

프로그래밍 전문 훈련기관. 민코딩

MacBook 에서 C++

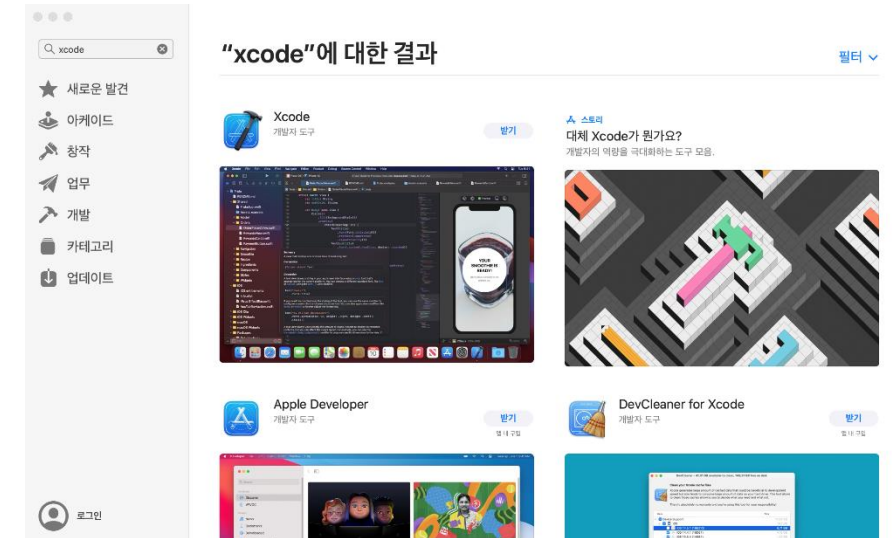


필수 프로그램 설치

설치 할 프로그램

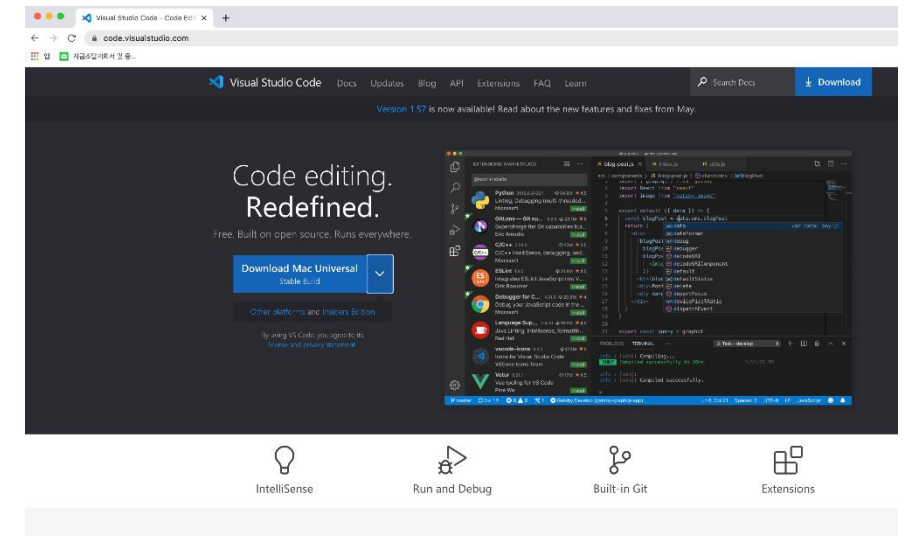
1. Xcode 설치

- App Store에서 설치



2. Visual Studio Code 설치

- code.visualstudio.com 에서 접속하여 다운로드 및 설치



[참고] Xcode

Apple의 공식 IDE

- Apple 기기에서 동작하는 App 만들 때 사용
(안드로이드 App 개발 불가)
Swift 라는 프로그래밍 언어를 사용하여 App 개발
- C / C++ / Java / Python 지원

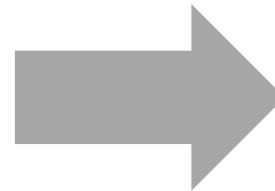
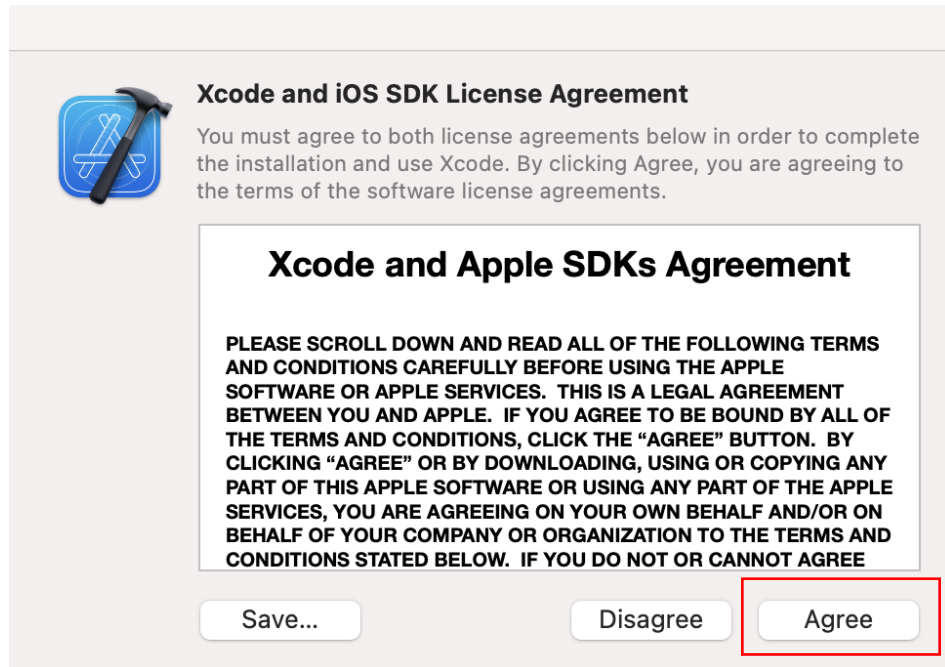


Xcode를 설치하는 이유

- 컴파일러 (번역기) 가 맥북에 자동 설치되어 편리한 코딩 준비 가능.
- Xcode는 사용하지 않는다.

Xcode 설치 후

설치 완료 후 1회성으로 실행을 해야 한다.
약관에 Agree를 눌러주고, Xcode를 종료한다.

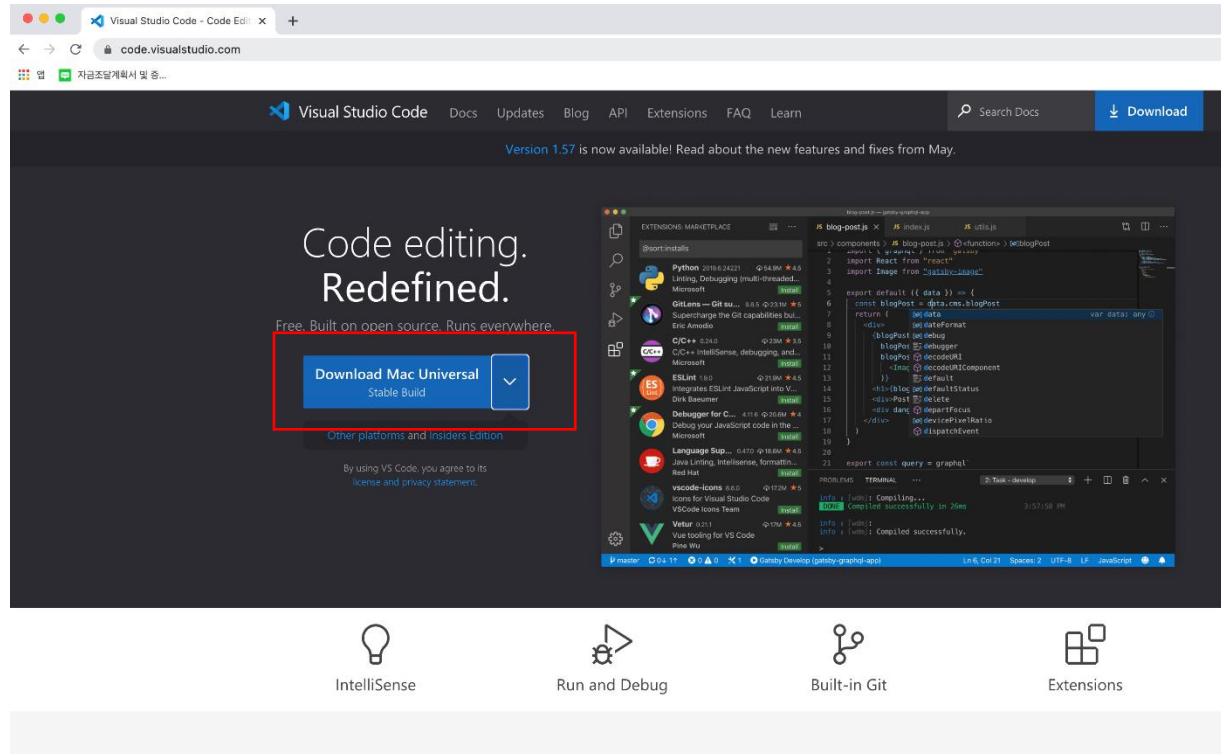


MacBook
컴퓨터 재부팅을
꼭 해주세요

Visual Studio Code 다운로드

Visual Studio 공식 홈페이지에서 다운로드 받고, 설치한다.

- code.visualstudio.com



[참고] Visual Studio for Mac과 차이

Visual Studio for Mac

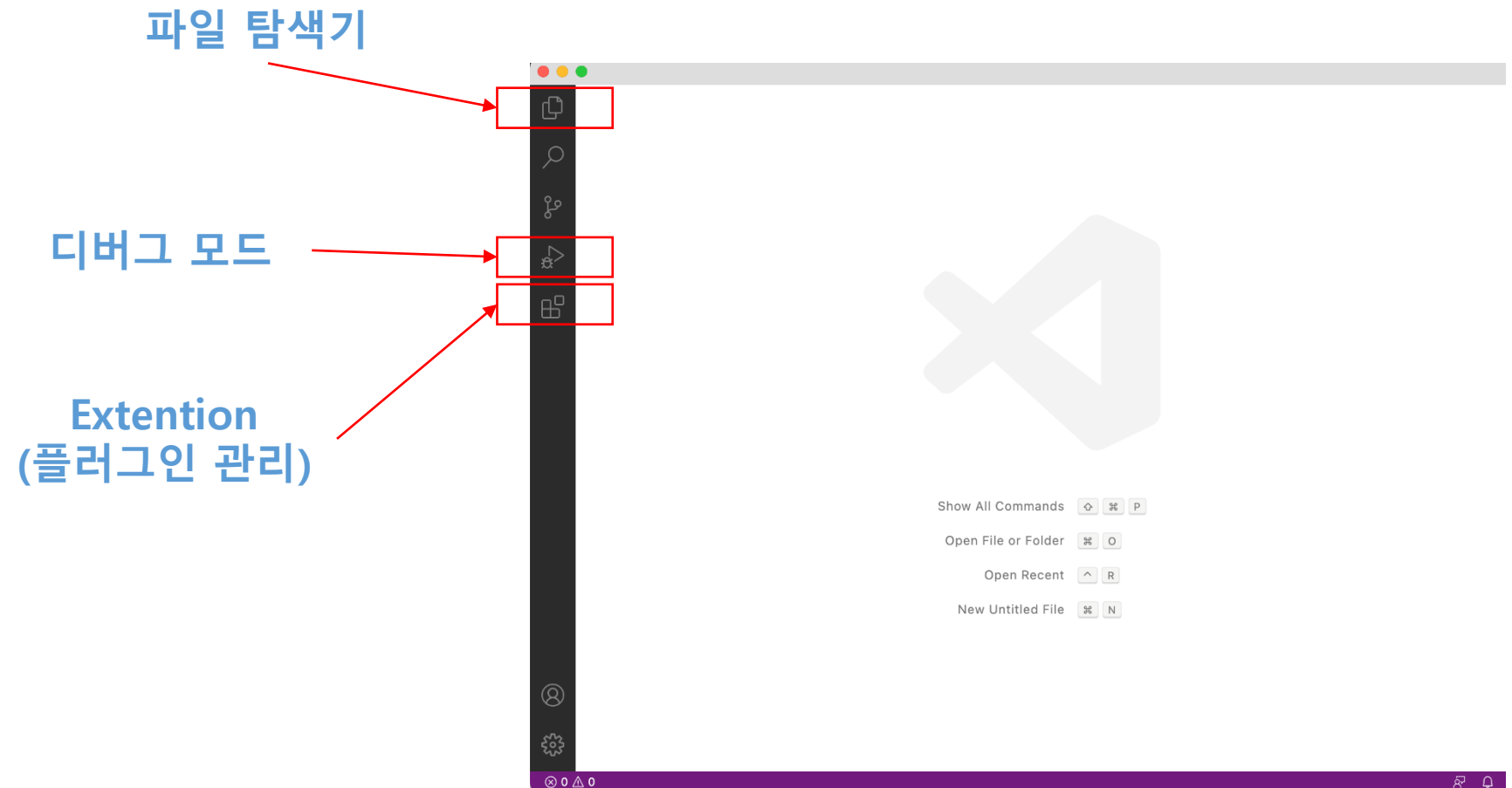
- 컴파일러, 에디터, 디버깅 도구 등 프로그래밍에 필요한 기능이 모두 포함되어 있는 종합세트 프로그램
- Mac에서는 C++ 을 지원하지 않는다.

Visual Studio Code

- 컴파일러 없는, 에디터만 존재
- 필요한 기능은 Extention (플러그인) 으로 추가할 수 있다.

Visual Studio Code 실행

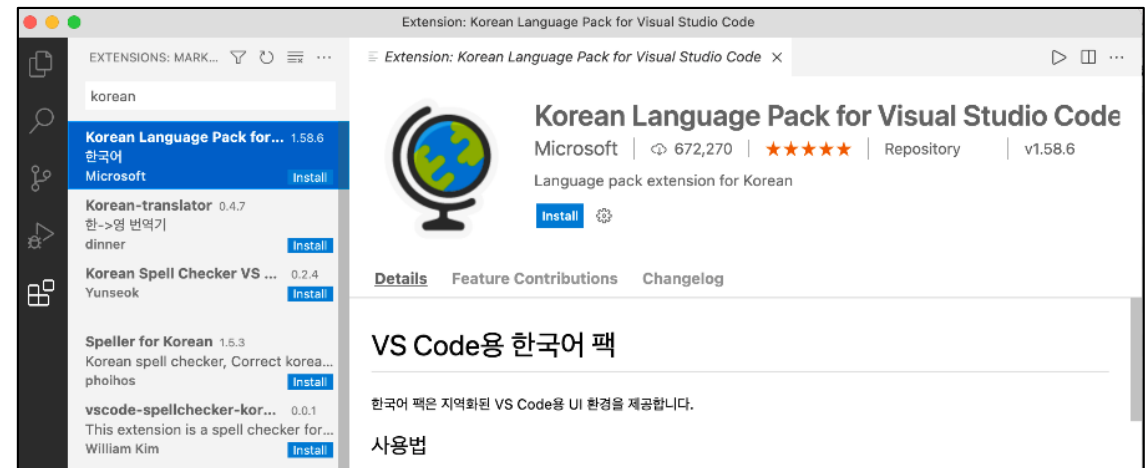
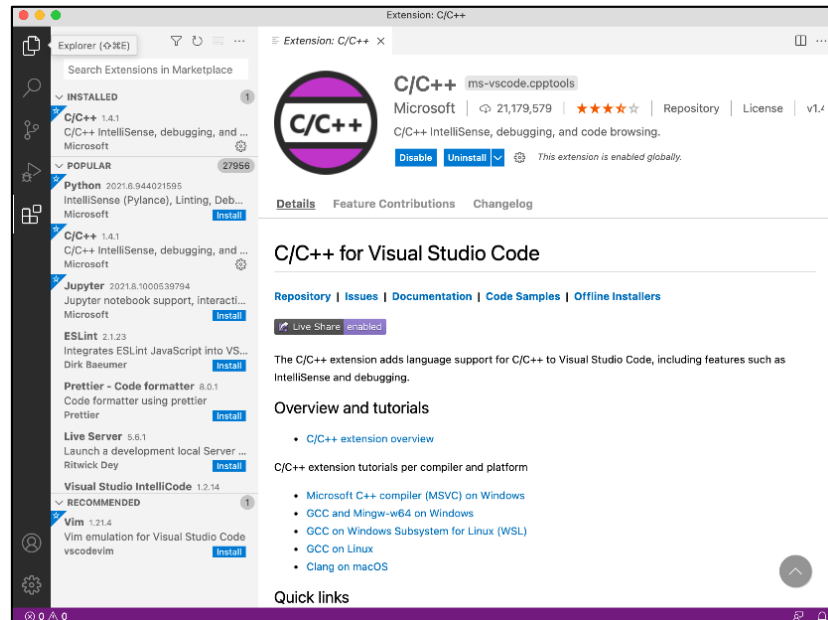
세 가지 메뉴
사용 예정



Extention 설치 하기

두 가지 Extention (플러그인) 설치

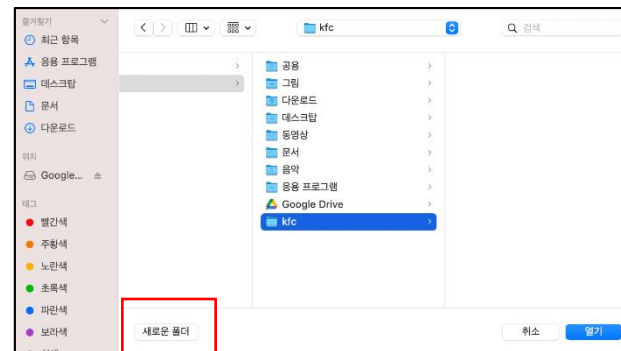
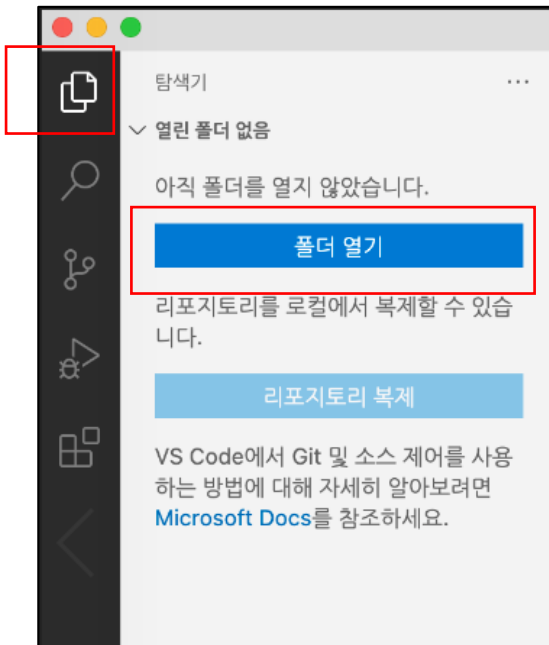
1. C / C++ : 디버깅 기능 / 자동완성 기능
2. Korean Language Pack : 한글화



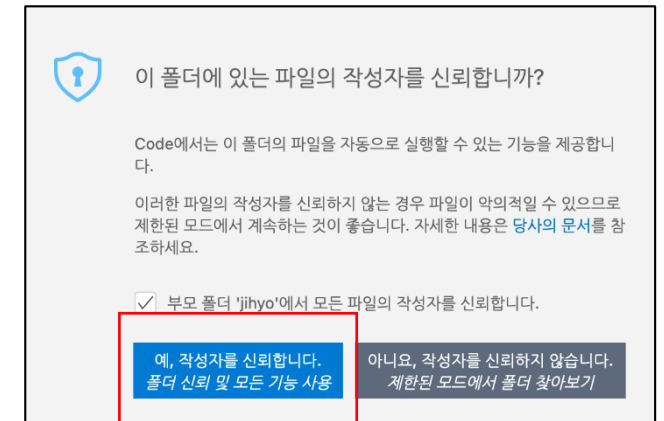
빌드 후 실행

Visual Studio Code 규칙

폴더를 하나 만들어 두고,
그 내부에 cpp 파일을 만들어서 코딩한다.



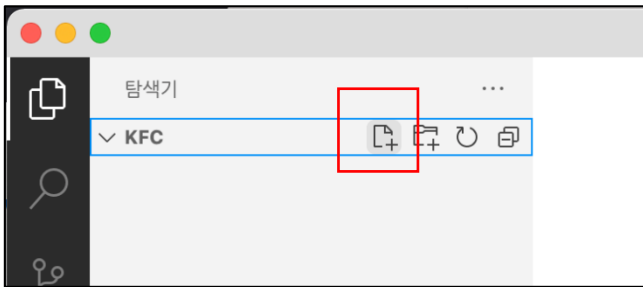
KFC 폴더 하나 만들었음



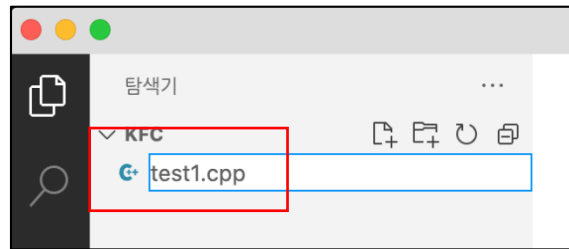
폴더를 신뢰함 선택

폴더에 파일 생성하기

- KFC 쪽에 마우스 포인터를 가져다 대면, 생성 아이콘이 나옴
- test1.cpp 파일을 추가 후, 기본 코드 작성



마우스를 가져다 대야
아이콘이 나온다.



```
G+ test1.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "KFC\n";
7      cout << "MC\n";
8
9      return 0;
10 }
```

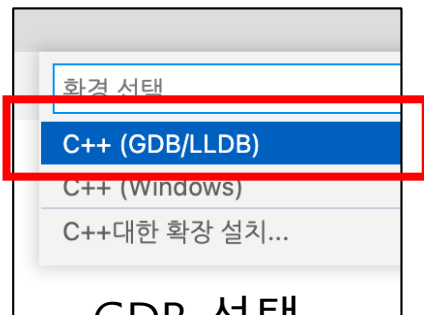
빌드 준비 단계

Build 하기 위한 세팅 파일 생성이 필요함

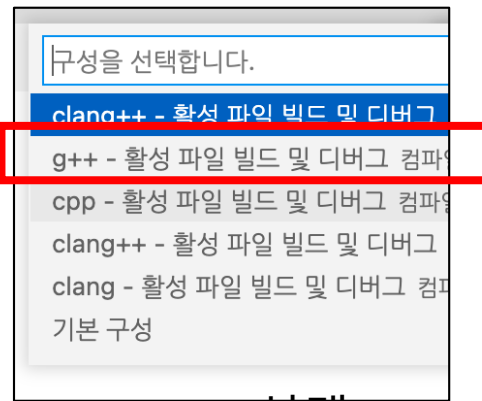
- 아래대로 수행하면 자동으로 설정 파일 생성
- 1회성으로 수행 필요



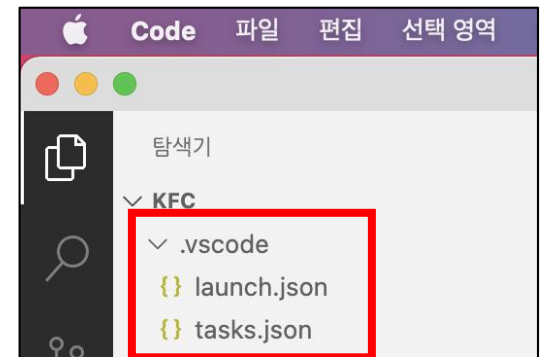
디버깅 시작



GDB 선택



g++ 선택

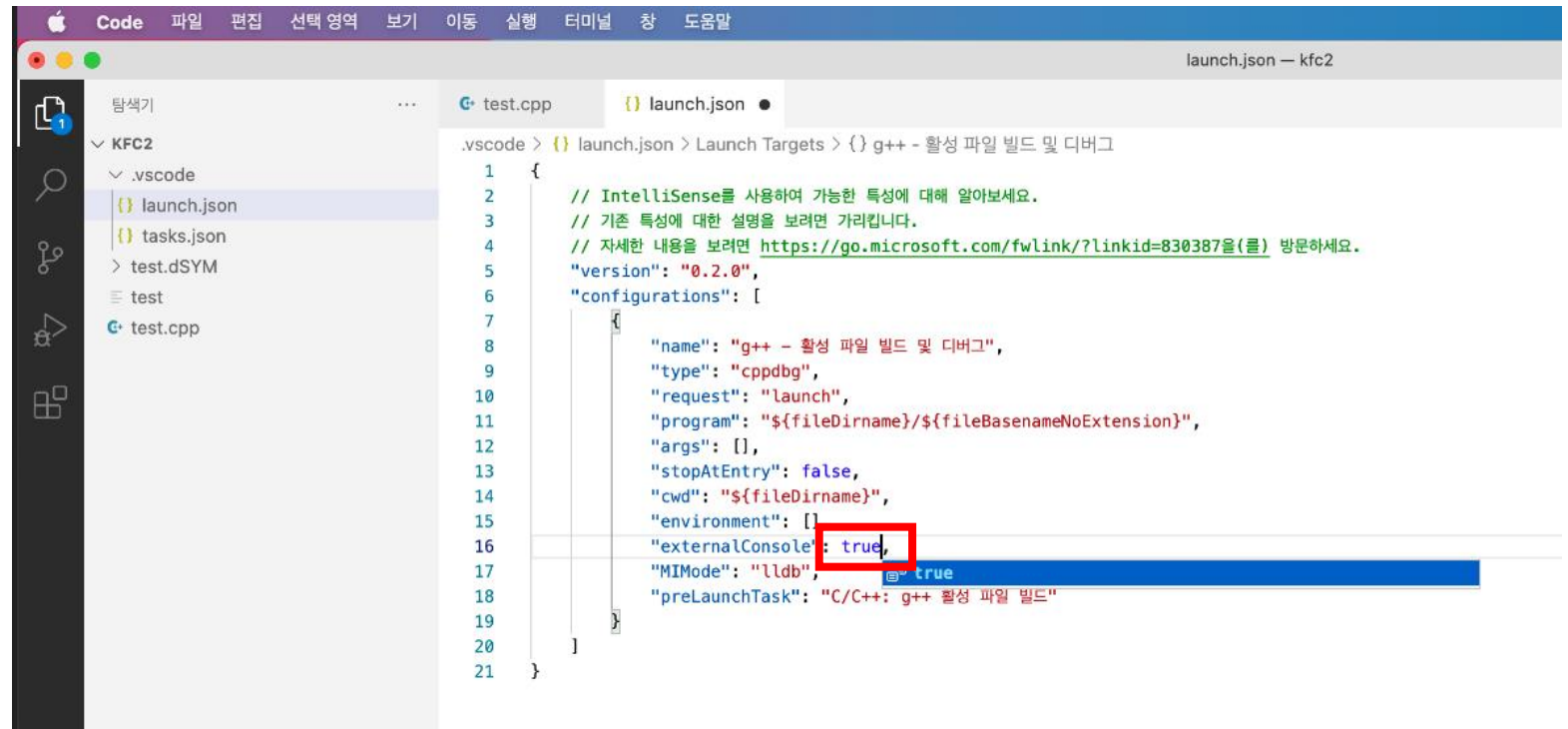


설정파일
생성됨

한 가지 옵션 수정

외부 콘솔창 사용 false를 true로 변경하기

- 이 옵션을 켜야, cout 뿐만 아니라 cin 도 정상 수행됨



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a project named 'KFC2' with a folder '.vscode' containing 'launch.json' and 'tasks.json'. The 'launch.json' file is selected. The main editor displays the 'launch.json' file with the following content:

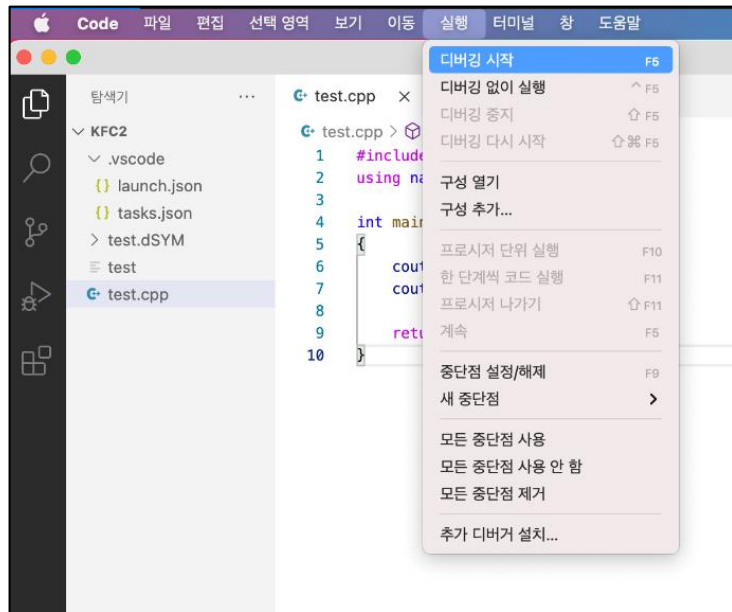
```
.vscode > {} launch.json > Launch Targets > {} g++ - 완성 파일 빌드 및 디버그
1 {
2   // IntelliSense를 사용하여 가능한 특성에 대해 알아보세요.
3   // 기존 특성에 대한 설명을 보려면 가리킵니다.
4   // 자세한 내용을 보려면 https://go.microsoft.com/fwlink/?linkid=830387을(를) 방문하세요.
5   "version": "0.2.0",
6   "configurations": [
7     {
8       "name": "g++ - 완성 파일 빌드 및 디버그",
9       "type": "cppdbg",
10      "request": "launch",
11      "program": "${fileDirname}/${fileBasenameNoExtension}",
12      "args": [],
13      "stopAtEntry": false,
14      "cwd": "${fileDirname}",
15      "environment": [],
16      "externalConsole": true,
17      "MIMode": "lldb",
18      "preLaunchTask": "C/C++: g++ 완성 파일 빌드"
19    }
20  ]
21 }
```

The value 'true' for the 'externalConsole' property is highlighted with a red rectangle.

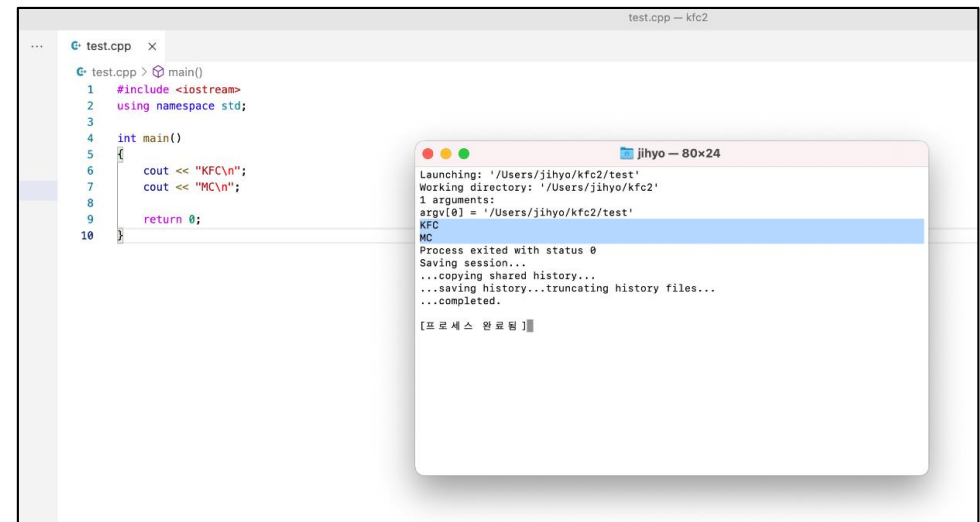
빌드 테스트

실행 > 디버깅 시작

- 다시 한번 클릭하면 정상적으로 빌드 후 실행까지 됨



단축키 : F5

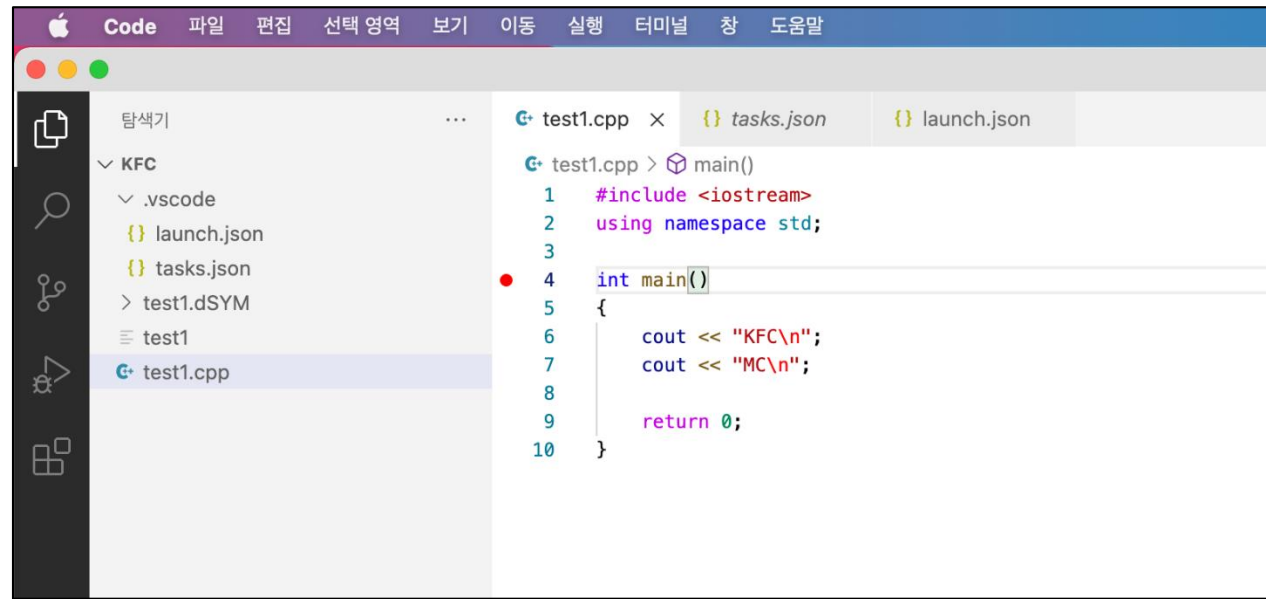


창 닫기 단축키 : command + w

Trace

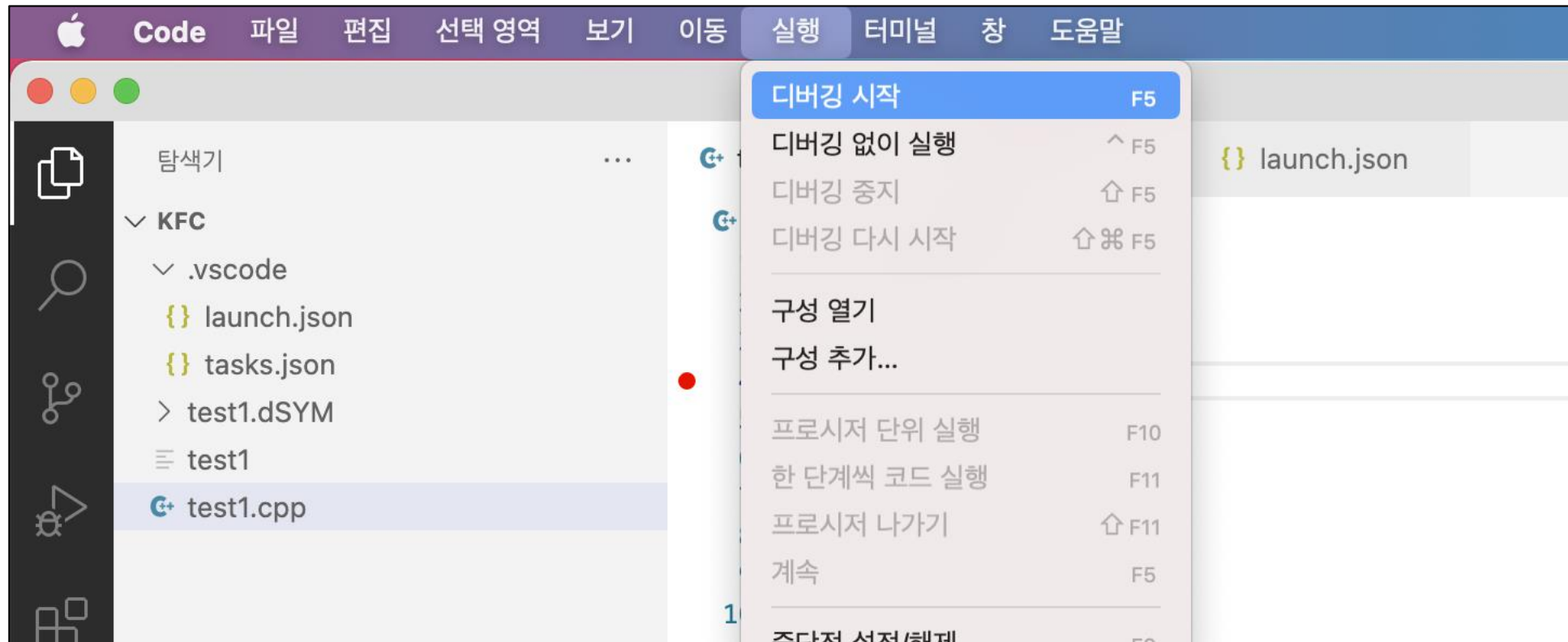
중단점 추가하기

int main() 코드 클릭 후, F9 를 눌러
빨간점 (Breakpoint)이 나오도록 만든다.



디버깅 시작

F5를 누르거나, 다음 메뉴에서 "디버깅 시작" 버튼을 누른다.



Trace 시작하기

첫 번째 코드에서 노란색 줄이 생성된다.

- F10 을 누르면 Trace가 가능하다.
- 디버깅 종료는 Shift + F5 를 누르면 된다.

