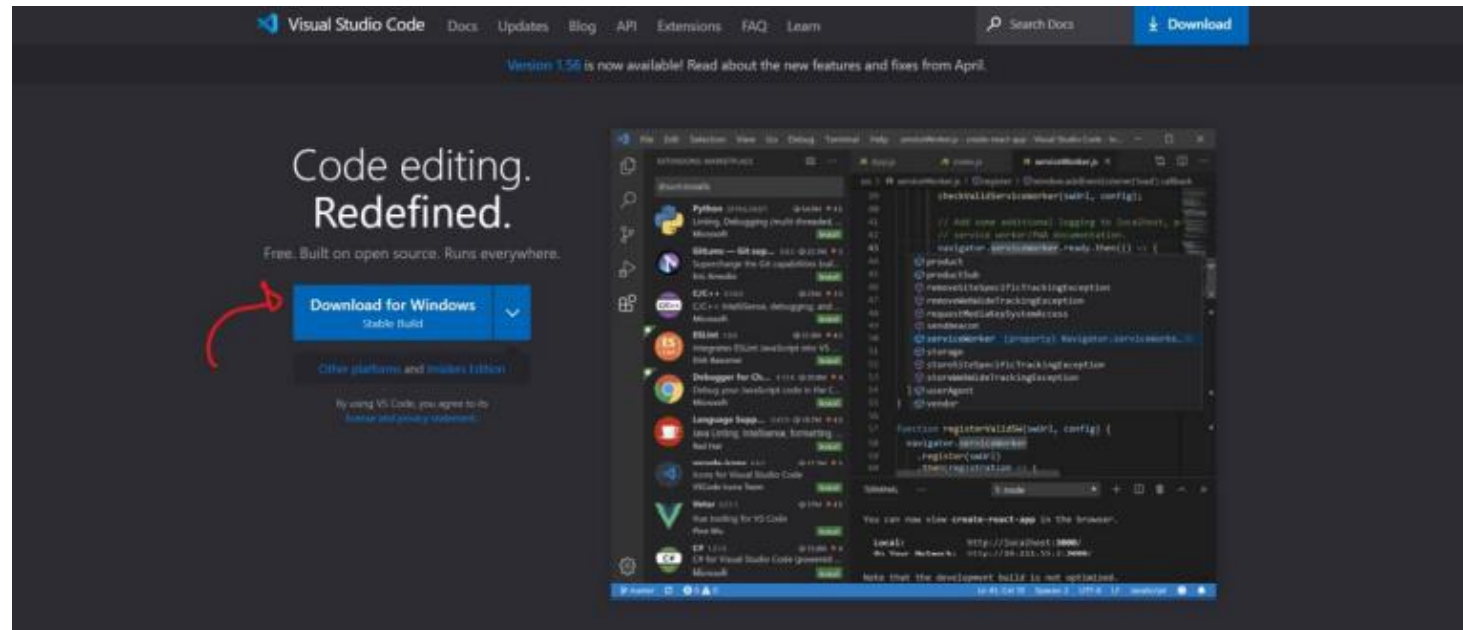


# C 프로그래밍 환경 설정



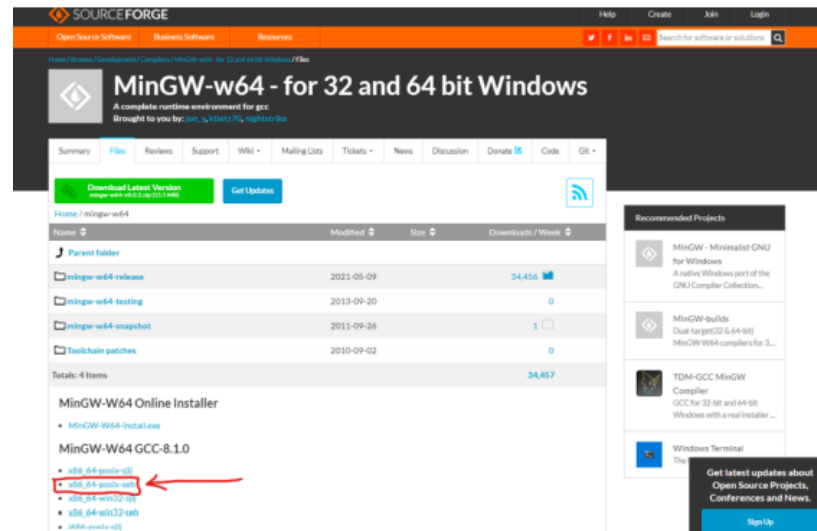
# Visual Studio Code 설치

- Visual Studio Code는 코드 작성 프로그램입니다.
- 다음 사이트 접속: <https://code.visualstudio.com/>
- 아래 버튼으로 각자 운영체제에 맞는 설치파일을 다운 받고 설치
  - 모두 next 를 누르면 됩니다.



# gcc 설치 (Windows)

- Windows 운영 체제에만 해당합니다.
- Mingw는 Windows에서 gcc를 사용하게 해 줍니다.
- 다음 사이트 접속: <https://sourceforge.net/projects/mingw-w64/files/mingw-w64/>
- x86\_64-posix-seh 을 클릭하면 압축 파일이 받아집니다.



# gcc 설치 (Windows)

- 다운 받은 파일을 압축 해제하면 mingw64 폴더가 있을 것입니다. 해당 폴더를 로컬 디스크(C:)에 위치시킵니다.

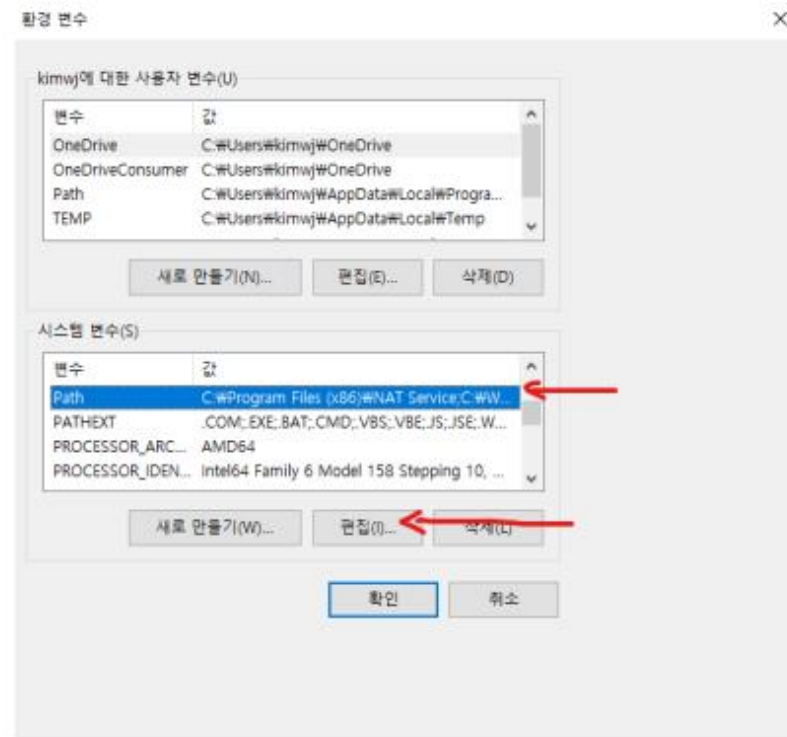
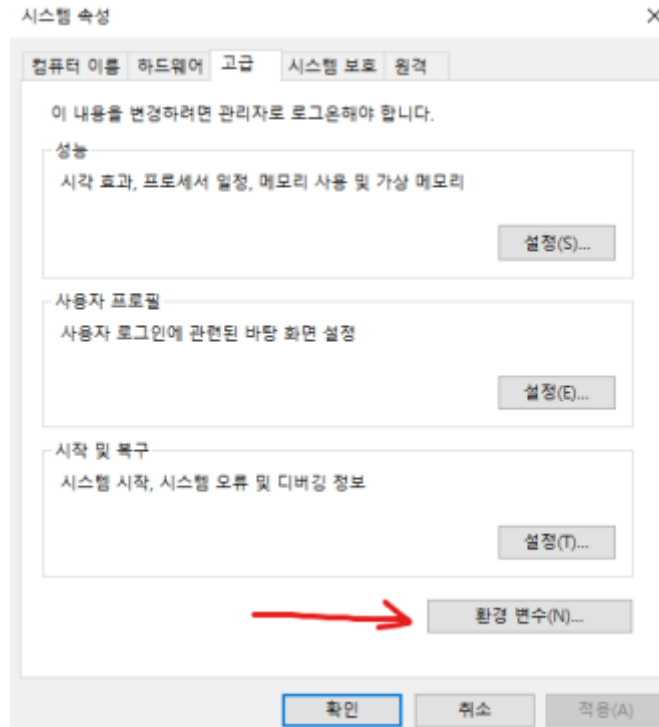


- 윈도우 실행 창에 sysdm.cpl을 입력하고 엔터를 치면 ‘시스템 속성’ 창이 뜹니다.



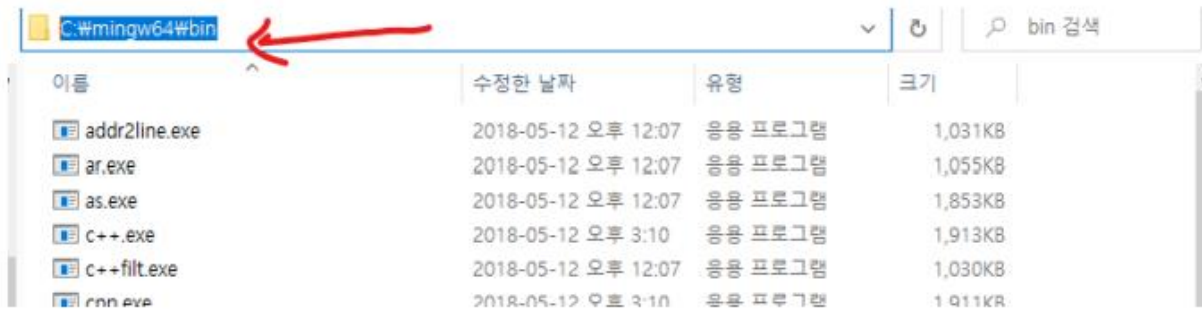
# gcc 설치 (Windows)

- ‘고급’ 탭의 ‘환경 변수’ 클릭
- ‘시스템 변수’ 칸에서 ‘Path’를 선택 후 ‘편집’ 클릭

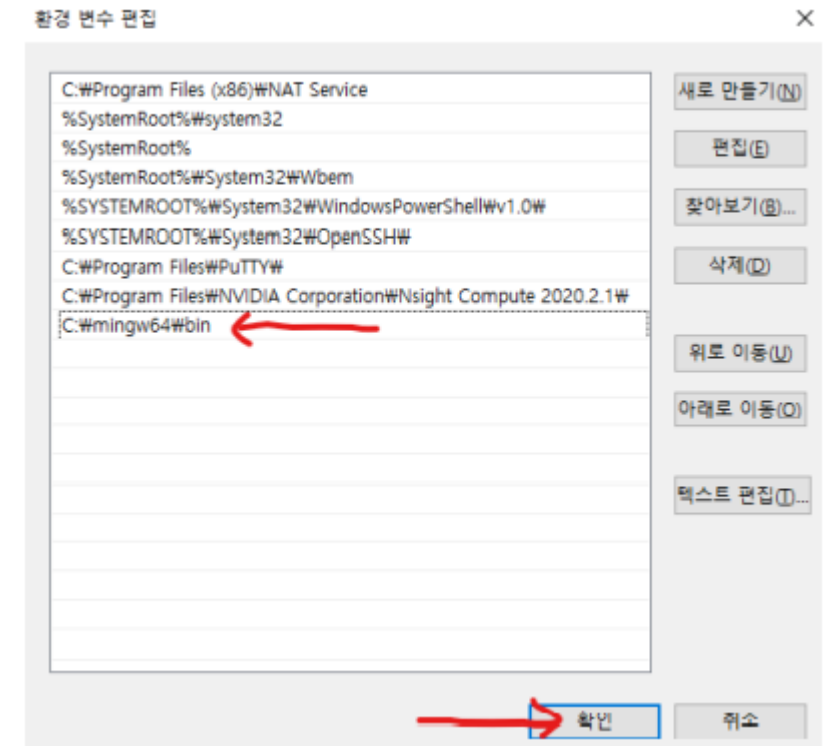


# gcc 설치 (Windows)

- 새로 만들기를 누르고 아까 C드라이브에 위치 시킨 mingw64 폴더의 bin 폴더의 위치를 입력합니다.



- 그 다음 열렸던 창에 모두 확인을 누릅니다.





# gcc 설치 (Windows)

- 윈도우 검색 창에 cmd를 입력해서 커맨드 창을 띄운 뒤 gcc -v 를 입력했을 때, 다음과 같이 설치가 된 것을 확인할 수 있습니다.



```
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Wkimwj>gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=C:/mingw64/bin/./libexec/gcc/x86_64-w64-mingw32/8.1.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../../src/gcc-8.1.0/configure --host=x86_64-w64-mingw32 --build=x86_64-w64-mingw32 --target=x86_64-w64-mingw32 --prefix=/mingw64 --with-sysroot=/c/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64 --enable-shared --enable-static --disable-multilib --enable-languages=c,c++,fortran,lto --enable-libstdcxx-time=yes --enable-threads=posix --enable-libgomp --enable-libatomic --enable-lto --enable-graphite --enable-checking=release --enable-fully-dynamic-string --enable-version-specific-runtime-libs --disable-libstdcxx-pch --disable-libstdcxx-debug --enable-bootstrap --disable-rpath --disable-win32-registry --disable-nls --disable-werror --disable-symvers --with-gnu-as --with-gnu-ld --with-ard-nocoma --with-tune=core2 --with-libiconv --with-system-zlib --with-gmp=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --with-mpfr=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --with-mpc=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --with-isl=/c/mingw810/prerequisites/x86_64-w64-mingw32-static --with-pkgversion='x86_64-posix-seh-rev0, Built by MinGW-W64 project' --with-bugurl=https://sourceforge.net/projects/mingw-w64 CFLAGS='-O2 -pipe -fno-ident -I/c/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c/mingw810/prerequisites/x86_64-zlib-static/include -I/c/mingw810/prerequisites/x86_64-w64-mingw32-static/include' CXXFLAGS='-O2 -pipe -fno-ident -I/c/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c/mingw810/prerequisites/x86_64-zlib-static/include -I/c/mingw810/prerequisites/x86_64-w64-mingw32-static/include' CPPFLAGS='-I/c/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/include -I/c/mingw810/prerequisites/x86_64-zlib-static/include -I/c/mingw810/prerequisites/x86_64-w64-mingw32-static/include' LDFLAGS='-pipe -fno-ident -L/c/mingw810/x86_64-810-posix-seh-rt_v6-rev0/mingw64/opt/lib -L/c/mingw810/prerequisites/x86_64-zlib-static/lib -L/c/mingw810/prerequisites/x86_64-w64-mingw32-static/lib'
Thread model: posix
gcc version 8.1.0 (x86_64-posix-seh-rev0, Built by MinGW-W64 project)
```

# gcc 설치 (MacOS)

- **MacOS**인 경우, terminal을 열고 terminal에 다음과 같이 명령어를 입력합니다.

```
xcode-select --install
```

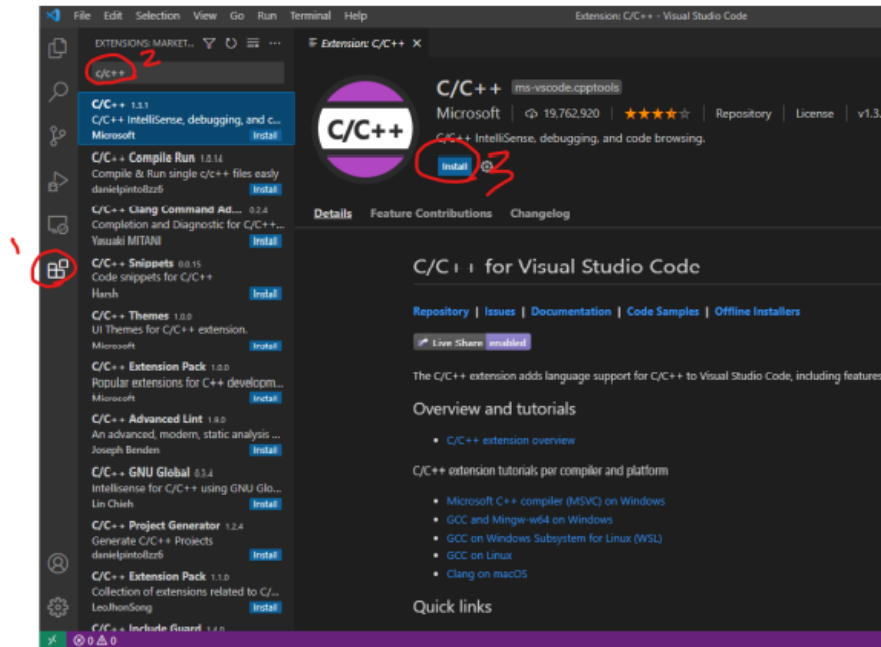
- 설치 창이 뜨면 install 버튼으로 설치를 진행합니다.
- 설치가 끝나면 terminal에 다음과 같이 입력했을 때, 오류가 발생하지 않아야 합니다.

```
gcc -v
```



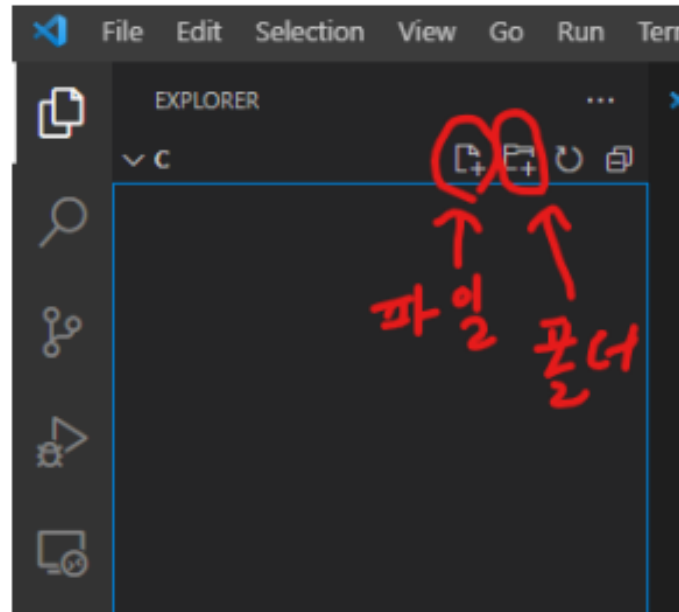
# C 코드 작성

- Visual Studio Code를 실행하고, 다음과 같이 ‘C/C++’ extension을 설치합니다. 오른쪽 아래에
- 설치 중이라는 알림이 뜨고, 설치가 다 되면 Visual Studio Code를 껐다가 다시 엽니다.



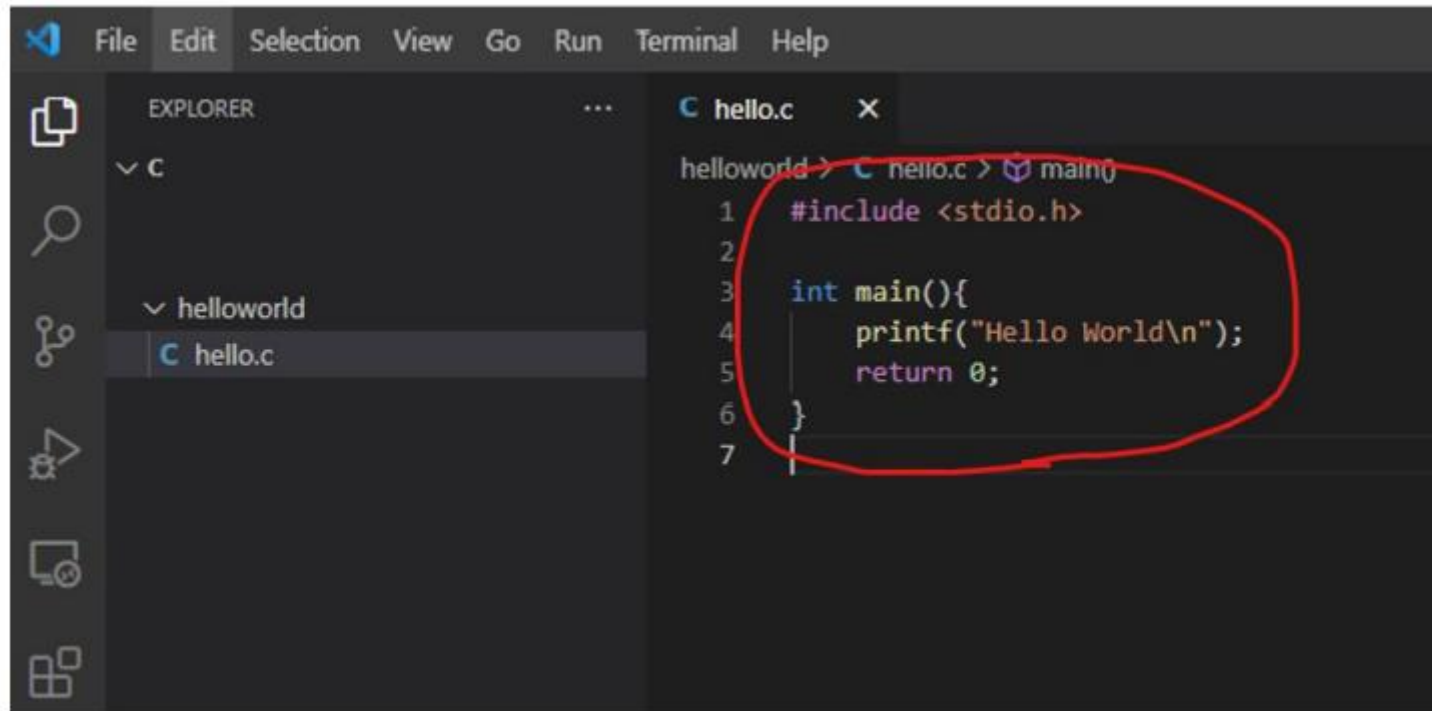
# C 코드 작성

- 왼쪽의 Open folder를 눌러서 작업을 진행할 폴더를 선택합니다.
- 작업할 위치에서 다음과 같이 파일이나 폴더를 추가할 수 있습니다.  
저는 helloworld라는 폴더를 만들었습니다.



# C 코드 작성

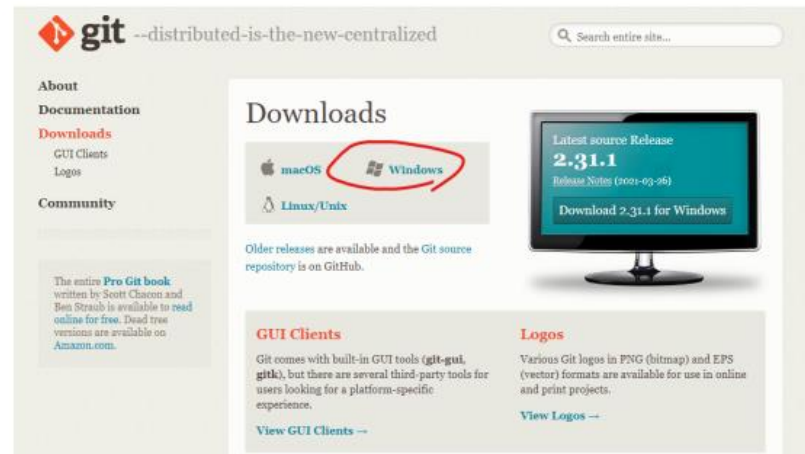
- 이제 파일 추가 버튼을 누른 다음 hello.c 라는 파일을 만들고 다음과 같이 입력을 하고 ctrl+s로 저장을 합니다.



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  c
  helloworld
    C hello.c
C hello.c
1  #include <stdio.h>
2
3  int main(){
4      printf("Hello World\n");
5      return 0;
6  }
7
```

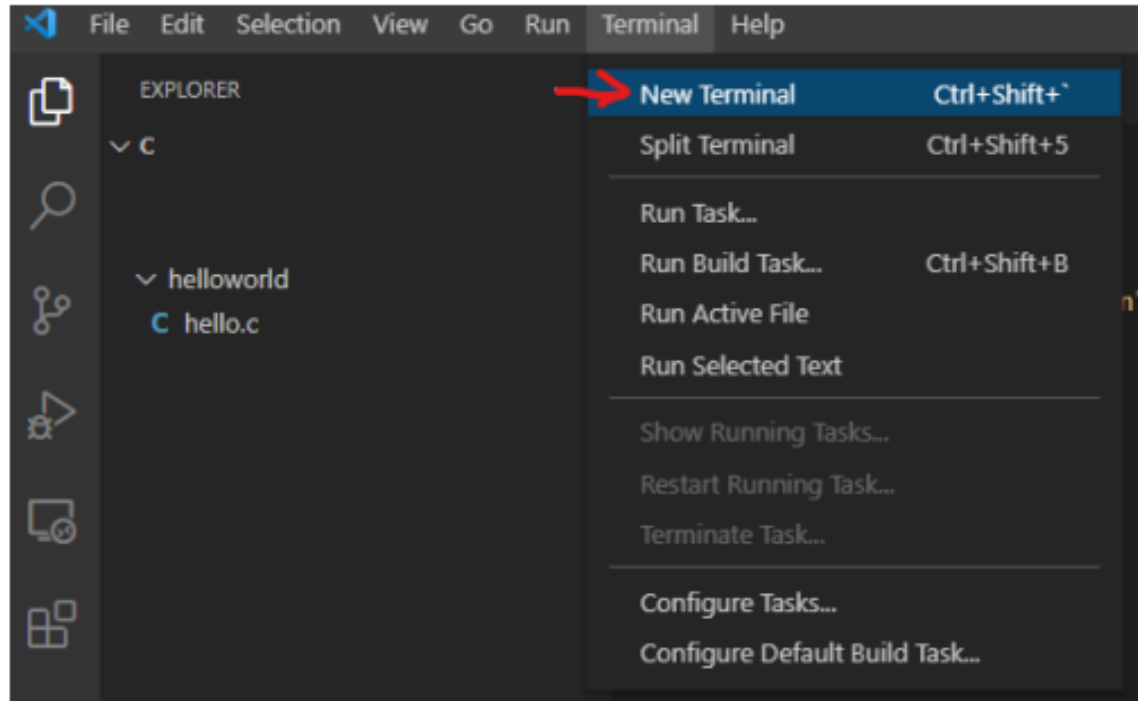
# Git bash 설치

- **Windows 운영체제**인 경우, Visual Studio Code에서 linux terminal을 사용하기 위해 git bash를 설치합니다. MacOS인 경우 굳이 설치할 필요 없습니다.
- 다음 사이트에서 git bash를 설치합니다. <https://git-scm.com/downloads>
- 아래 버튼을 눌러서 실행파일을 받고 다 next를 누르면 됩니다.



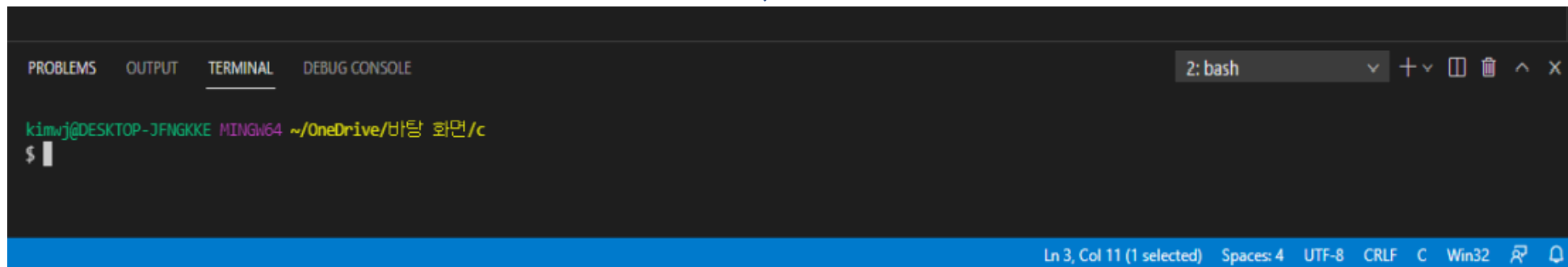
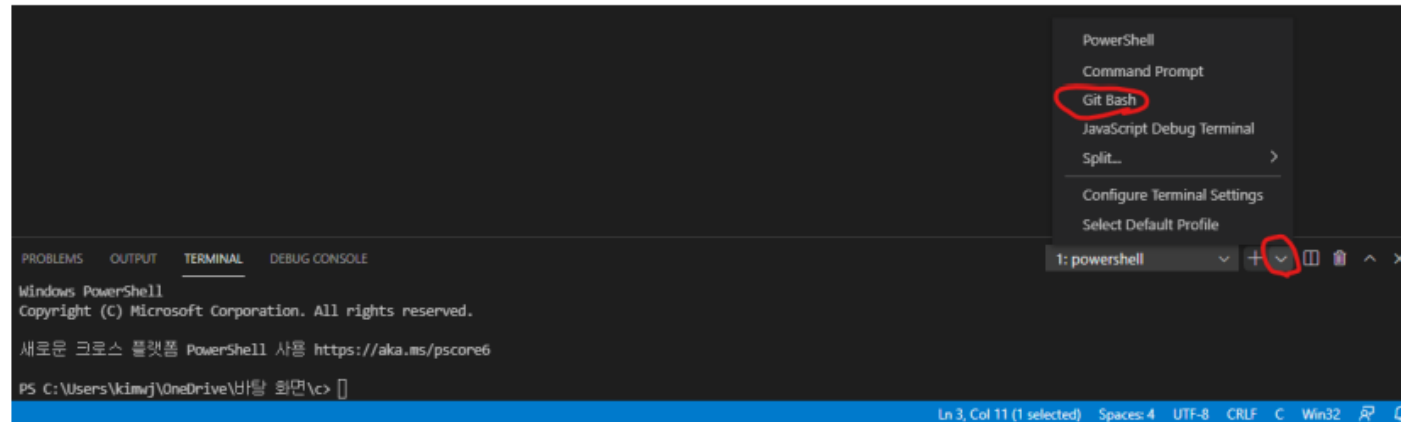
# C 코드 컴파일

- 다시 Visual Studio Code로 돌아와서, Terminal → New Terminal을 클릭하면 아래에 terminal이 뜹니다.



# C 코드 컴파일

- Window의 경우, 다음과 같이 아까 설치한 Git Bash를 클릭하면 terminal이 바뀝니다.



# C 코드 컴파일

- 이제 terminal에서 이전에 작성한 hello.c 파일의 위치로 가야 합니다. 다음 명령어를 이용합니다.
  - `ls` → 현재 directory에 있는 파일, 폴더 목록이 뜹니다.
  - `cd 폴더이름` → 해당 폴더로 이동
  - 파일 또는 폴더 이름을 적을 때, `tab` 버튼을 누르면 자동 완성이 됩니다. (prefix가 유일한 경우)



The screenshot shows a terminal window with the following commands and output:

```
kimwj@DESKTOP-JFNGKKE MINGW64 ~/OneDrive/바탕 화면/c
$ ls
helloworld/

kimwj@DESKTOP-JFNGKKE MINGW64 ~/OneDrive/바탕 화면/c
$ cd helloworld/

kimwj@DESKTOP-JFNGKKE MINGW64 ~/OneDrive/바탕 화면/c/helloworld
$ ls
hello.c ← 확인!

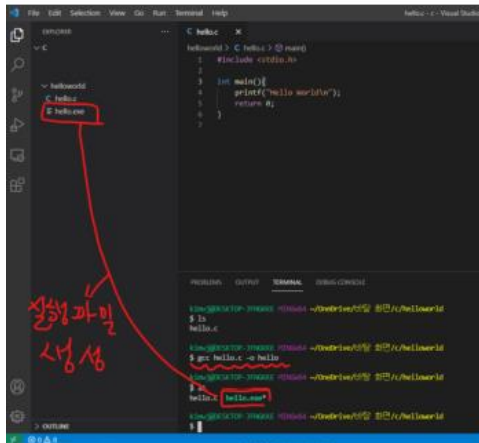
kimwj@DESKTOP-JFNGKKE MINGW64 ~/OneDrive/바탕 화면/c/helloworld
$
```

A red arrow points from the handwritten text "확인!" (Check!) to the file "hello.c" in the output of the second `ls` command.



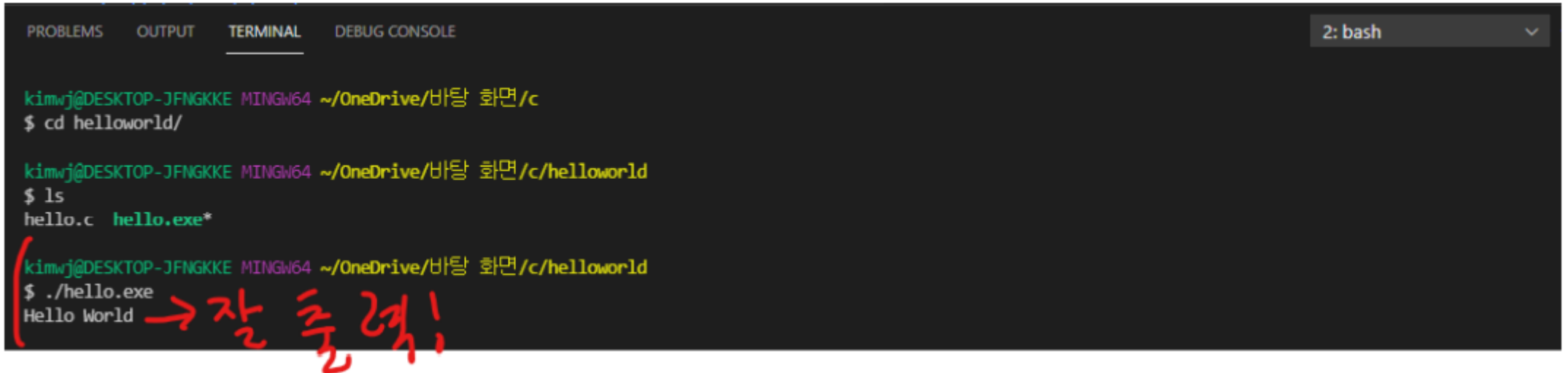
# C 코드 컴파일

- hello.c 를 찾았으면, 이제 컴파일을 합니다. 다음과 같이 terminal에 입력합니다.
  - `gcc hello.c -o hello`
  - (hello.c 파일을 컴파일해서 hello라는 실행파일을 만들겠다는 의미)
- 그러면 hello라는 실행파일이 생성됩니다. (MacOS의 경우, .exe가 없이 그냥 hello 일 것입니다.)



# C 코드 컴파일

- 다음과 같이 입력해서 만들어진 실행 파일을 실행합니다.
  - ./hello.exe (MacOS의 경우: ./hello)
- Hello World가 출력되면 제대로 실행된 것입니다.
- 실행이 잘 되지 않을 경우 구글링 및 슬랙 질의응답 게시판을 이용해 주세요.



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 2: bash
kimwj@DESKTOP-JFNGKKE MINGW64 ~/OneDrive/바탕 화면/c
$ cd helloworld/

kimwj@DESKTOP-JFNGKKE MINGW64 ~/OneDrive/바탕 화면/c/helloworld
$ ls
hello.c  hello.exe*

kimwj@DESKTOP-JFNGKKE MINGW64 ~/OneDrive/바탕 화면/c/helloworld
$ ./hello.exe
Hello World → 잘 출력!
```

Thanks!