

# 데이터 구조 개발환경

ECE20010

한동 대학교

김영섭 교수

idebtor@gmail.com

Atom  
GCC

# Piazza App 소개

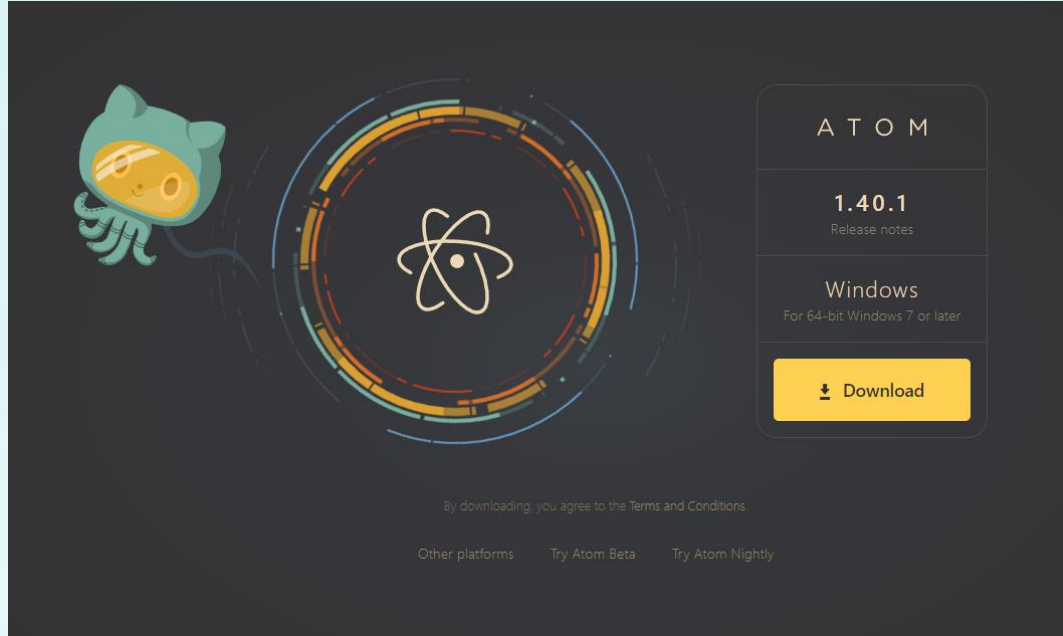
---



- 수업에서 활용하는 Piazza 사이트는 안드로이드, IOS 스토어에서 다운로드 받을 수 있습니다. 모바일에서 편리하게 이용하세요.
- 안드로이드: <https://play.google.com/store/apps/details?id=com.piazza.android&hl=ko>
- IOS: <https://apps.apple.com/us/app/piazza/id453142230>

# Atom 설치하기

- <https://atom.io/>로 접속합니다.

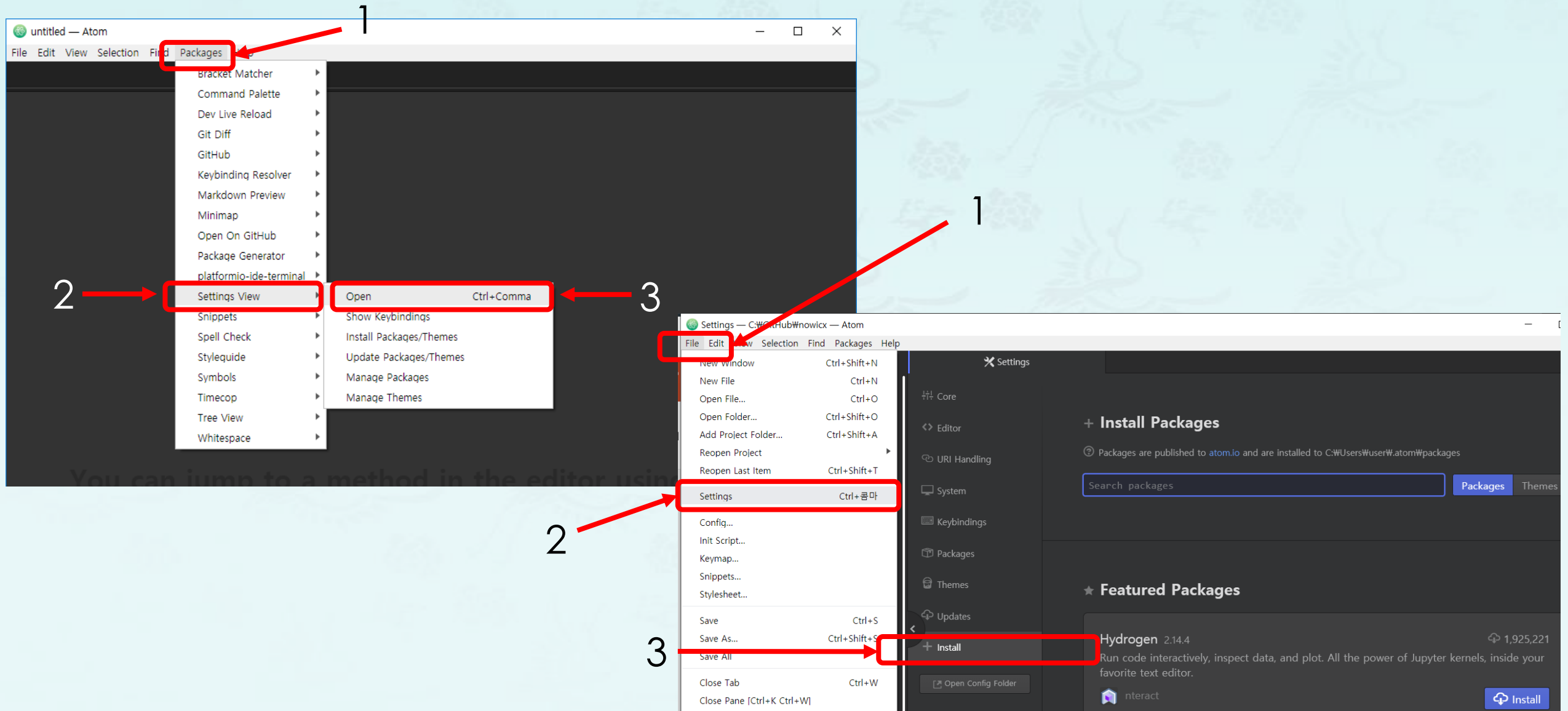


- AtomSetup-x64.exe를 실행하면 자동으로 Atom이 설치가 완료됩니다.
- 다음 폴더를 사용자 혹은 시스템의 환경 변수 **Path**에 추가하면 편리합니다.  
**C:\Users\<username>\AppData\Local\atom\bin**

↖  
your own user's name

# Atom 패키지 설치하기

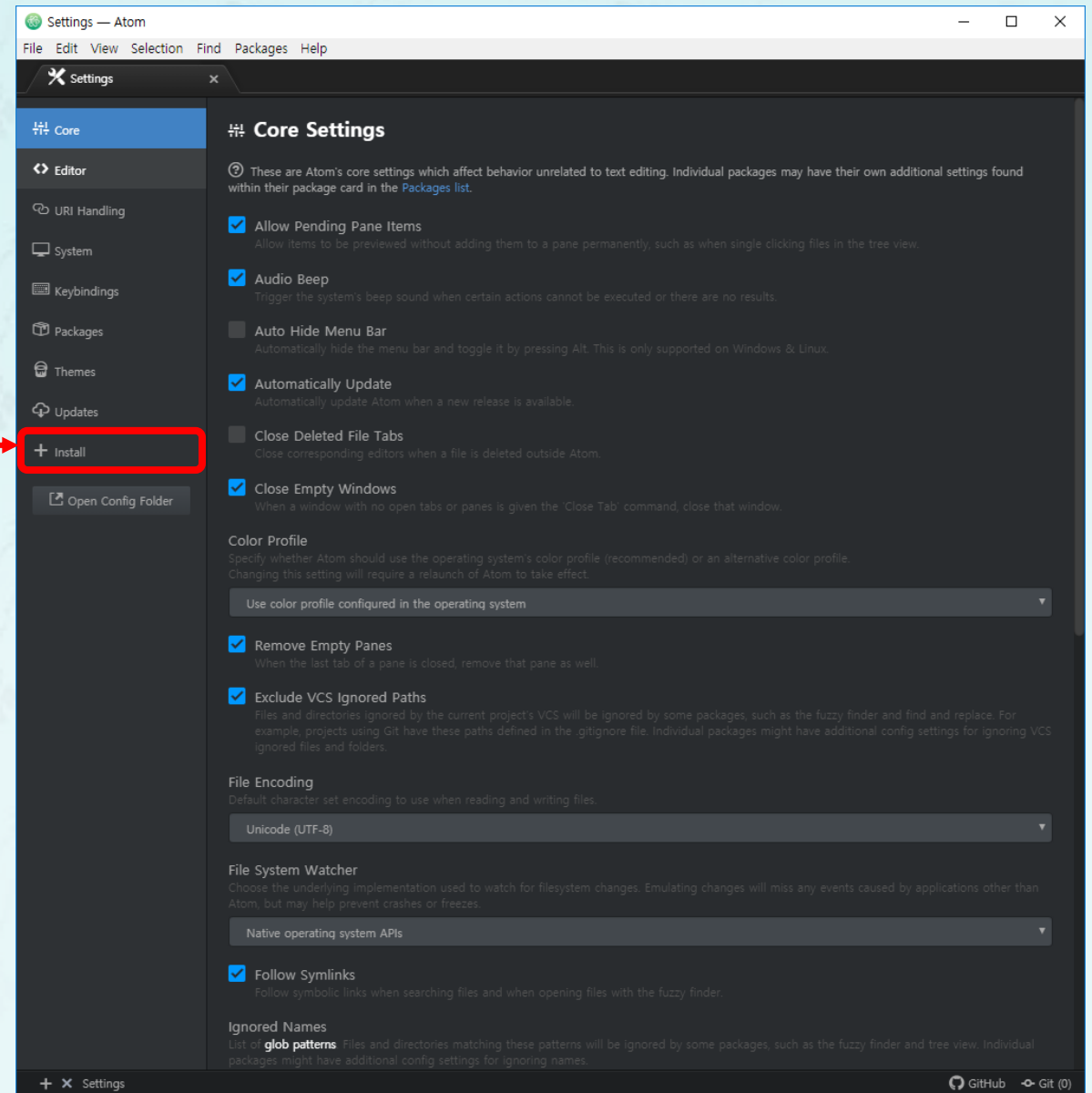
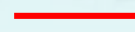
- atom을 시작하고, Package 혹은 File 메뉴를 통해 "Install Package" 페이지로 이동.



# Atom 패키지 설치하기

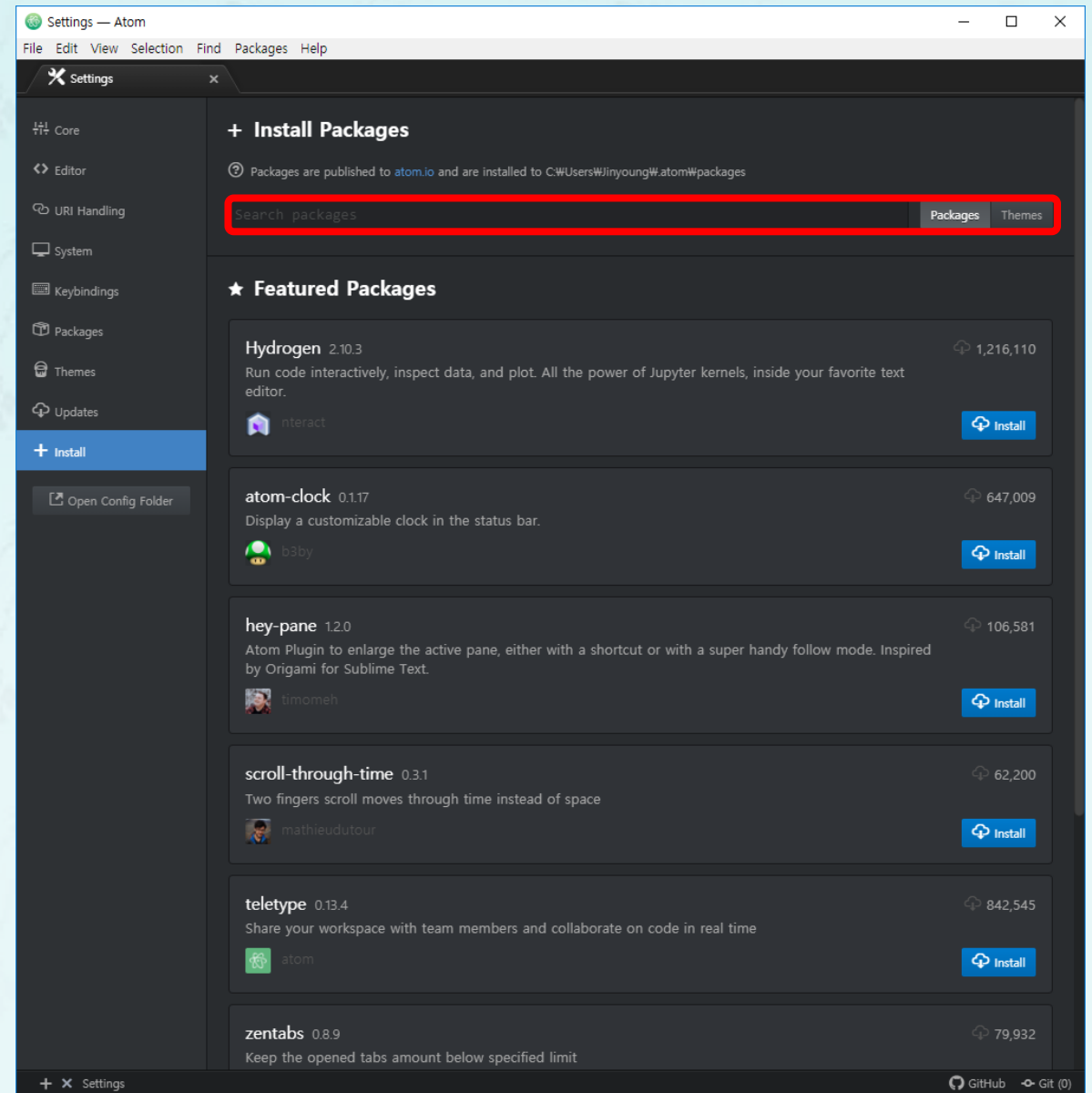
- Setting에서 Install를 클릭합니다.

클릭



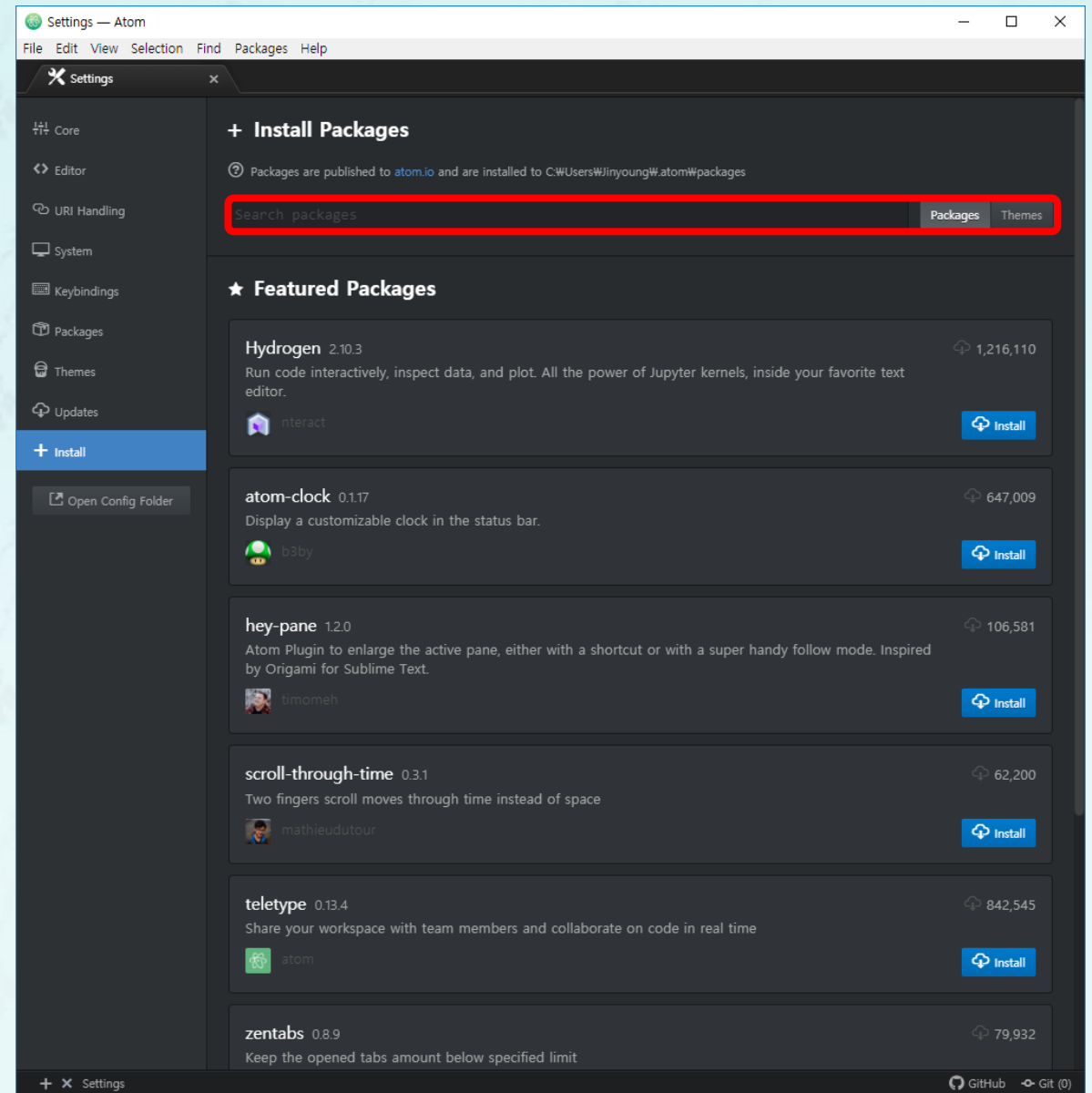
# Atom 패키지 설치하기

- 검색창에 다음의 패키지 이름들을 검색하여 설치합니다.
- platformio-ide-terminal
- file-icons
- minimap
- autocomplete-clang



# Atom 패키지 설치하기

- 검색창에 다음의 패키지 이름들을 검색하여 설치합니다.
- platformio-ide-terminal
- file-icons
- minimap
- autocomplete-clang





# g++ for macOS

---

- Open a terminal. (command + space, type terminal in the search field.)
- Enter the following command and run it.
  - `$ xcode-select -install`
- Check the installation with the following command.  
(If you see some version numbers, it was successfully installed.)
  - `$ g++ -v`
- Check the option such that all filename extensions are to be displayed.  
[Finder] → [Preferences ...] → [Advanced] → Show all filename extensions



# g++ for macOS

- Use this code to test that g++ can compile and run the program.

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello, World!\n";
    return 0;
}
```

1. Copy the above program and save the file as **hello.cpp** on your desktop.
2. Open a terminal and type **cd Desktop**
3. To compile, enter the following in the terminal and run it. We see the file **hello** appear on the desktop under an exec icon.

```
$ g++ -std=c++11 hello.cpp -o hello
```

4. Run the compiled program by typing the following in the terminal. You will see "Hello World!" printed in the terminal.

```
$ ./hello
```

# MSYS2/MinGW-w64 for Windows

- 두 종류의 패키지, 즉 **MSYS2** 와 **MinGW-w64** 를 설치해야 합니다.  
MSYS2를 **먼저** 설치하지 않으면 어려움을 겪는 경우가 종종 발생합니다.

## MSYS2 설치하기

- <http://www.MSYS2.org/>로 접속합니다.

MSYS2를 통해 MinGW-w64를 설치를 위해  
아래 website를 참조해도 좋습니다.

- <https://vl0011.tistory.com/14>
- <https://stackoverflow.com/questions/30069830/how-to-install-mingw-w64-and-msys2/30071634#30071634>

## MSYS2

Software Distribution and Building Platform for Windows

MSYS2 is a collection of tools and libraries providing you with an easy-to-use environment for building installing and running native Windows software.

It consists of a command line terminal called [mintty](#), bash, version control systems like git and subversion tools like tar and awk and even build systems like autotools, all based on a modified version of [Cygwin](#), some of these central parts being based on Cygwin, the main focus of MSYS2 is to provide a build environment for native Windows software and the Cygwin-using parts are kept at a minimum. MSYS2 provides up-to-date native builds for GCC, mingw-w64, CPython, CMake, Meson, OpenSSL, FFmpeg, Rust, Ruby, just to name a few.

To provide easy installation of packages and a way to keep them updated it features a package management system called [Pacman](#), which should be familiar to Arch Linux users. It brings many powerful features like dependency resolution and simple complete system upgrades, as well as straight-forward and reproducible package building. Our package repository contains [more than 1800 pre-built packages](#) ready to install.

For more details see '[What is MSYS2?](#)' which also compares MSYS2 to other software distributions and development environments like [Cygwin](#), [WSL](#), [Chocolatey](#), [Scoop](#), ... and '[Who Is Using MSYS2?](#)' to see which projects are using MSYS2 and what for.

### Installation

1. Download the installer: [msys2-x86\\_64-20200720.exe](#)

SHA256 checksum: [ad6b58b355a852d9df7c6dc2722d0eca139042cf262ae7363e0a5e7dd2a6c5c9](#)

2. Run the installer. MSYS2 requires 64 bit Windows 7 or newer.

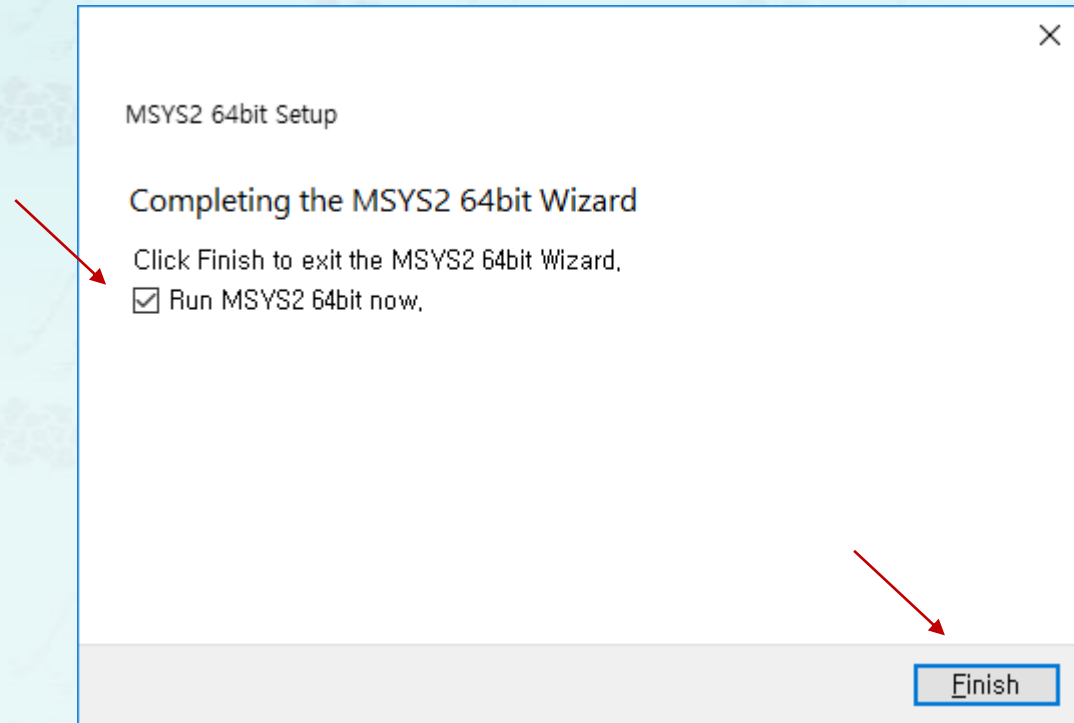
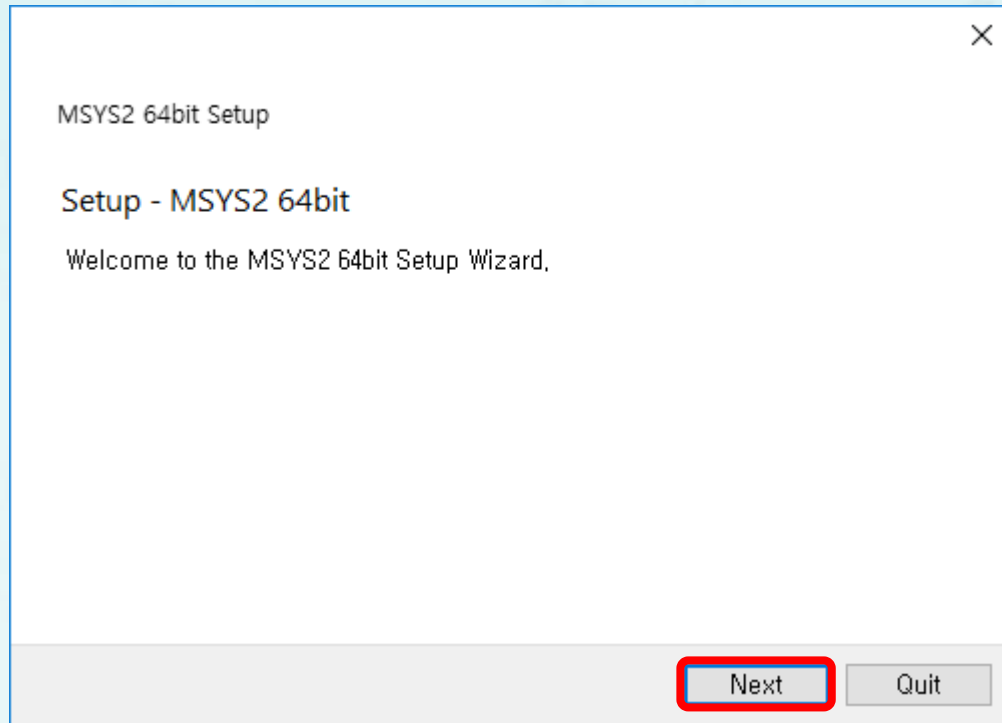
3. Click "Next"

클릭

# MSYS2/MinGW-w64 for Windows

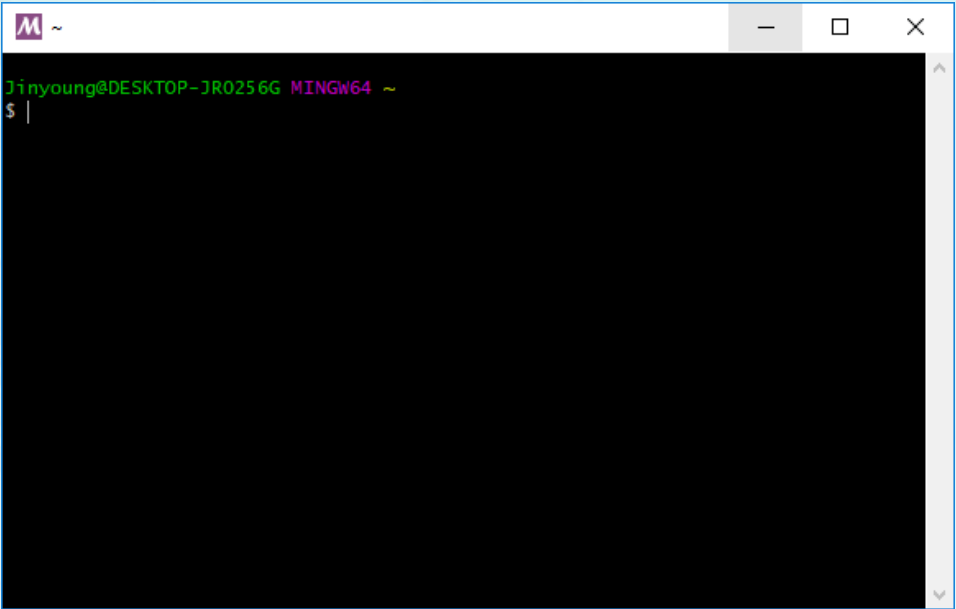
## MSYS2 설치하기

- MSYS2-x86\_64-20200720.exe를 실행합니다. 혹은 최신 버전으로.
- Finish를 클릭하여 MSYS2를 실행합니다.



# MSYS2/MinGW-w64 for Windows

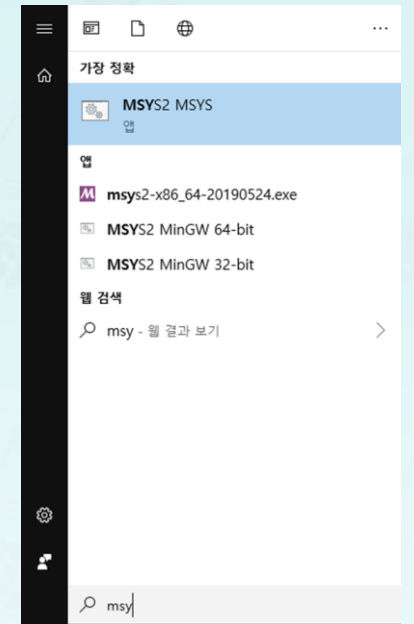
## MSYS2 설치하기

- MSYS2를 실행한 화면 입니다.
- 
- Command Shell (cmd) 혹은 PowerShell에서 “**pacman -Syu**”를 입력하고 Enter 키를 눌러 패키지 목록과 MSYS2를 업데이트 합니다.
  - “::설치를 진행하시겠습니까? [Y/N]”라는 질문이 나오면 Enter 키를 누릅니다.
  - 실행 결과를 보면 “경고:terminate MSYS2 without returning....”, “경고: for example close your terminal window...”이라는 메시지가 나오면 경고를 무시하고 다음을 진행합니다.

# MSYS2/MinGW-w64 for Windows

## GCC(GNU Compiler Collection) 설치하기

- MSYS2를 실행한 화면(콘솔)에서 다음을 입력하여 GCC를 설치하십시오.  
**pacman -S mingw-w64-x86\_64-gcc**
  - 만약, MSYS2 실행 화면(콘솔)이 없어졌으면, 윈도우 검색창에 “MSYS2”를 입력하거나 MSYS2를 찾아가서 MSYS2 msys를 실행하여 콘솔을 다시 생성하고 명령어를 입력 하십시오.
- 설치할 때 질문이 나오면, <Enter>로 응답하면 됩니다.
- Toolchain 설치 - make, gdb 등 gcc와 함께 쓰이는 툴들을 설치합니다.  
**pacman -S mingw-w64-x86\_64-toolchain**  
설치할 때 질문이 나오면, <Enter>로 응답하면 됩니다.

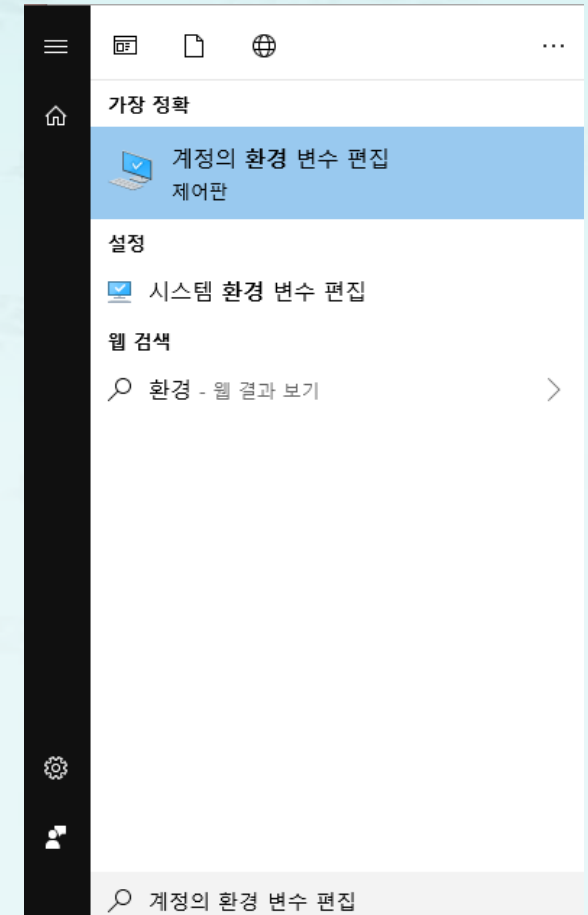




# MSYS2/MinGW-w64 for Windows

## 환경변수 설정

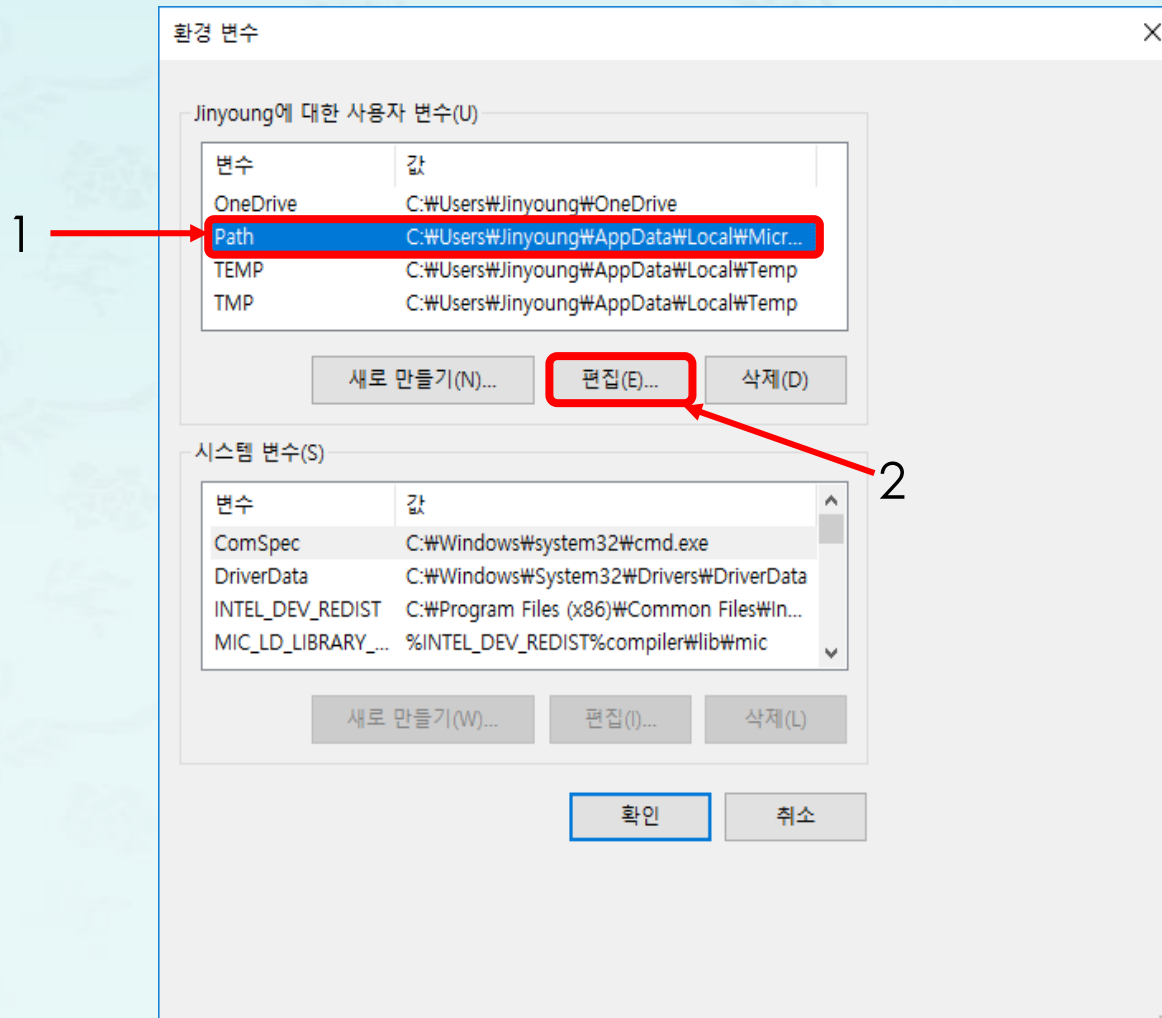
- 다음은 시스템 환경 변수를 설정해야 합니다.
- 환경변수를 설정하기 위해 윈도우 검색에서 “환경 변수”를 입력하고 계정의 환경 변수 편집으로 들어갑니다.



# MSYS2/MinGW-w64 for Windows

## 환경변수 설정

- 사용자 변수 창 혹은 시스템 변수 창에서 Path를 선택한 후 편집을 클릭합니다.





# MSYS2/MinGW-w64 for Windows

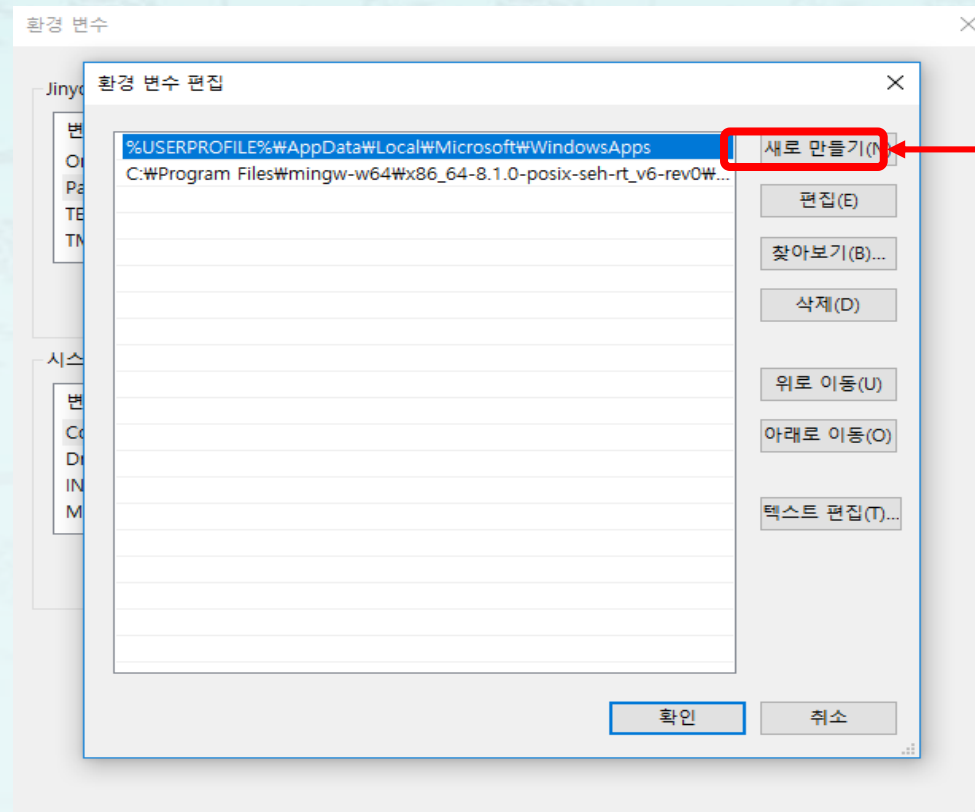
## 환경변수 설정

- [새로 만들기]를 클릭 후 g++.exe 가 존재하는 path를 입력하거나 [붙여넣기] 합니다.
  - msys2와 mingw64를 설치한 곳에서, **g++.exe** 가 존재하는 폴더를 찾아서, 그 **Path**를 복사하여 붙여 넣기를 하는 것이 안전합니다.
  - 만약, msys2 설치 시에 디폴트를 사용했다면, 다음과 같지만, 한번 확인해 보는 것이 좋습니다

**C:\msys64\mingw64\bin**

**C:\msys64\usr\bin**

위의 두번째 폴더는 ls, cp, rm 같은 Bash 명령어를 사용하기 위해 필요한 폴더입니다.

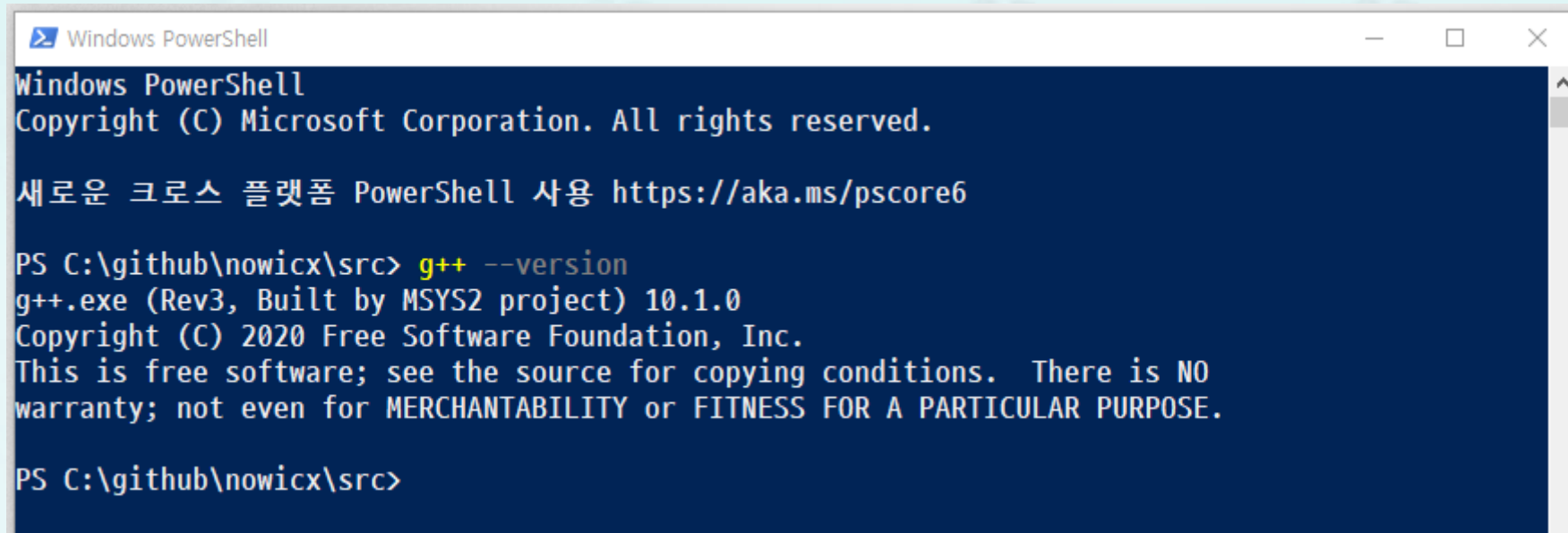


클릭 후 path 입력

# MSYS2/MinGW-w64 for Windows

## 환경변수 설정

- 윈도우 검색창에 “**cmd**” 명령어를 실행하여, 콘솔(console or terminal)을 띄웁니다.
- 프롬프트 창에 “**g++ --version**”을 입력하여 다음과 같이 화면이 나온다면 path추가가 정상적으로 된 것 입니다.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\github\nowicx\src> g++ --version
g++.exe (Rev3, Built by MSYS2 project) 10.1.0
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

PS C:\github\nowicx\src>
```

# MSYS2/MinGW-w64 for Windows

---

## MSYS2/MinGW-w64 설치하기 오류에 관하여...

- MSYS2가 아니라 **MinGW-w64**를 먼저 설치할 경우, 설치 오류가 발행하는 경우가 종종 있습니다.
- 또한 한글로 사용자 이름이 설정되어 있는 경우, 어려움을 많이 겪습니다. 이런 경우, 특히 블로그의 설명에 따르면 재설치하는 경우 문제가 발생한다고 합니다.
  - 이런 경우, 기존의 있는 폴더를 완전히 삭제한 후, MSYS2로 다시 설치해보십시오.
  - **MSYS2를 먼저 설치하고**, 그 안에서 MinGW-w64를 추가 설치하는 방법을 택하길 적극 권장합니다.
  - 그래도, 설치에 어려움이 있다면, (특히 한글 사용자 이름으로 인하여), 새로운 사용자를 만들어 시도해보십시오.
- <https://taeguu.tistory.com/3>

# MSYS2/MinGW-w64/Atom 설치 테스트

- Use this code to test that g++ can compile and run the program.

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello, World!\n";
    return 0;
}
```

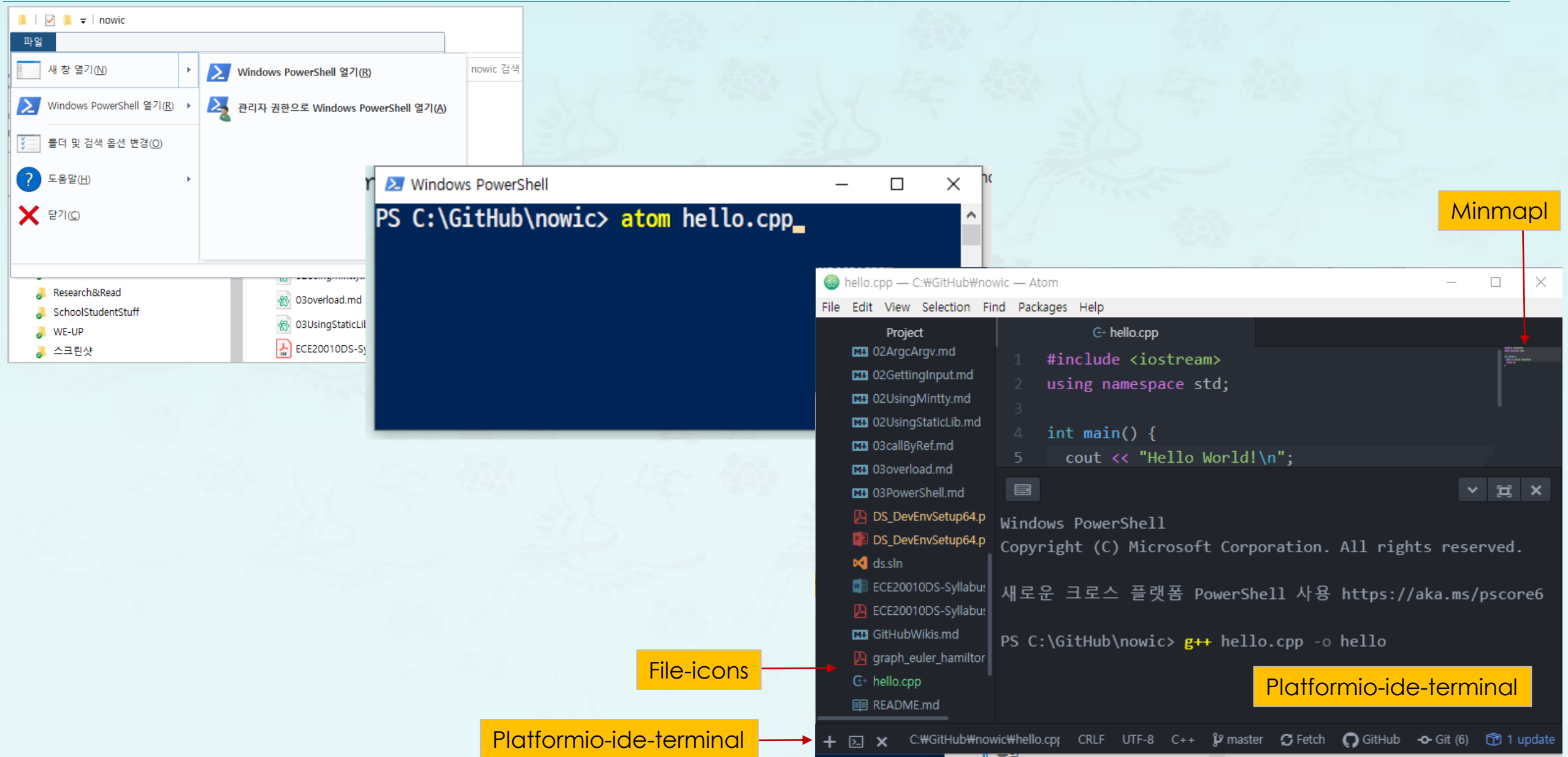
1. Copy the above program and save the file as **hello.cpp** on your folder.
2. Open a terminal at the folder where **hello.cpp** exists.  
Use the [File] → [Windows PowerShell 열기] menu at the top left corner of the **File Explorer**.
3. To compile, enter the following in the terminal and run it. We see the file **hello** appear on the desktop under an exec icon.

```
$ g++ hello.cpp -o hello
```

4. Run the compiled program by typing the following in the terminal. You will see "Hello World!" printed in the terminal.

```
$ ./hello
```

# MSYS2/MinGW-w64/Atom 설치 테스트



# 데이터 구조 개발환경

ECE20010

한동 대학교

김영섭 교수

idebtor@gmail.com

GCC  
Atom