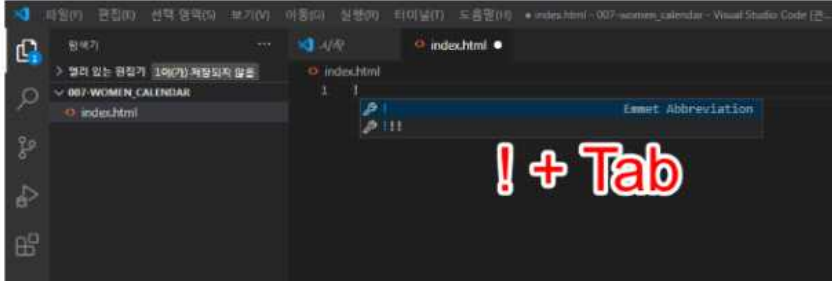
File > Preferences > Settings



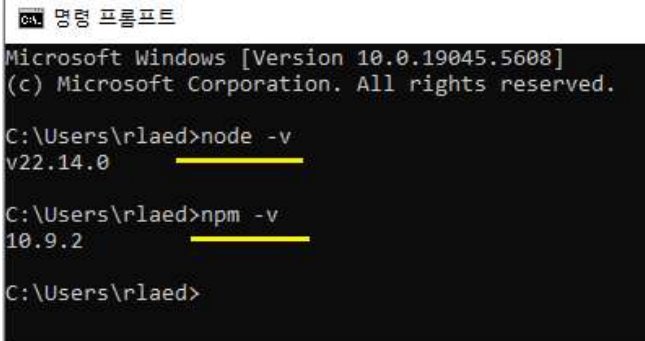

검색 : window : Open Files In New Window 에서 On으로 설정

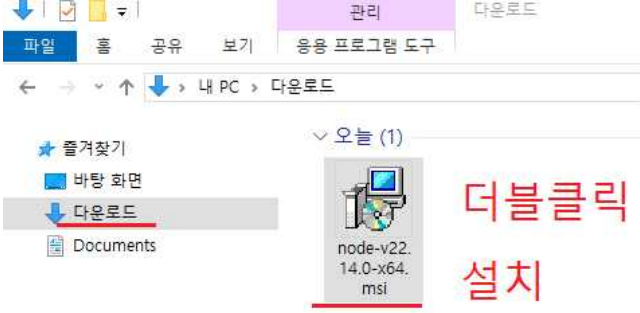
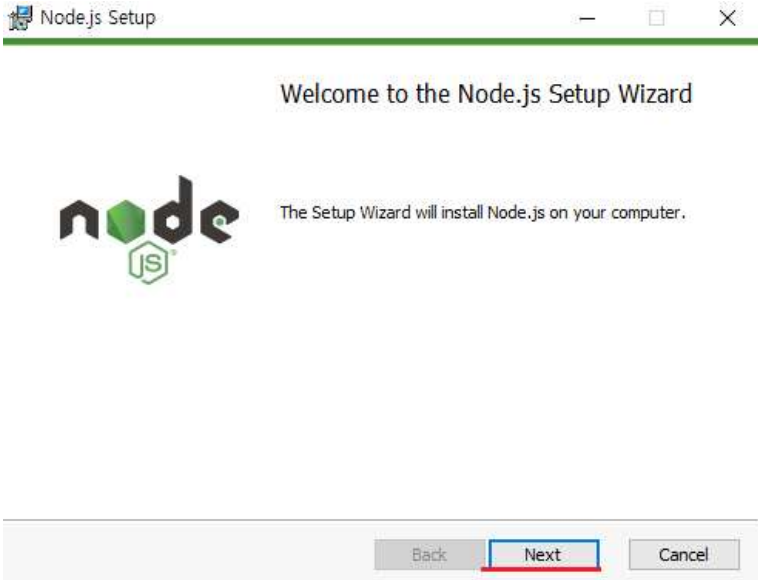


비주얼스튜디오 단축키

| | |
|-------------|---|
| html 자동완성하기 | <ol style="list-style-type: none"> 1. 파일저장하기 : index.html 2. ! + Tab 키 <p>! + Tab</p>  |
| 코드 정렬하기 | <ol style="list-style-type: none"> 1. Ctrl + A 키를 눌러 코드 전체를 모두 선택 2. Ctrl 키를 누른 상태에서 K 와 F 를 차례대로 클릭 (json 정렬도 같음) 3. prettier 코드 자동 정렬 (Ctrl + Shift + B) |
| 브라우저 열기 | Alt + B |

2 node 설치 (설치 전 cmd에서 node 있는지 확인, 없으면 설치)

| | |
|---|--|
| cmd에서 node가 있는지 확인 |  <pre> Microsoft Windows [Version 10.0.19045.5608] (c) Microsoft Corporation. All rights reserved. C:\Users\r\aed>node -v v22.14.0 C:\Users\r\aed>npm -v 10.9.2 C:\Users\r\aed> </pre> |
| https://nodejs.org/ko/ |  |

| | |
|---------------------------------------|---|
| <p>내컴퓨터에서 다운로드 폴더 확인 후 설치</p> |  |
| <p>설치</p> |  |

| | |
|-----------------|--|
| <p>동의 체크 설치</p> |  |
|-----------------|--|

설치 끝나고
cmd에서 확인

node -v
npm -v

명령 프롬프트

```
Microsoft Windows [Version 10.0.19045.5608]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rlaed>node -v
v22.14.0

C:\Users\rlaed>npm -v
10.9.2

C:\Users\rlaed>
```

3 react 프로젝트 만들기

탐색기로
작업 폴더 만들기

> 내 PC > 로컬 디스크 (C:) > react_0621

| | |
|---|--|
| <div>cmd 열기</div> <div>cd C:\react_0621</div> <div>npx create-react-app blog1</div> | <div>관리자: 명령 프롬프트</div> <div> Microsoft Windows [Version 10.0.19045.5965] (c) Microsoft Corporation. All rights reserved. C:\Users\Administrator.User -2025\RCIA>cd C:\react_0621 C:\react_0621>npx create-react-app blog1 Creating a new React app in C:\react_0621\blog1. Installing packages. This might take a couple of minutes. Installing react, react-dom, and react-scripts with cra-template... added 1324 packages in 38s 269 packages are looking for funding run `npm fund` for details Initialized a git repository. Installing template dependencies using npm... added 18 packages, and changed 1 package in 4s 269 packages are looking for funding run `npm fund` for details Removing template package using npm... removed 1 package, and audited 1342 packages in 3s 269 packages are looking for funding run `npm fund` for details 9 vulnerabilities (3 moderate, 6 high) To address all issues (including breaking changes), run: npm audit fix --force Run `npm audit` for details. Created git commit. 확인 Success! Created blog1 at C:\react_0621\blog1 Inside that directory, you can run several commands: npm start Starts the development server. npm run build Bundles the app into static files for production. </div> |
| <div>위의처럼 안뜨면</div> <div>cmd에서 코드 복사 붙여넣기</div> | <div>리액트 프로젝트가 생성되지 않으면 참고 url: https://blog.naver.com/ysboo2/223757816691</div> <div>cmd > 프로젝트 경로 > 아래코드 붙여넣고 실행 (아래코드 한줄, 한번에 복사)</div> <div> npm install react@18 react-dom@18 @testing-library/jest-dom@5.17.0 @testing-library/react@13.4.0 @testing-library/user-event@13.5.0 web-vitals </div> <div>npx create-react-app blog1</div> |

프로젝트 만들어졌는지
확인

> 내 PC > 로컬 디스크 (C:) > react_0621

react

이름

blog1

폴더 확인

> 내 PC > 로컬 디스크 (C:) > react_0621 > blog1 >

이름

확인

.git
node_modules
public
src
.gitignore
package.json
package-lock.json
README.md

> 내 PC > 로컬 디스크 (C:) > react_0621 > blog1 >

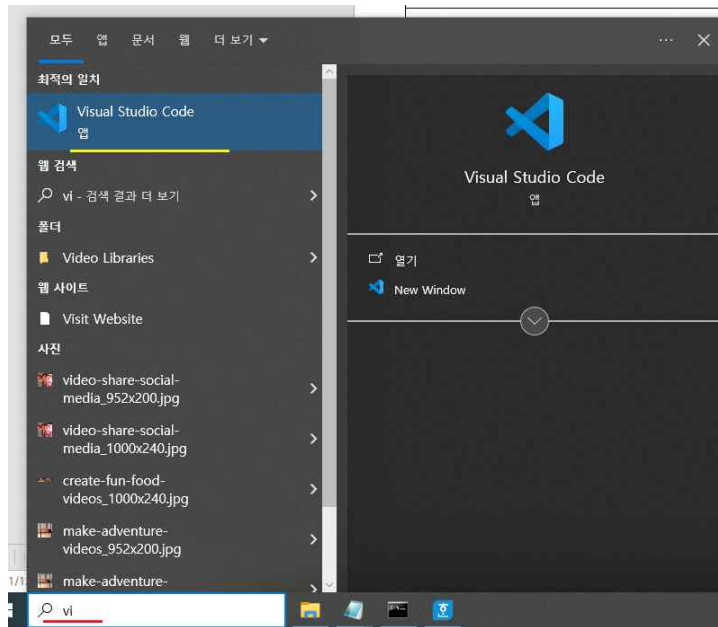
이름

node_modules 라이브러리 저장소 (자동 생성)
public 정적 이미지 넣는 곳
src 코딩하는 곳
.gitignore
package.json
package-lock.json 프로젝트의 설계도 (라이브러리 기록)
README.md - 버전, 프로젝트 이름

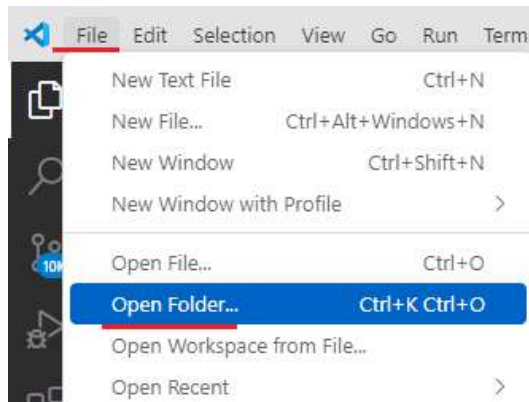
탐색기에서 프로젝트가 만들어졌는지 확인하고 나서는 cmd창의 닫는다.

4 react 프로젝트 실행되는지 확인

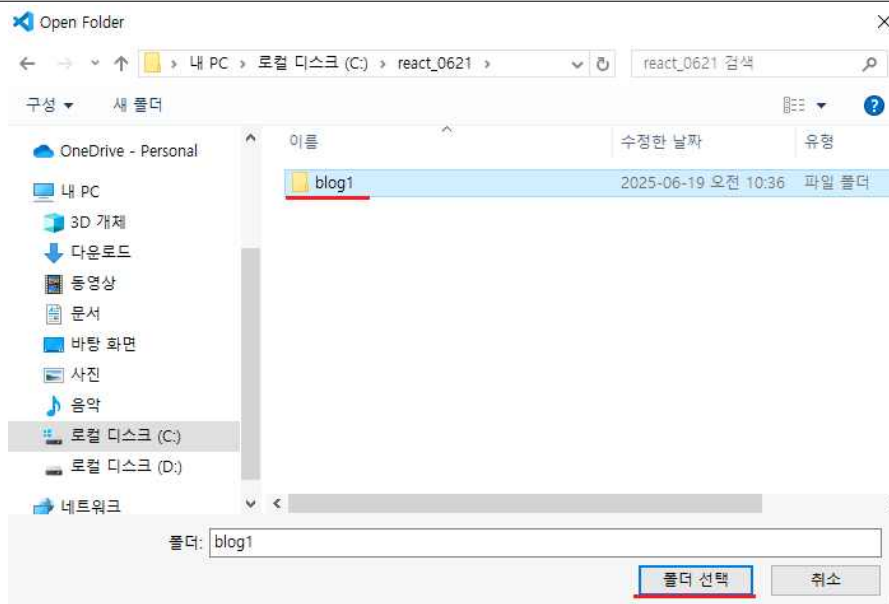
비주얼 스튜디오
열기

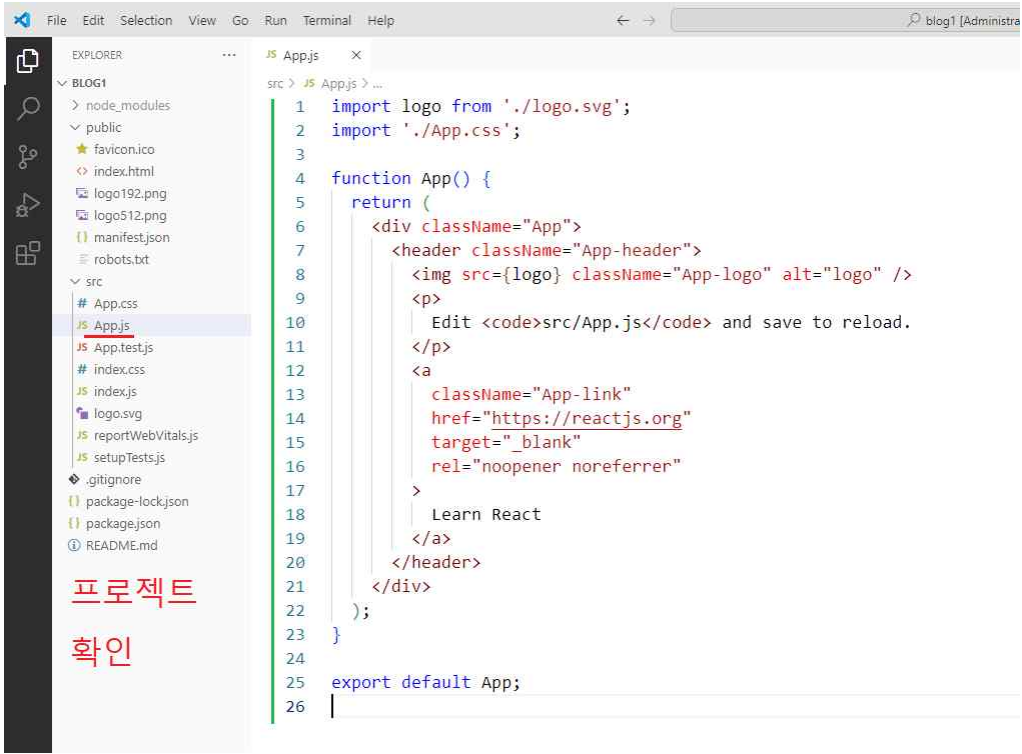
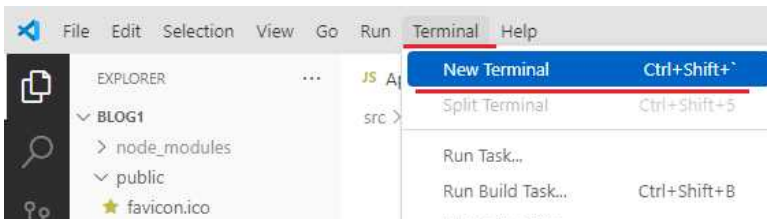

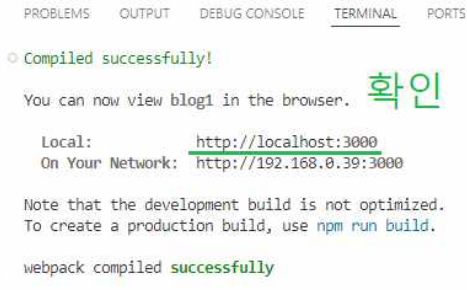
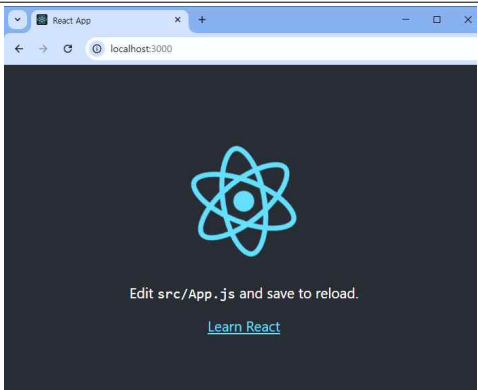


vi 코드에서
Open Folder



blog1 폴더 열기



| | |
|---|--|
| <p>App.js 선택</p> |  <p>프로젝트 확인</p> |
| <p>터미널 열기 단축키 Ctrl + Shift + ` (백틱)</p> |  |
| <p>터미널에서 npm start</p> |  <p>프론트 서버 실행</p> |
| <p>React 앱은 http://localhost:3000 포트에서 실행</p> |  <p>확인</p> |
| <p>브라우저에서 확인 http://localhost:3000/ 으로 자동으로 오픈</p> |  <p>옆의 이미지처럼 나와야 성공</p> |

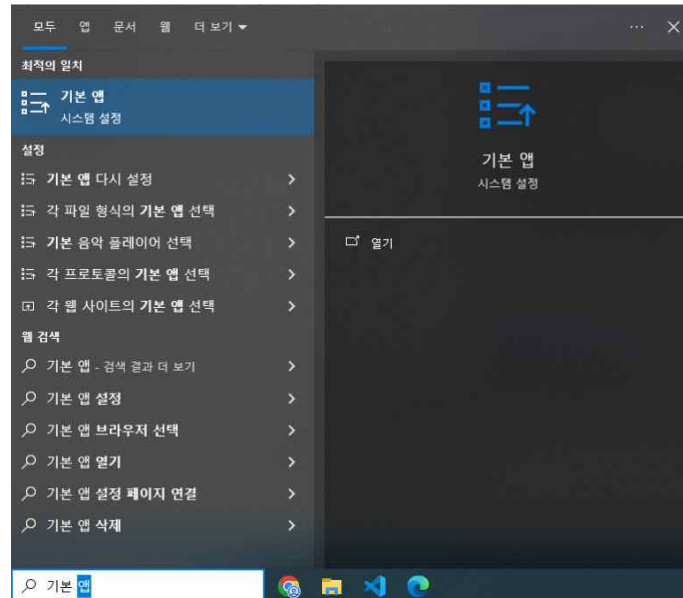
리엑트 프로젝트가 크롬 브라우저로 안 뜬다면

참고 url

react 프로젝트 기본 브라우저가 크롬이 아닐경우 (크롬 브라우저)
<https://blog.naver.com/ysboo2/223841170588>

기본 브라우저가 크롬이 아닐경우

윈도우 창에서
검색 기본앱

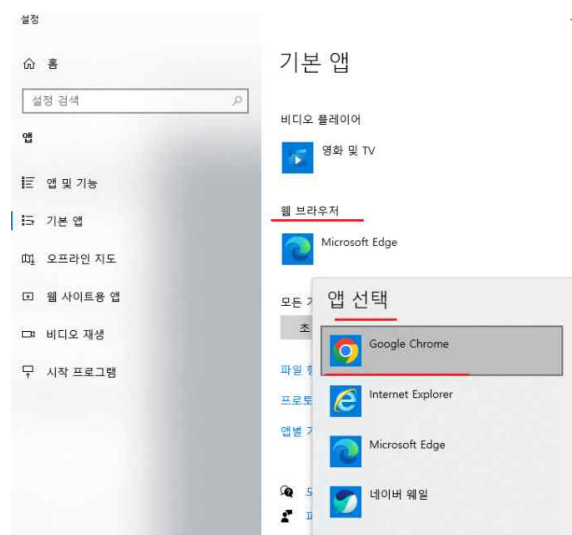


기본앱 아래쪽에

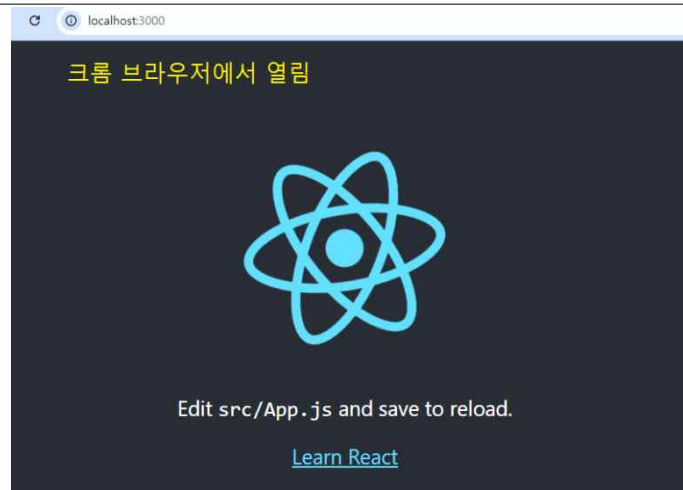
웹브라우저

> 앱선택

> 구글 크롬 선택

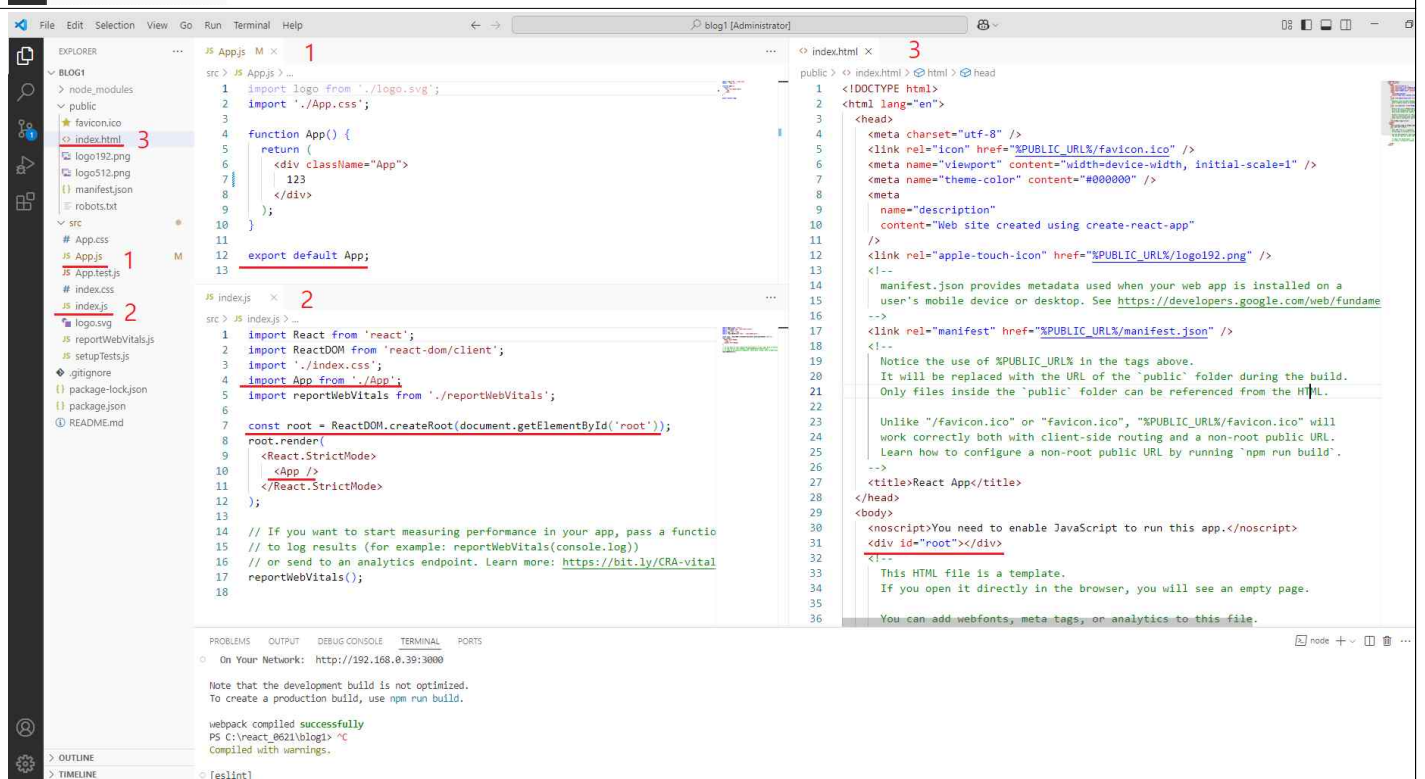
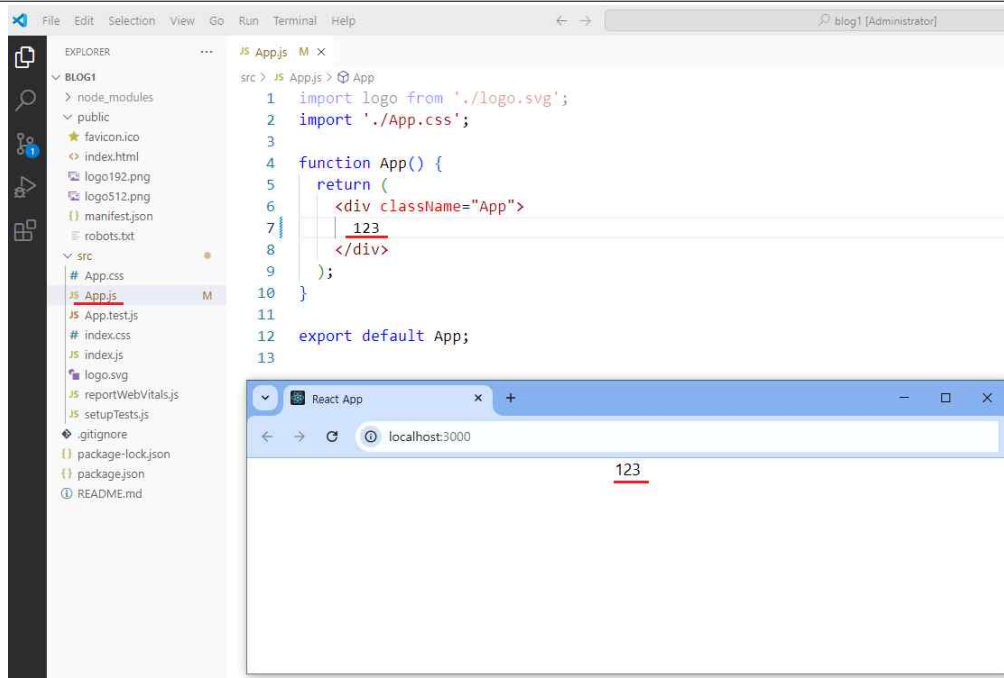


크롬으로 열리는지
확인



프론트 서버 종료
터미널에서 Ctrl + C

App.js > 작업 header를 다 지우고 > 123만 작성 > 터미널에서 npm start > 브라우저에서 확인



| | | |
|---|---|---|
| React 구조 | React는 public/index.html이라는 HTML 껍데기 하나만 있고, 진짜 화면 내용은 src/App.js처럼 자바스크립트(컴포넌트)로 채워지는 구조 | |
| 동작순서 | 1 index.html | <div id="root"></div> 라는 빈 공간이 있음 (HTML 껍데기) |
| | 2 index.js | root라는 그 빈 공간을 찾아서, App 컴포넌트를 거기다가 넣어줌 const root = ReactDOM.createRoot(document.getElementById('root')); root.render(<App />); |
| | 3 App.js | 실제로 보이는 화면 내용을 여기서 만들고 구성함 |
| index.html은 틀이고, 실제 내용은 App.js부터 컴포넌트로 구성 | | |

그림으로 이해 1

index.html contains an empty <div>
with the id "root"

1. index.html

```
<div id="root">/div>
```



index.js finds the root and renders the
App component in it

2. index.js

```
const root = ReactDOM.createRoot  
(document.getElementById('root'));  
root.render(<App />);
```



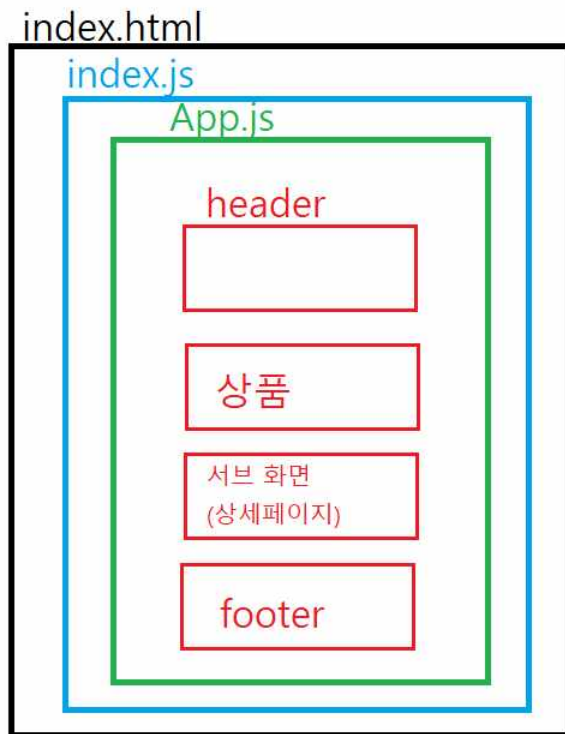
App.js component is created in root

3. App.js

App

그림으로 이해 2

기존처럼 html을 여러개 만드는게 아니라 하나의 html안에 한페이지에 컴포넌트로 채움(SPA)



App.js 에서
모든 컴포넌트를 다 넣고 SPA

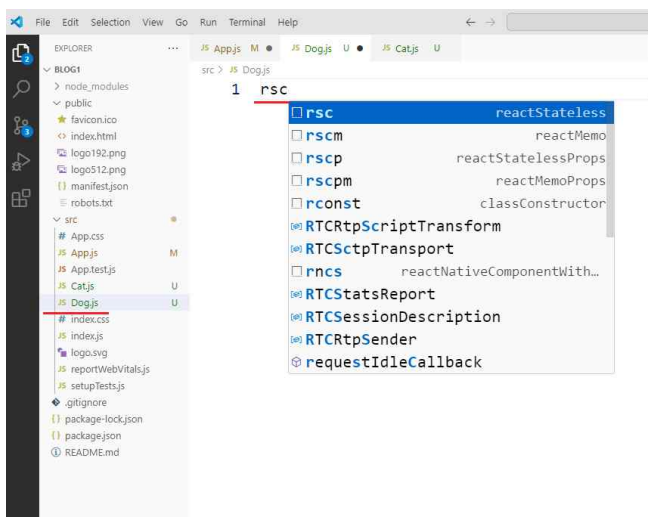
조건이 되면
컴포넌트가 보였다 안보였다 제어한다.

App.js는 모든 컴포넌트의 중심이고,
조건(if, useState, 라우터)에 따라 화면을 바꿔가며 보여주는 SPA 구조

| | |
|------------|--|
| SPA의 단점 | 문제는 서버에서 렌더링하면 검색이 잘되는데 리액트로 만든 SPA는 검색에 노출이 잘 안되서 Next.js를 많이 사용한다. Next.js는 TypeScript가 기본이다 |
| | 일반적인 React(SPA)는 모든 화면을 브라우저에서(Javascript로) 그리는 방식이다. 그래서 검색 엔진(구글, 네이버)이 페이지 내용을 제대로 읽지 못할 수 있다. 반면에 Next.js는 페이지를 서버에서 미리 만들어서(SSR) 브라우저에 보내준다. 그래서 검색 엔진에도 잘 노출되고, SEO(검색 최적화)에 유리하다. 또한, Next.js는 기본 설정에 TypeScript도 쉽게 쓸 수 있게 되어 있어서, 대규모 프로젝트나 팀 개발에 많이 사용되고 있어요. |
| SPA | Single Page Application 하나의 페이지에서 모든 기능이 동작하는 앱 처음에 HTML 하나만 로딩하고, 이후에는 페이지를 다시 새로고침하지 않고 필요한 부분만 JavaScript로 바꿔서 보여주는 방식 한 페이지 안에서 필요한 화면만 바꿔 보여주는 웹앱 |
| SSR | Server Side Rendering (서버 사이드 렌더링) 사용자가 요청하면 서버가 HTML을 만들어 보내줌 검색엔진이 쉽게 읽을 수 있어 SEO에 유리함 |
| SEO | Search Engine Optimization 검색 엔진에 잘 노출되게 만드는 기술, 검색엔진 최적화 Search → 검색 Engine → 검색엔진 (구글, 네이버 등) Optimization → 잘 보이도록 최적화 |

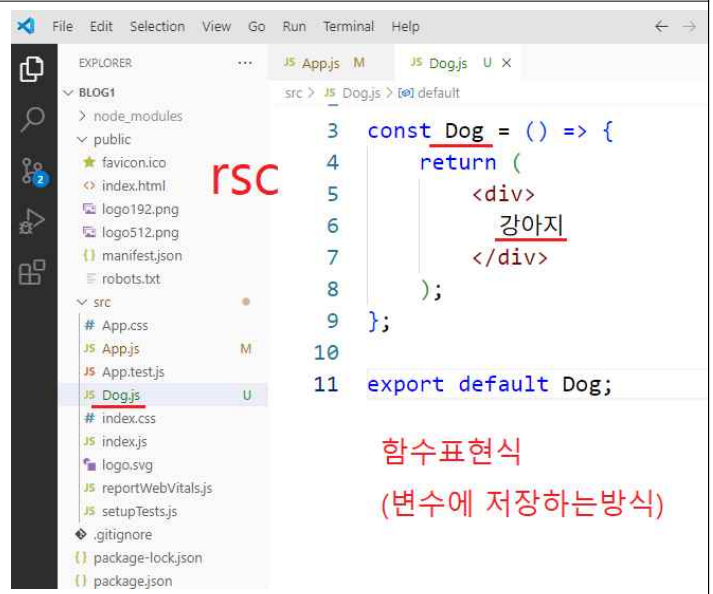
리엑트에서 함수 만드는 방법 - **대문자로 컴포넌트**를 만들고 src폴더 아래 > Dog.js 만들기

함수표현식 1 (변수에 답아쓰는 방법)



1 **rsc**

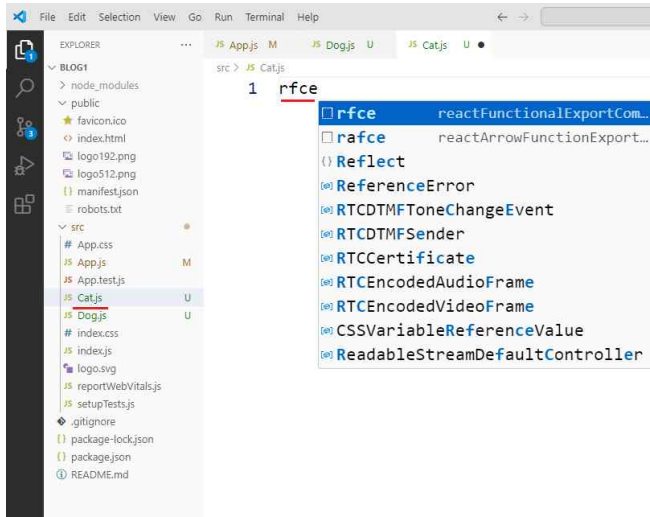
- reactStateless
- reactMemo
- reactStatelessProps
- reactMemoProps
- classConstructor
- RTCTrtpScriptTransform
- RTCTrtpTransport
- reactNativeComponentWith...
- RTCTrtpStatsReport
- RTCTrtpSessionDescription
- RTCTrtpSender
- requestIdleCallback



3 **const** Dog = () => {
4 **return** (
5 <div>
6 강아지
7 </div>
8);
9 };
10
11 **export default** Dog;

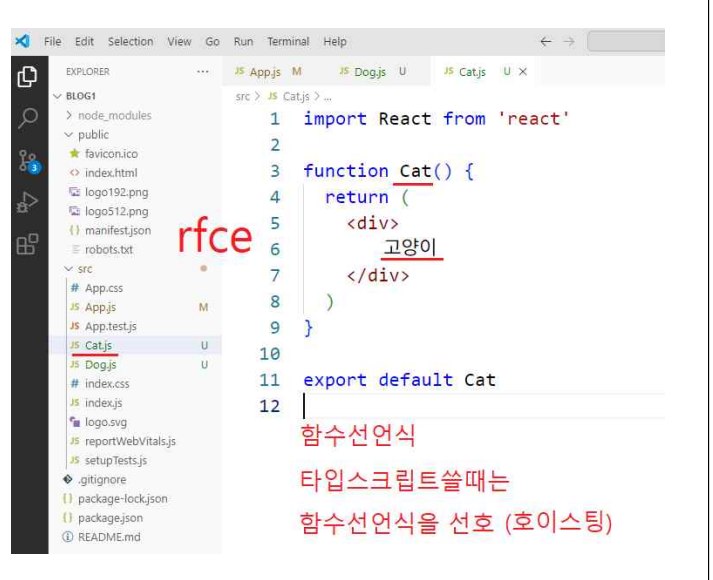
함수표현식
(변수에 저장하는방식)

함수선언식2 - 타입스크립트에서는 함수선언식을 선호 (호이스팅)



1 **rfce**

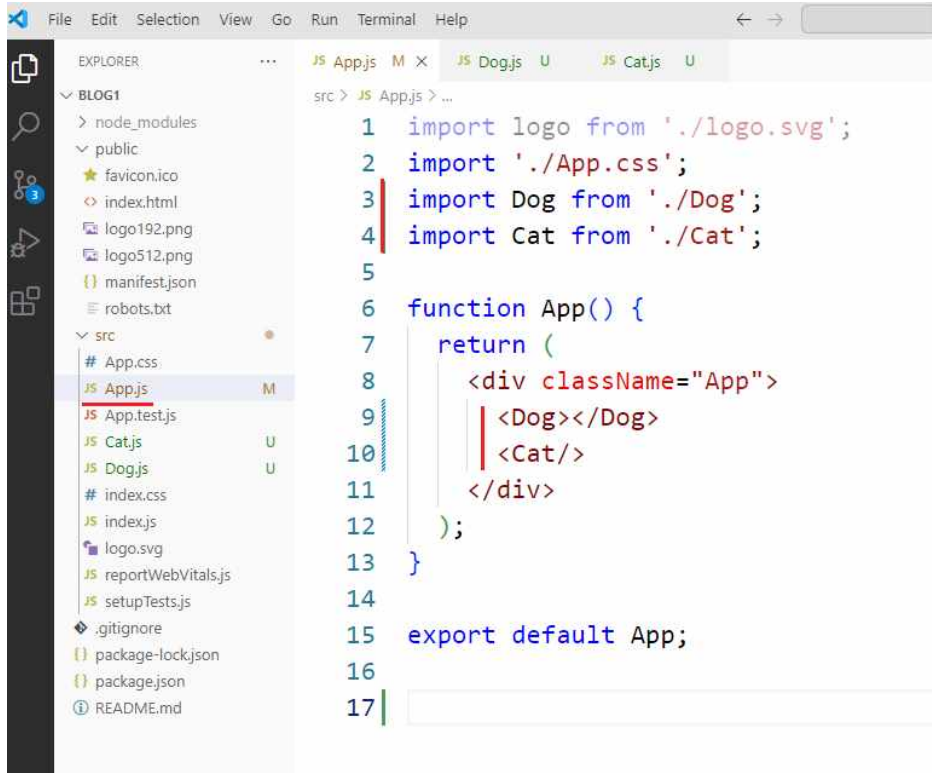
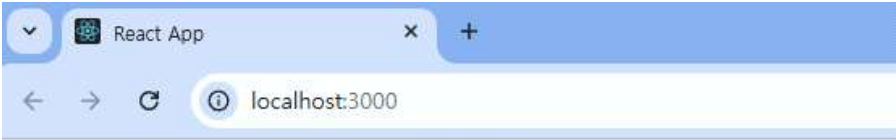
- reactFunctionalExportCom...
- reactArrowFunctionExport...
- Reflect
- ReferenceError
- RTCDTMFToneChangeEvent
- RTCDTMFSender
- RTCCertificate
- RTCEncodedAudioFrame
- RTCEncodedVideoFrame
- CSSVariableReferenceValue
- ReadableStreamDefaultController



1 **import** React **from** 'react'
2
3 **function** Cat() {
4 **return** (
5 <div>
6 고양이
7 </div>
8)
9 }
10
11 **export default** Cat
12

함수선언식
타입스크립트쓸때는
함수선언식을 선호 (호이스팅)

App.js 에 컴포넌트 가져오기

| | |
|---------------------|--|
| App.js |  <pre> 1 import logo from './logo.svg'; 2 import './App.css'; 3 import Dog from './Dog'; 4 import Cat from './Cat'; 5 6 function App() { 7 return (8 <div className="App"> 9 <Dog></Dog> 10 <Cat/> 11 </div> 12); 13 } 14 15 export default App; 16 17 </pre> |
| App.js | <pre> import './App.css'; import Dog from './Dog'; import Cat from './Cat'; function App() { return (<div className="App"> <Dog></Dog> <Cat/> </div>); } export default App; </pre> |
| 터미널 | npm start |
| 브라우저 확인 |  <p>강아지 고양이</p> |
| 터미널 프론트 서버 종료 | Ctrl + C |



| | |
|---------|--|
| header | 헤더를 의미, 머리말 |
| nav | 네비게이션을 의미 / 메뉴,로그인 서브메뉴 |
| section | 여러 중심 내용을 감싸는 공간을 의미, 웹컨텐츠들을 그룹으로 묶어주는 역할을 담당 |
| article | 글자가 많이 들어가는 부분을 의미, 본문내용 웹페이지 상에서의 실제 내용을 의미 |
| figure | 이미지사진 등 |
| aisde | 사이드에 위치하는 공간을 의미 (퀵메뉴, 서브메뉴) |
| footer | 푸터를 의미, 웹사이트의 저작권 정보나 저작권표기 |



카테고리

쿠팡

오늘 밤 12시까지 주문해도
로켓배송은 내일 도착!

로켓배송

혁신 탈모증상완화

로그인 회원가입 고객센터 판매자 가입

전체

찾고 싶은 상품을 검색해보세요!

🔍

마이쿠팡

장바구니

우편플래미

로켓배송

로켓프레시

다시 구매

biz 쿠팡비즈

로켓식구

골드박스

이달의신상

판매자특가



Galaxy Z Fold7 | Z Flip7

사전 구매

쿠팡캐시+용량 업그레이드

오늘이 마지막!

| | |
|---------------------------|---|
| Galaxy Z Fold7 Z Flip7 |  |
| 여름 한정 스포츠 액세서템 |  |
| iPad Air |  |
| 쿨링 샌리데, 아기상어 기저귀 |  |
| 무본 예매 |  |
| 새로운 워진 바리스타를스 |  |

Main

오늘의 판매자 특가




특가전행중

맛있는 찹옥수수 미백 냉동 옥수수 특품 10개,...

외우팔인가 27%

13,090원
★★★★★ (849)



특가전행중

맛있는 찹옥수수 냉동 옥수수 10개, 옥수수증

외우팔인가 17%

13,140원
★★★★★ (5,714)



특가전행중

아심찬 찹쌀냉면 10인분 + 냉면옥수수 x 10봉, 1세트,...

외우팔인가 25%

11,900원
★★★★★ (2,867)



특가전행중

(정순푸드) 당일제조 천라도 식 알타리 총각김치, 1개,...

외우팔인가 48%

17,850원
★★★★★ (1,165)



특가전행중

프리미엄갈치 총2kg 30토막, 약 200g(3토막), 5개

외우팔인가 28%

14,310원
★★★★★ (1,1814)


전세계 핫딜 로켓직구 글로벌특가



저가 49% 할인 중

Tiger Pavilion 찰아시 뎀트형 모기장


6,990원 로켓직구
★★★★★ (37)



저가 83% 할인 중

류카쿠산 약 찰 먹이네 찰리형 오몰라토 포도맛 맥, 5개, 200g


3,950원 로켓직구
★★★★★ (1)



저가 57% 할인 중

자수 장아치 고양이 이미지 커여 운 바렌스커튼 2층


13,020원 로켓직구
★★★★★ (68)



저가 11% 할인 중

GNCMEGAMEN 40+ 베타락 프로그립, 30회분, 2개

84,020원 로켓직구
★★★★★ (4)

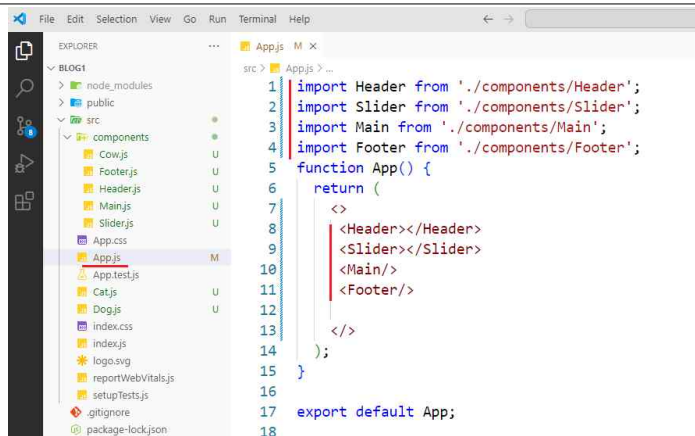


저가 49% 할인 중

clever 반려견을 낚는 울타리, 화이트, 1개

17,990원 로켓직구
★★★★★ (12)

컴포넌트를 이용한 웹 페이지 – App.js



| 폴더 | 파일 |
|------------|-----------|
| components | Header.js |
| | Slider.js |
| | Main.js |
| | Footer.js |

| | | | |
|-----------------------------|---|------------------------------|---|
| Header.js 단축키 rsc | <pre>import React from 'react'; const Header = () => { return (<div> 헤더 </div>); }; export default Header;</pre> | Slider.js 단축키 rsc | <pre>import React from 'react'; const Slider = () => { return (<div> 슬라이더 </div>); }; export default Slider;</pre> |
| Main.js 단축키 rfce | <pre>import React from 'react' function Main() { return (<div> 메인 </div>) } export default Main</pre> | Footer.js 단축키 rfce | <pre>import React from 'react' function Footer() { return (<div> 풋터 </div>) } export default Footer</pre> |
| App.js | <pre>import Header from './components/Header'; import Slider from './components/Slider'; import Main from './components/Main'; import Footer from './components/Footer'; function App() { return (<> <Header></Header> <Slider></Slider> <Main/> <Footer/> </>); } export default App;</pre> | | |

Header.js 컴포넌트에 Style 입히기 – 1 인라인 스타일 style={} 머스테치(Mustache)

| | |
|--|---|
| <div> <div>EXPLORER</div> <div> <div>BLOG1</div> <div> <div>node_modules</div> <div>public</div> <div>src</div> <div>components</div> <div>Cow.js</div> <div>Footer.js</div> <div>Header.js</div> <div>Main.js</div> <div>Slider.js</div> <div>App.css</div> <div>App.js</div> <div>App.test.js</div> <div>Cat.js</div> <div>Dog.js</div> </div> </div> </div> | <div> <div>Header.js U X</div> <div>src > components > Header.js > ...</div> <div> <div>1 import React from 'react';</div> <div>2</div> <div>3 const Header = () => {</div> <div>4 return (</div> <div>5 <div style={{width:"100%", color:"white", height:"50px", backgroundColor:"green"}}></div> <div>6 헤더</div> <div>7 </div></div> <div>8);</div> <div>9 };</div> <div>10</div> <div>11 export default Header;</div> <div>12</div> <div>13</div> </div> <div>코드작성</div> <div>카멜기법</div> </div> |
| <div>components > Header.js</div> | <div> <div>import React from 'react';</div> <div>const Header = () => {</div> <div> return (</div> <div> <div style={{width:"100%", color:"white", height:"50px", backgroundColor:"green"}}></div> <div> 헤더</div> <div> </div></div> <div>);</div> <div>};</div> <div>export default Header;</div> </div> |
| <div>터미널 확인 npm start</div> | <div> <div>localhost:3000</div> <div>헤더</div> <div>슬라이더</div> <div>메인</div> <div>풋터</div> </div> |
| <div>style={{ }} 머스테치(Mustache)</div> | <div> <div>JavaScript 객체를 표현하는 JSX 문법</div> <div>바깥 { ... } → JSX 안에서 JavaScript 표현식을 넣기 위한 괄호</div> <div>안쪽 { ... } → JavaScript의 객체 리터럴 (스타일 정의용)</div> <div>JSX 안에 객체를 넣는다.</div> </div> |

slider.js 컴포넌트에 Style 입히기 – 2 props로 스타일 적용하기

VS Code Explorer shows the project structure with components like Cow.js, Footer.js, Header.js, Main.js, Slider.js, App.css, App.js, App.test.js, Cat.js, Dog.js, index.css, index.js, logo.svg, reportWebVitals.js, and setupTests.js.

```

1 import Header from './components/Header';
2 import Slider from './components/Slider';
3 import Main from './components/Main';
4 import Footer from './components/Footer';
5 function App() {
6   return (
7     <>
8     <Header/>
9     <Slider style={{width:"100%", color:"white", height:"200px", backgroundColor:"gold"}}/></Slider>
10    <Main/>
11    <Footer/>
12  </>
13  );
14 }
15
16 export default App;
  
```

코드추가

components >
App.js

```

import Header from './components/Header';
import Slider from './components/Slider';
import Main from './components/Main';
import Footer from './components/Footer';
function App() {
  return (
    <>
    <Header/>
    <Slider style={{width:"100%", color:"white", height:"200px", backgroundColor:"gold"}}/> </Slider>
    <Main/>
    <Footer/>
    </>
  );
}
export default App;
  
```

VS Code Explorer shows the project structure with components like Cow.js, Footer.js, Header.js, Main.js, Slider.js, App.css, App.js, App.test.js, Cat.js, index.css, index.js, logo.svg, reportWebVitals.js, and setupTests.js.

```

1 import React from 'react';
2
3 const Slider = (props) => {
4   return (
5     <div style={props.style}>
6       슬라이더
7     </div>
8   );
9 };
10
11 export default Slider;
12
  
```

코드추가

Slider.js

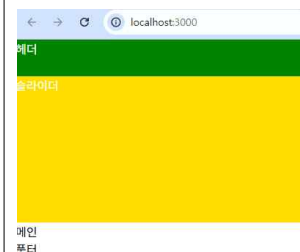
```

import React from 'react';

const Slider = (props) => {
  return (
    <div style={props.style}>
      슬라이더
    </div>
  );
};

export default Slider;
  
```

터미널 확인
npm start



| | |
|-----------------|--|
| style={{ ... }} | 부모 컴포넌트(App)에서 스타일을 자식(Slider)에게 전달함 |
| props | 자식 컴포넌트가 전달받은 속성을 담은 객체 |
| props.style | 부모가 보낸 스타일 객체를 꺼내어 <div>에 적용함 |
| props란 | props는 properties(속성)의 줄임말로, 부모 컴포넌트가 자식 컴포넌트에게 값을 전달할 때 사용하는 객체 props는 자식 컴포넌트가 외부 데이터를 전달받는 방법 |



| | | | |
|-------------------------|---|---------------------|--|
| components > Main.js | <pre> import React from 'react' function Main() { //라인 스타일을 객체 형태로 정의 const style2 = { width: '100%', height: '200px', backgroundColor: 'skyblue', color: '#fff' } return (<div style={style2}> 메인 </div>) } export default Main </pre> | 터미널 확인 npm start | |
| style2 | 스타일 속성을 가진 객체 | | |
| style={style2} | style={객체} 구조는 JSX에서 인라인 스타일을 적용하는 기본 문법 (JSX에서 style에 객체를 넣는 방식) | | |

| | |
|---|---|
| <div> <div> <div>EXPLORER</div> <div> <div>BLOG1</div> <div> <div>node_modules</div> <div>public</div> <div>src</div> <div> <div>components</div> <div> <div>Cow.js</div> <div>Footer.js</div> <div>Header.js</div> <div>Main.js</div> <div>Slider.js</div> <div>App.css</div> <div>App.js</div> <div>App.test.js</div> <div>Cat.js</div> </div> </div> </div> </div> <div> <div>App.css M X</div> <div>src > App.css > ...</div> <div> <div>38</div> <div>}</div> <div>39</div> <div>40</div> <div>41</div> <div>.footerStyle {</div> <div>42</div> <div>background-color: tomato;</div> <div>43</div> <div>color: #fff;</div> <div>44</div> <div>height: 100px;</div> <div>45</div> <div>}</div> <div>46</div> <div>47</div> </div> <div>코드작성</div> </div> </div> </div> | <div>App.css – 맨아래 추가</div> <div> <pre>.footerStyle { background-color: tomato; color: #fff; height: 100px; }</pre> </div> |
| <div> <div> <div>EXPLORER</div> <div> <div>BLOG1</div> <div> <div>node_modules</div> <div>public</div> <div>src</div> <div> <div>components</div> <div> <div>Cow.js</div> <div>Footer.js</div> <div>Header.js</div> <div>Main.js</div> <div>Slider.js</div> <div>App.css</div> <div>App.js</div> <div>App.test.js</div> <div>Cat.js</div> <div>Dog.js</div> <div>index.css</div> <div>index.js</div> <div>logo.svg</div> </div> </div> </div> </div> <div> <div>Footer.js U X</div> <div>src > components > Footer.js > ...</div> <div> <div>1</div> <div>import './App.css';</div> <div>2</div> <div>import React from 'react'</div> <div>3</div> <div>4</div> <div>function Footer() {</div> <div>5</div> <div>return (</div> <div>6</div> <div><div className="footerStyle"></div> <div>7</div> <div> 풋터</div> <div>8</div> <div></div></div> <div>9</div> <div>)</div> <div>10</div> <div>}</div> <div>11</div> <div>12</div> <div>export default Footer</div> <div>13</div> </div> <div>코드 추가</div> </div> </div> </div> | <div>Footer.js</div> <div> <pre>import './App.css'; import React from 'react' function Footer() { return (<div className="footerStyle"> 풋터 </div>) } export default Footer</pre> </div> |
| <div>터미널 확인</div> <div>npm start</div> | <div>localhost:3000</div> <div> <div>헤더</div> <div>슬라이더</div> <div>메인</div> <div>풋터</div> </div> |
| <div>className 속성</div> | <div>Footer 컴포넌트에서 외부 CSS 파일(App.css)의 클래스를 불러와서, className 속성을 통해 JSX 요소에 스타일을 적용하는 방식</div> |

| | |
|---|---|
| <div> <div> <div>EXPLORER</div> <div> <div>BLOG1</div> <div> <div>node_modules</div> <div>public</div> <div>src</div> <div> <div>components</div> <div> <div>Cow.js</div> <div>Footer.js</div> <div>Footer2.js</div> <div>Header.js</div> <div>Main.js</div> <div>Slider.js</div> <div>App.css</div> <div>App.js</div> <div>App.test.js</div> <div>Cat.js</div> <div>Dog.js</div> <div>index.css</div> <div>index.js</div> <div>logo.svg</div> </div> </div> </div> </div> <div> <div>App.css M X</div> <div>src > App.css > ...</div> <div> <div>39</div> <div>40</div> <div>41</div> <div>42</div> <div>43</div> <div>44</div> <div>45</div> <div>46</div> <div>47</div> <div>48</div> <div>49</div> <div>50</div> <div>51</div> <div>52</div> <div>53</div> </div> <div> <div>.footerStyle {</div> <div>background-color: tomato;</div> <div>color: #fff;</div> <div>height: 100px;</div> <div>}</div> <div>.footerStyle2 {</div> <div>background-color: red;</div> <div>color: #fff;</div> <div>height: 100px;</div> <div>}</div> </div> <div>코드추가</div> </div> </div> </div> | <div>App.css – 맨아래 추가</div> <div> <div>.footerStyle {</div> <div>background-color: tomato;</div> <div>color: #fff;</div> <div>height: 100px;</div> <div>}</div> <div>.footerStyle2 {</div> <div>background-color: red;</div> <div>color: #fff;</div> <div>height: 100px;</div> <div>}</div> </div> |
| <div> <div> <div>EXPLORER</div> <div> <div>BLOG1</div> <div> <div>node_modules</div> <div>public</div> <div>src</div> <div> <div>components</div> <div> <div>Cow.js</div> <div>Footer.js</div> <div>Footer2.js</div> <div>Header.js</div> <div>Main.js</div> <div>Slider.js</div> <div>App.css</div> <div>App.js</div> <div>App.test.js</div> <div>Cat.js</div> <div>Dog.js</div> <div>index.css</div> <div>index.js</div> <div>logo.svg</div> </div> </div> </div> </div> <div> <div>Footer2.js U X</div> <div>src > components > Footer2.js > ...</div> <div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> <div>8</div> <div>9</div> <div>10</div> <div>11</div> <div>12</div> </div> <div> <div>import React from 'react'</div> <div>function Footer() {</div> <div>return (</div> <div><div className='footerStyle2'></div> <div> 풋터</div> <div></div></div> <div>)</div> <div>}</div> <div>export default Footer</div> </div> <div>컴포넌트 추가</div> <div>코드추가</div> </div> </div></div> | <div>Footer2.js</div> <div> <div>import React from 'react'</div> <div>function Footer() {</div> <div>return (</div> <div><div className='footerStyle2'></div> <div> 풋터</div> <div></div></div> <div>)</div> <div>}</div> <div>export default Footer</div> </div> |
| <div> <div> <div>EXPLORER</div> <div> <div>BLOG1</div> <div> <div>node_modules</div> <div>public</div> <div>src</div> <div> <div>components</div> <div> <div>Cow.js</div> <div>Footer.js</div> <div>Footer2.js</div> <div>Header.js</div> <div>Main.js</div> <div>Slider.js</div> <div>App.css</div> <div>App.js</div> <div>App.test.js</div> <div>Cat.js</div> <div>Dog.js</div> <div>index.css</div> <div>index.js</div> <div>logo.svg</div> <div>reportWebVitals.js</div> <div>setupTests.js</div> <div>.gitignore</div> <div>package-lock.json</div> <div>package.json</div> <div>README.md</div> </div> </div> </div> </div> <div> <div>App.js M X</div> <div>src > App.js > ...</div> <div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> <div>8</div> <div>9</div> <div>10</div> <div>11</div> <div>12</div> <div>13</div> <div>14</div> <div>15</div> <div>16</div> <div>17</div> <div>18</div> <div>19</div> <div>20</div> <div>21</div> </div> <div> <div>import './App.css';</div> <div>import Header from './components/Header';</div> <div>import Slider from './components/Slider';</div> <div>import Main from './components/Main';</div> <div>import Footer from './components/Footer';</div> <div>import Footer2 from './components/Footer2';</div> <div>function App() {</div> <div>return (</div> <div><></div> <div><Header/></div> <div><Slider style={{width:"100%", color:"white", height:"200px", backgroundColor:"gold"}}></Slider></div> <div><Main > </Main></div> <div><Footer></Footer></div> <div><Footer2/></div> <div></></div> <div>);</div> <div>}</div> <div>export default App;</div> </div> <div>코드추가</div> </div> </div></div> | |
| <div>App.js</div> <div>코드추가 전역 범위(global)</div> | <div> <div>import './App.css';</div> <div>import Header from './components/Header';</div> <div>import Slider from './components/Slider';</div> <div>import Main from './components/Main';</div> <div>import Footer from './components/Footer';</div> <div>import Footer2 from './components/Footer2';</div> <div>function App() {</div> <div>return (</div> <div><></div> <div><Header/></div> <div><Slider style={{width:"100%", color:"white", height:"200px", backgroundColor:"gold"}}></Slider></div> <div><Main > </Main></div> <div><Footer></Footer></div> <div><Footer2/></div> <div></></div> <div>);</div> </div> |

| | |
|-----------------------------|---|
| | <pre>} export default App;</pre> |
| <p>터미널 확인 npm start</p> |  |
| <p>className 속성</p> | <p>Footer 컴포넌트에서 외부 CSS 파일(App.css)의 클래스를 불러와서, className 속성을 통해 JSX 요소에 스타일을 적용하는 방식</p> |

Main 컴포넌트에 자식 컴포넌트(Dog, Cow) 포함하기

EXPLORER

BLOG1

node_modules

public

src

components

Cow.js

Footer.js

Footer2.js

Header.js

Main.js

Slider.js

App.css

App.js

App.test.js

Cat.js

Dog.js

index.css

index.js

logo.svg

reportWebVitals.js

setupTests.js

.gitignore

package-lock.json

package.json

README.md

원포인트.png

Main.js

src > components > Main.js > ...

1 import React from 'react'

2 import Dog from '../Dog';

3 import Cow from './Cow';

4

5 function Main() {

6 const style2 = {

7 width: '100%',

8 height: '200px',

9 backgroundColor: 'skyblue',

10 color: '#fff'

11 }

12 return (

13 <div style={style2}>

14 메인

15 <Dog></Dog>

16 <Cow></Cow>

17 </div>

18)

19 }

20

21 export default Main

22

23

Main.js

import React from 'react'

import Dog from '../Dog';

import Cow from './Cow';

function Main() {

const style2 = {

width: '100%',

height: '200px',

backgroundColor: 'skyblue',

color: '#fff'

}

return (

<div style={style2}>

메인

<Dog></Dog>

<Cow></Cow>

</div>

)

}

export default Main

← → ↺ ⓘ localhost:3000

헤더

슬라이더

메인

강아지

소

풋터

풋터

터미널 확인

npm start

- 31 -