

LESSON

파이썬(Python)

강사 : 김은숙

파이썬(Python)

1991년 네델란드의 수학자 귀도 반 로섬 (Guido van Rossum)이 C언어로 제작한 언어.

파이썬의 로고는 비단뱀 2마리가 꼬여 있는 모습을 형상화한 것.

그리스어 '피톤(python)'에서 유래했으며 '비단뱀'이라는 뜻을 가지고 있다.



Guido van Rossum



파이썬은 C로 만든 CPython, 자바로 만든 JPython, C#으로 만든 IronPython 등이 있다. 우리가 일반적으로 사용하는 파이썬은 CPython이다.

파이썬(Python)의 특징

인터프리터 언어(Interpreter Language)

인터프리터에 의해 한 줄씩 읽어 바로 실행 결과를 확인할 수 있는 언어이다.

동적(Dynamic) 타입 언어 예제로 이동 ▶

변수의 자료형을 지정하지 않고 선언만으로 값을 지정할 수 있다.

플랫폼에 독립적인 언어

운영체제(윈도우, 리눅스, 맥 등)에 관계없이 같은 코드를 사용할 수 있다.



플랫폼이란?

프로그램이 돌아가는 운영체제(OS: Operating System)환경이나 개발환경을 말한다.

파이썬(Python)의 장단점

장점

배우기 쉽다.

읽고 쓰기 쉬워서 유지 보수가 쉽고 개발 속도가 빠르다.

무료로 제공되는 수많은 라이브러리가 있다.

단점

컴파일 언어에 비해 속도가 느리다.



라이브러리(Library)?
어떠한 작업을 수행하는 로직을 재사용 가능
하도록 모듈화 시켜서 저장한 프로그램.

파이썬으로 할 수 있는 것들

- 웹 데이터 수집
- 웹 애플리케이션 개발

파이썬으로 제작된 응용프로그램들>>>드롭박스, 인스타그램, 핀터레스트...

- 데이터 분석
- 데이터 시각화
- 인공지능과 머신 러닝
- 업무 자동화
- 게임 개발

LESSON

파이썬 설치

파이썬 인터프리터 설치

파이썬 언어로 작성된 소스

```
import time
st=time.time()
li=[]
for i in range(1000000):
    li.append(i)

et=time.time()
print('for문:',et-st)
```

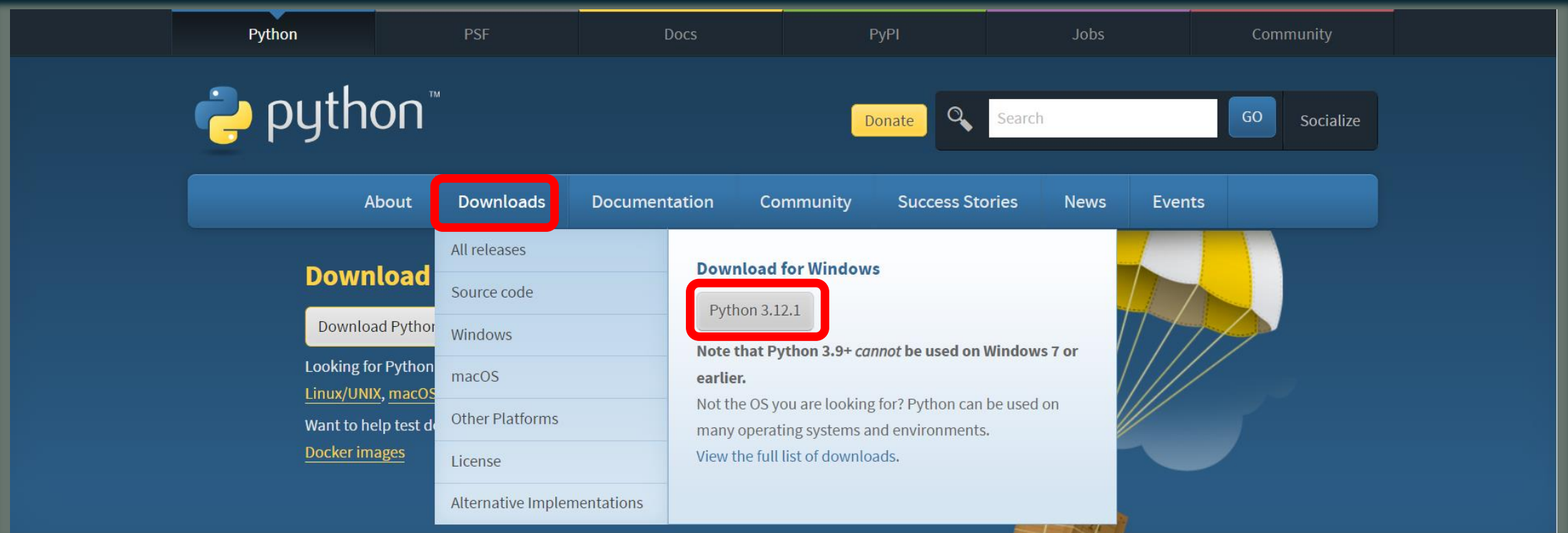
인터프리터

기계어

```
1000 1100 1010 0011 1111
0000 1010 1110 0001 0001
0000 1010 1110 0001 0001
1000 1100 1010 0011 1111
0000 1010 1110 0001 0001
0000 1010 1110 0001 0001
1000 1100 1010 0011 1111
```

파이썬 설치하기

1. <https://www.python.org> 에 접속한다.
- 2 [Downloads] 메뉴에 마우스를 올리면 사용중인 운영체제에 적합한 다운로드 버튼이 나타난다.



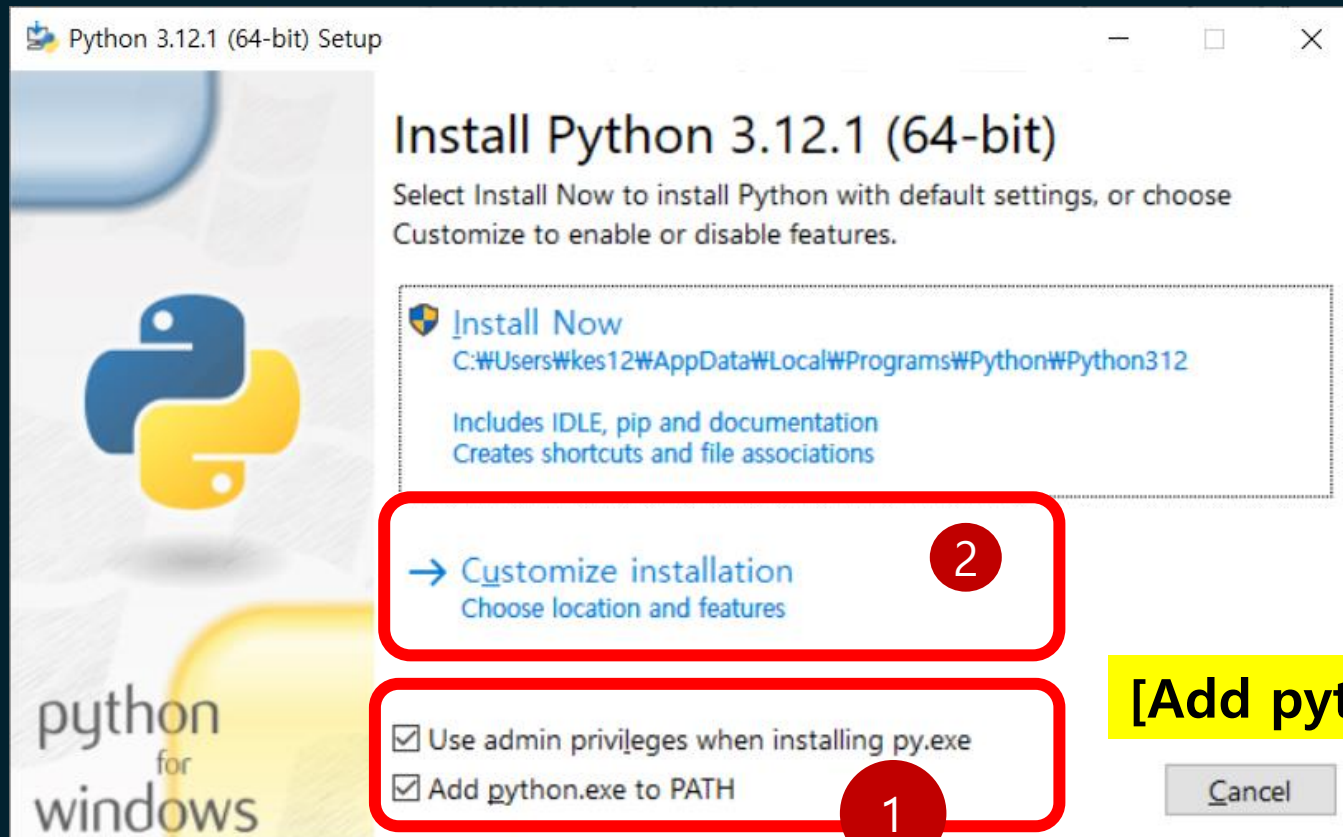
파이썬 설치하기

3. 다운로드 받은 파일을 실행한다.

설치 첫 단계에서 혹시 [이 앱이 디바이스를 변경할 수 있도록 허용하시겠습니까?] 라는 대화 상자가 나타난다면 [예]를 선택한다.

파이썬 설치하기

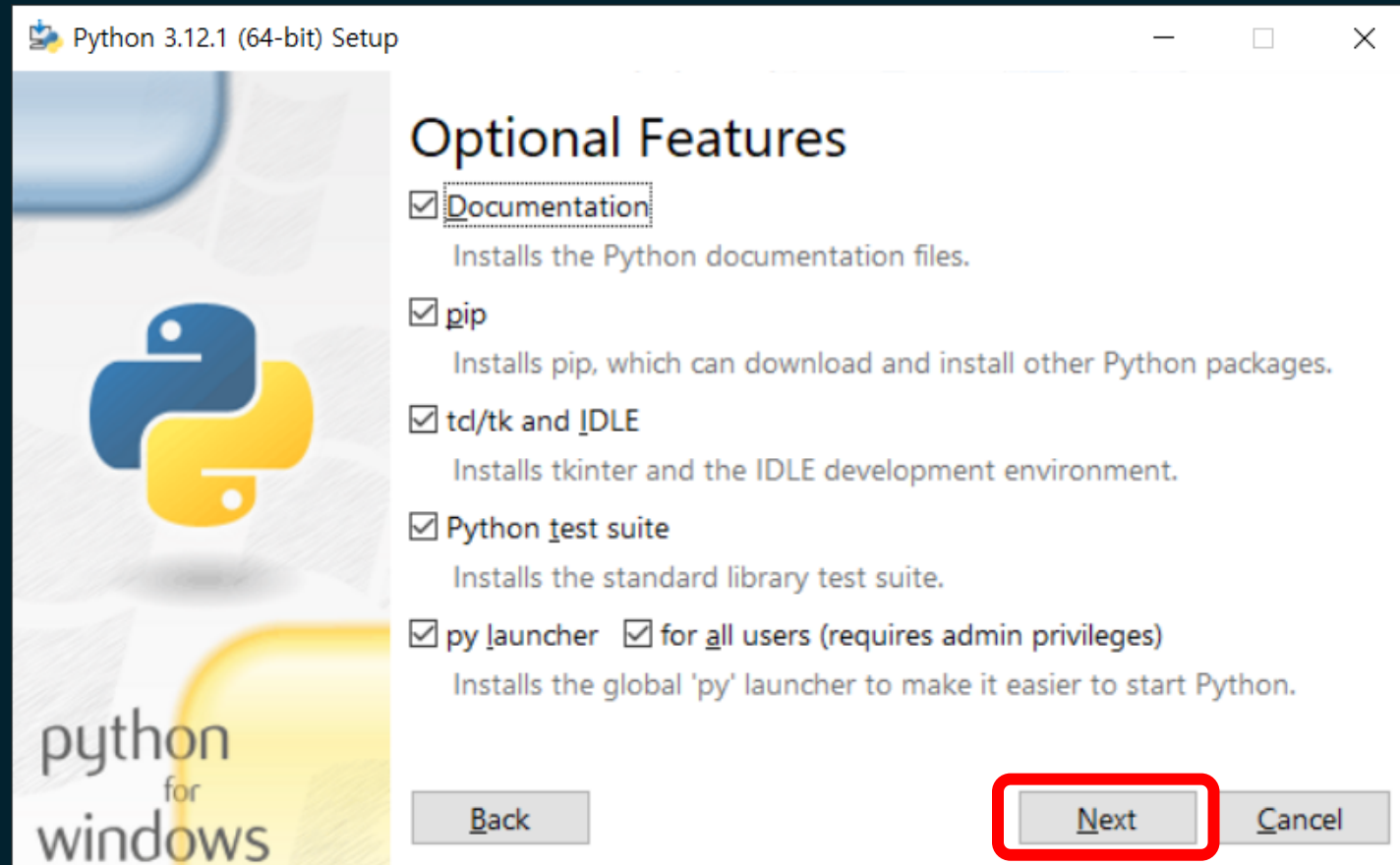
4. 다운로드 받은 파일을 실행하고 다음 화면과 같이 선택한 후 [**Customize installation**] 클릭한다.



[Add python.exe to PATH]를 반드시 선택할 것

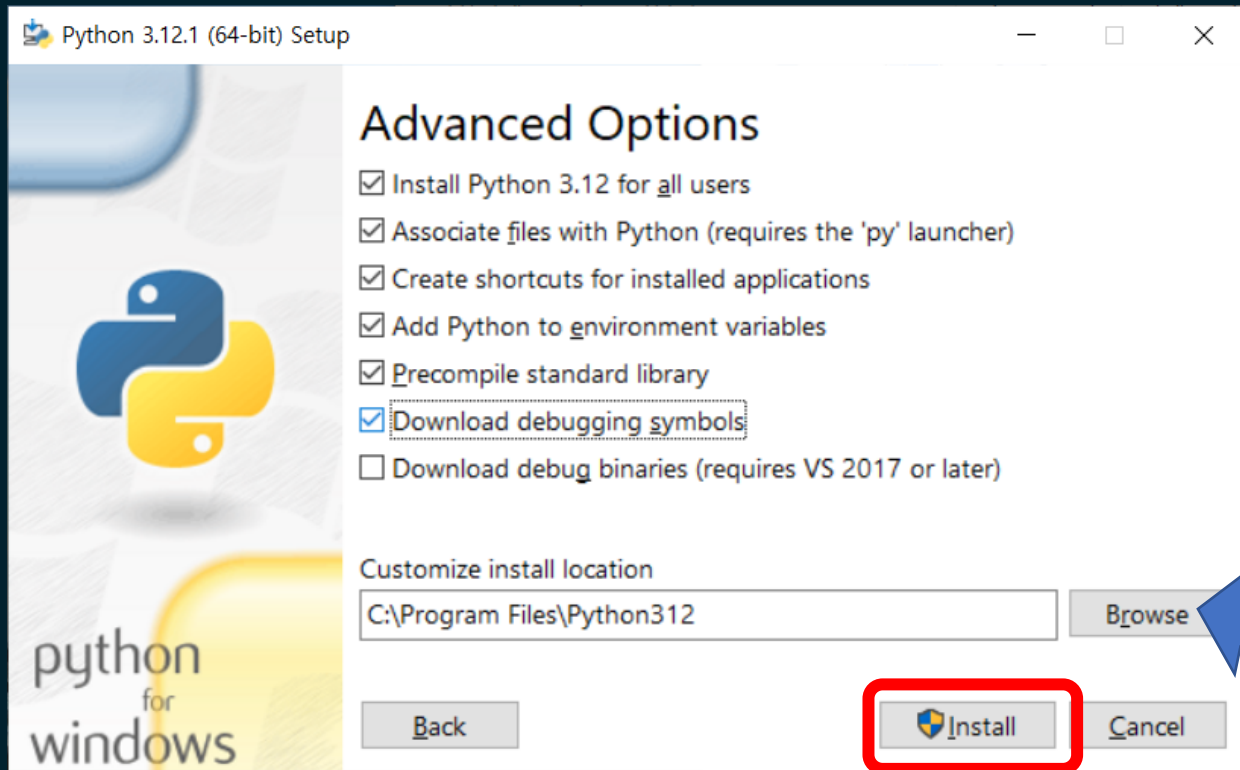
파이썬 설치하기

5. 다음 화면과 같이 모두 선택하고 [Next]를 클릭한다.



파이썬 설치하기

6. 다음 화면과 같이 선택하고 적절한 설치 경로를 설정한 후 [Install]을 클릭하면 설치가 진행된다.

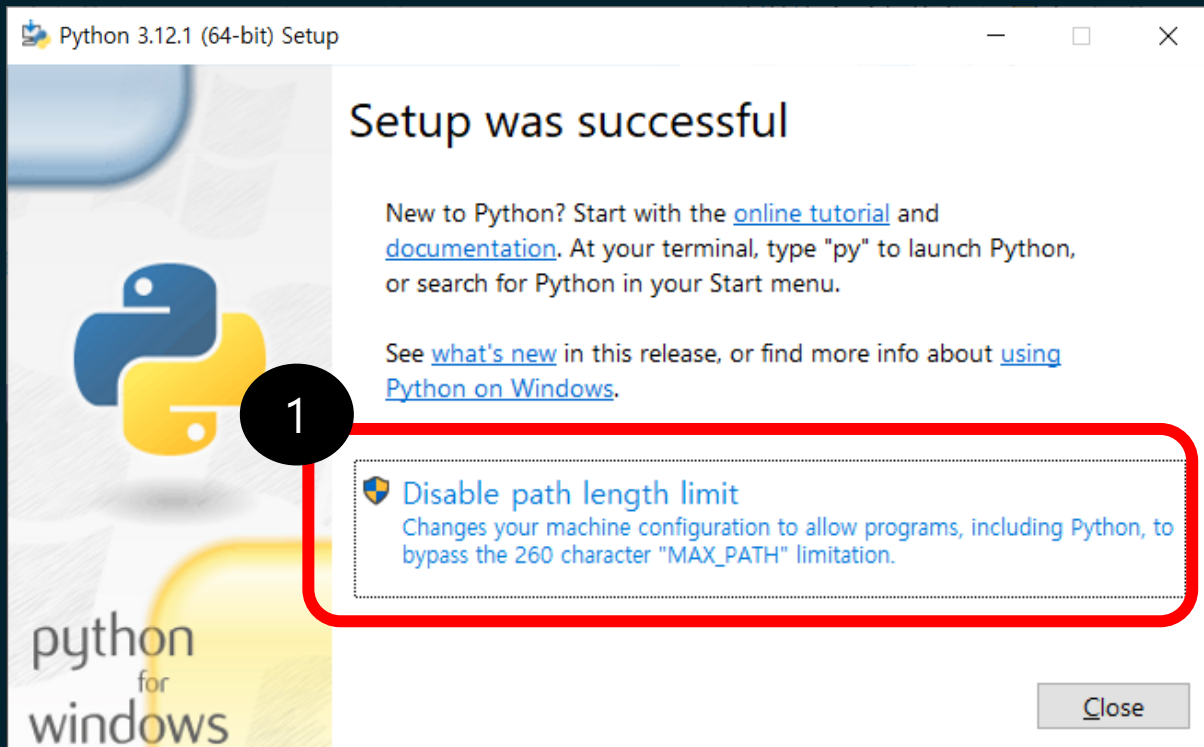


기본 설치되는 폴더가 표시된다.

설치를 원하는 폴더를 지정하려면 "Browse"를 클릭해서 설치할 폴더를 지정하면 된다.

파이썬 설치하기

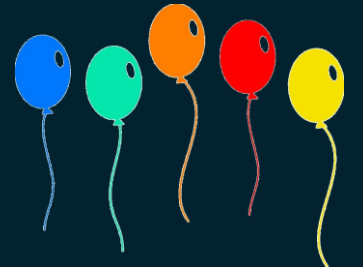
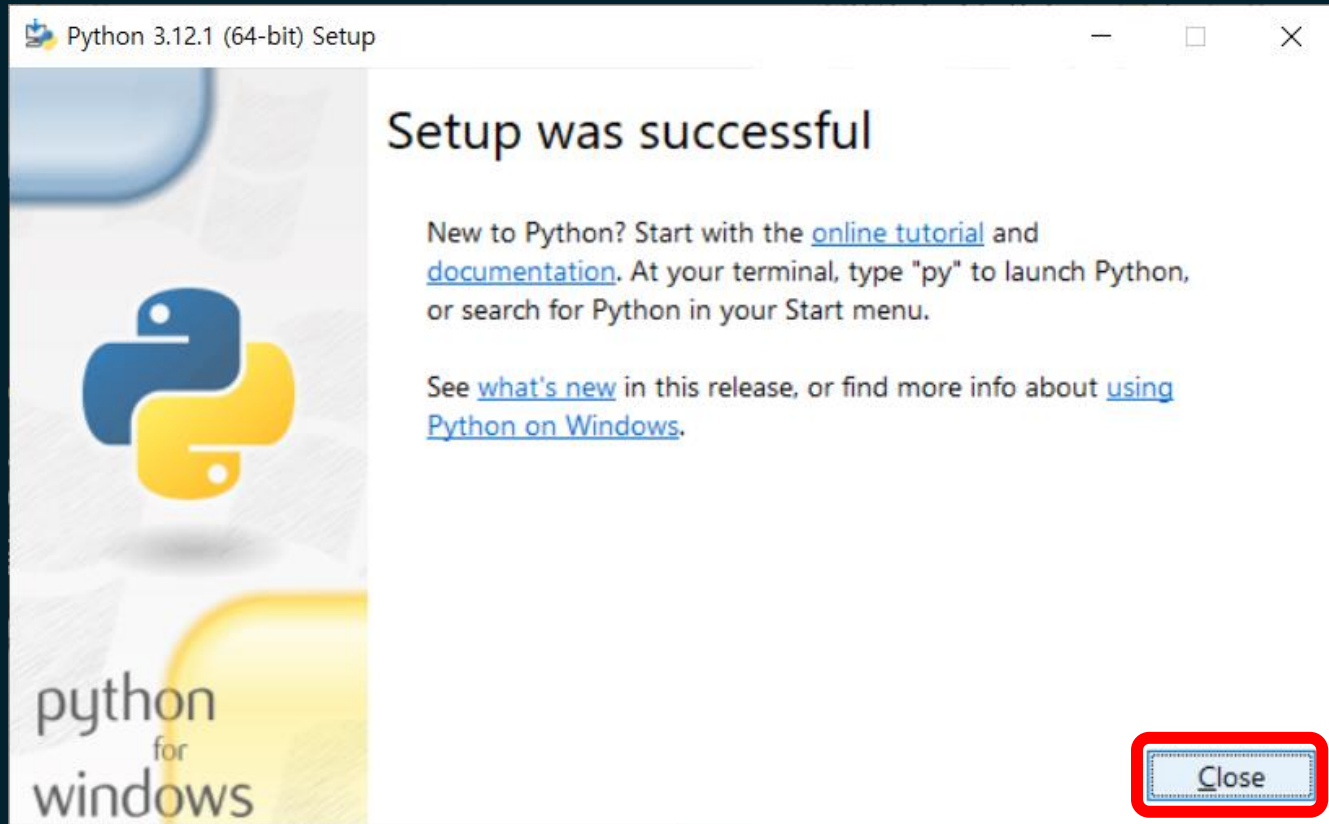
7. 다음 단계에서는 반드시 [Disable path length limit]를 클릭한다.
Windows는 파일 이름과 경로의 길이가 260자로 제한되어 있다. 260자를 넘더라도 문제가 발생하지않도록 길이 제한을 해제하는 것이다.



(참고) 컴퓨터에 파이썬 설치한적이 있는 경우 이 화면은 나오지 않음

파이썬 설치하기

8. 다음 화면이 나타나면 [Close]를 클릭하여 설치를 완료한다.



축하합니다!!!

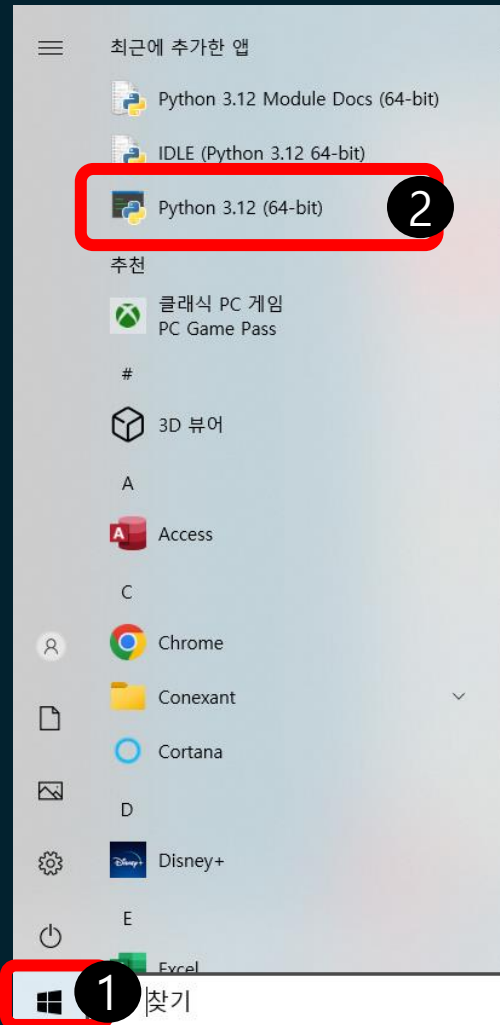
LESSON

파이썬 실행하기

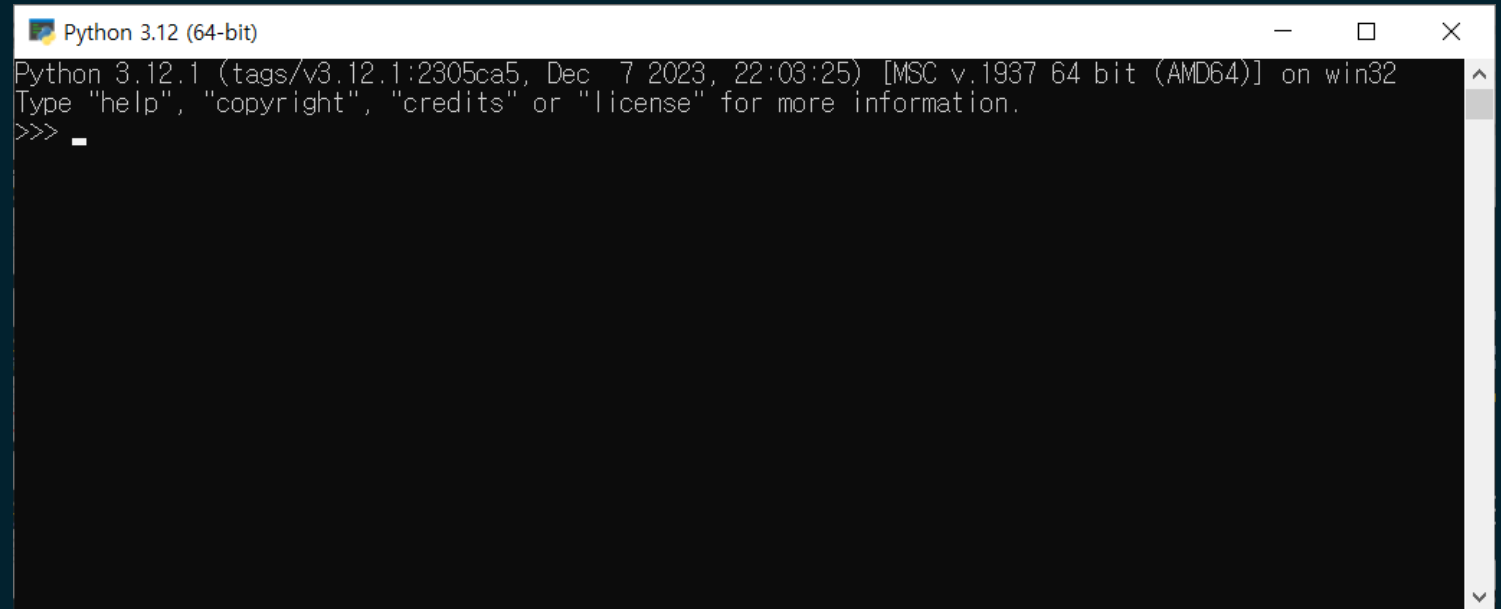
3가지 방법

파이썬을 실행하는 방법1

1. [시작]버튼을 누르면 방금 설치했던 Python 3.xx이 표시된다.

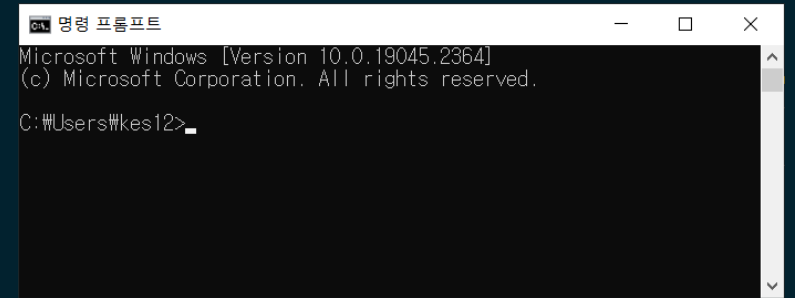
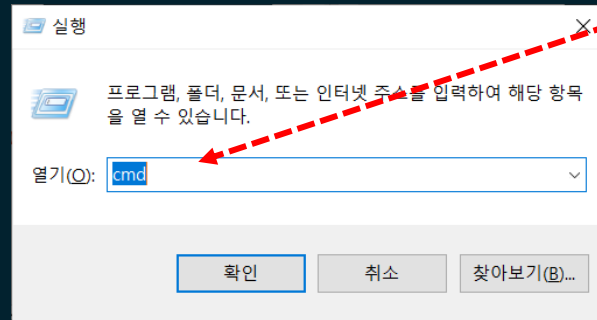


클릭하면 파이썬 인터프리터(셸)가 아래와 같이 나타난다.

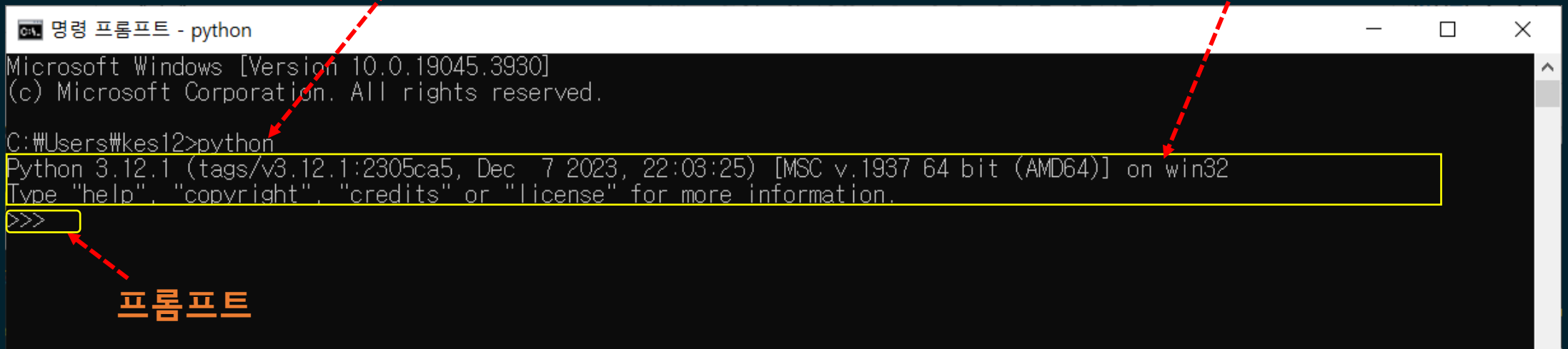


파이썬 실행하는 방법2

1. 키보드에서  + R 을 눌러 [실행]창이 나타나면, 'cmd'를 입력하고 엔터키를 누르면 명령 프롬프트 창이 표시 된다.



2. 명령 프롬프트에 "python"을 입력하고 엔터를 누르면 아래와 같이 파이썬 셸이 표시됨.



파이썬 실행하는 방법3

통합개발환경(IDE)에서 실행

컴퓨터로 문서를 작성하려면
메모장?
워드?

LESSON

통합개발환경(IDE)

Integrated Development Environment

효율적으로 소프트웨어를 개발하기 위한 프로그램.

“소스 코드 작성, 디버깅(debugging), 실행”



LESSON

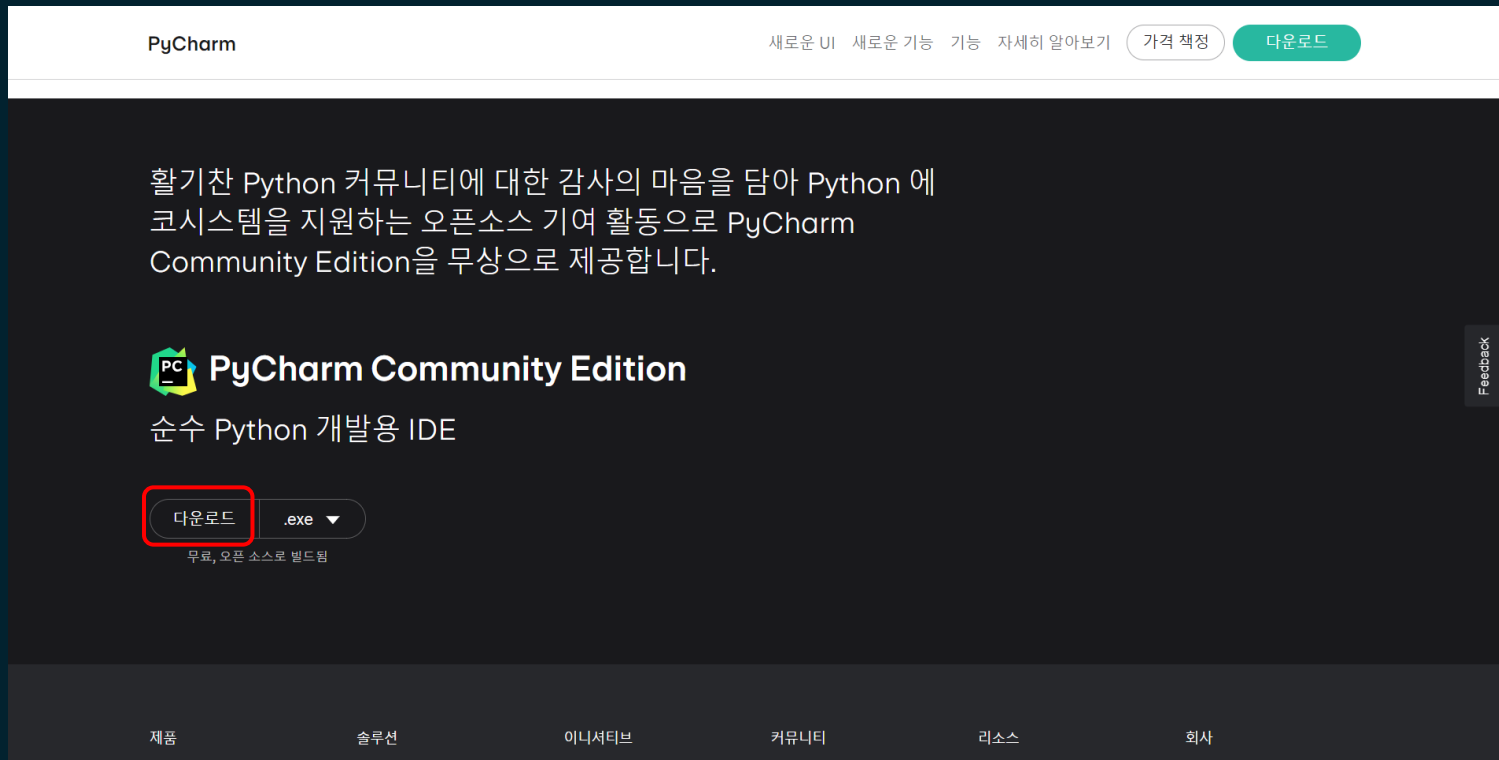
파이참(PyCharm) 설치



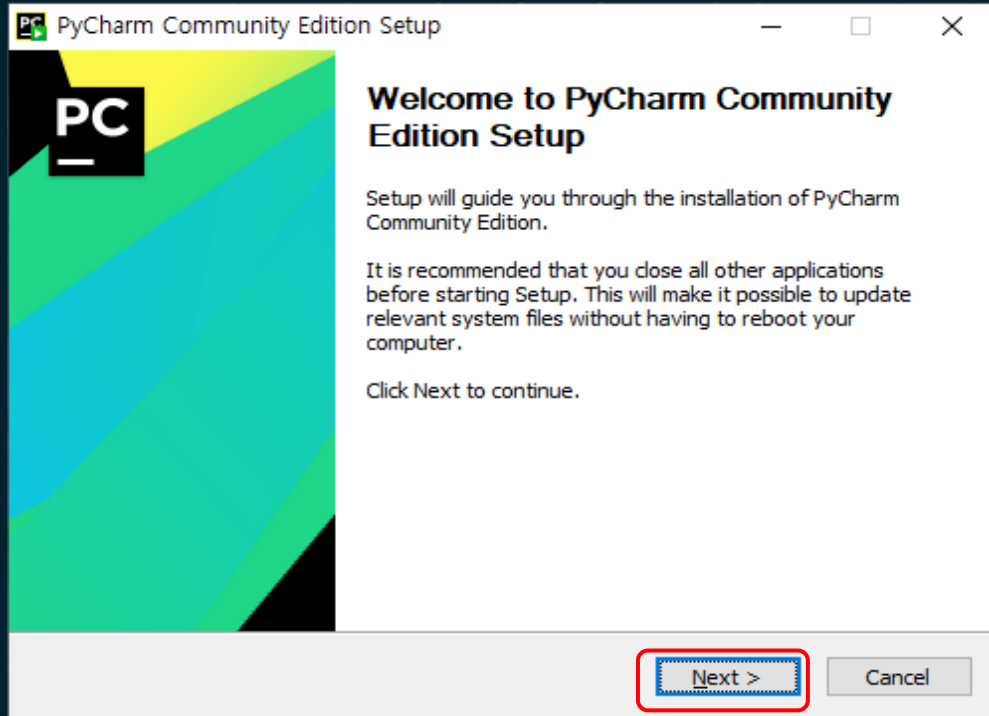
1. 아래 주소에 접속한다.

<https://www.jetbrains.com/pycharm/>

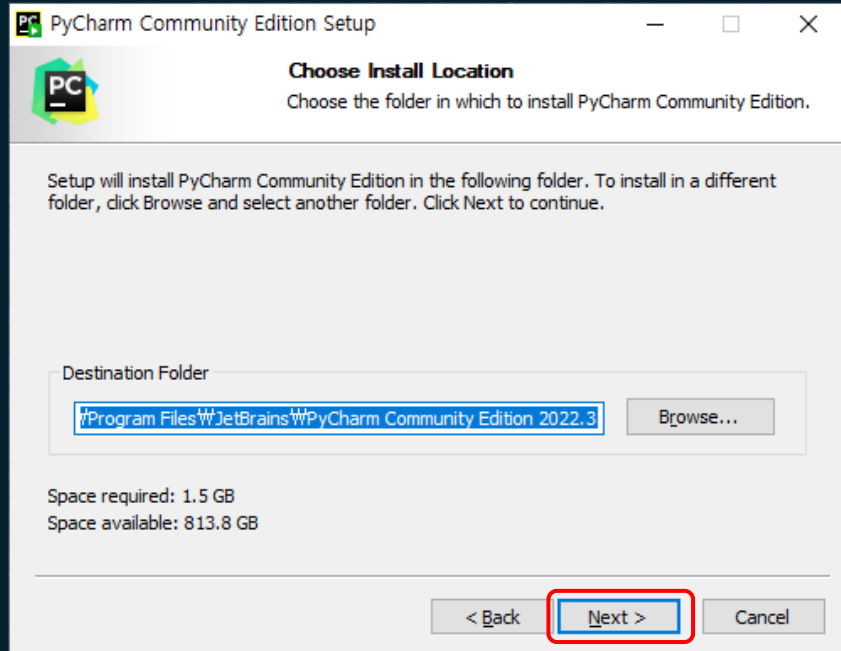
2. 화면을 아래로 스크롤해서 [Pycharm Community Edition] 부분의 [Download] 버튼을 클릭하여 설치 파일을 다운로드한다.



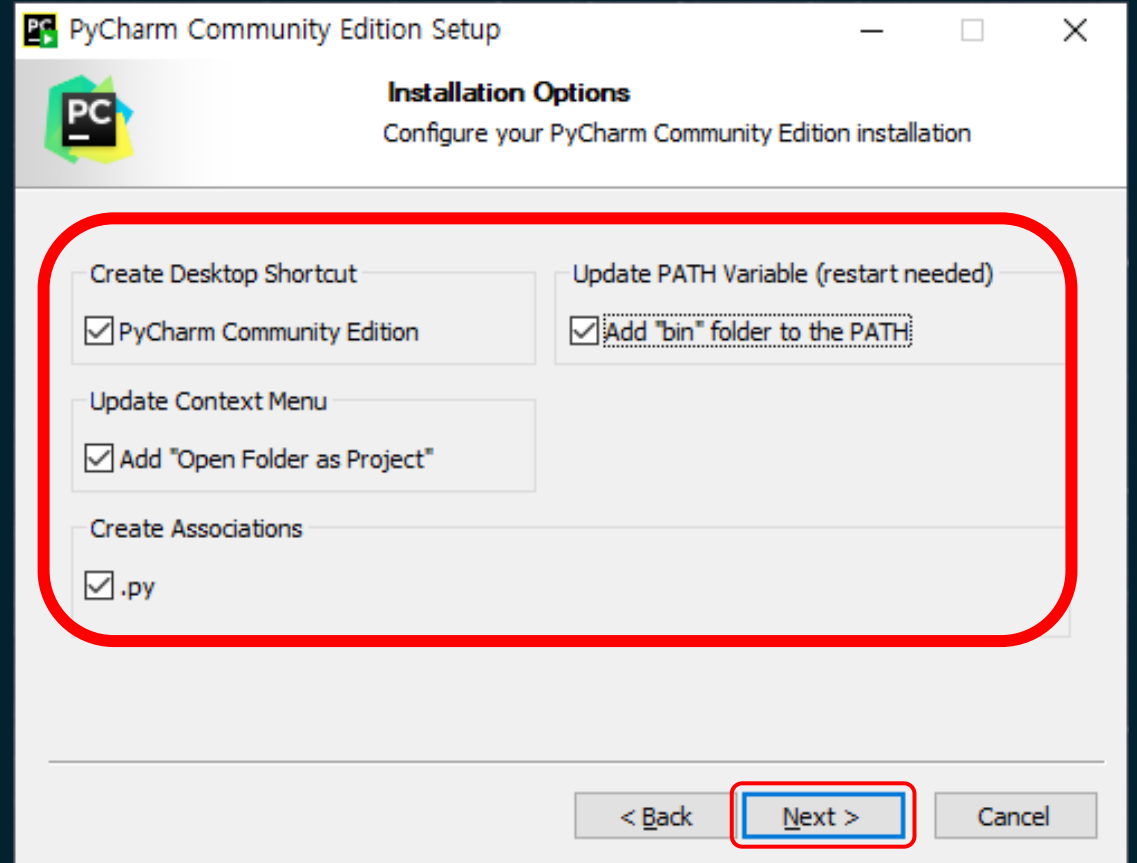
3. 다운로드 받은 파일을 더블클릭해서 실행하고 아래 화면에서 [Next]를 클릭한다.
※ 설치 첫 단계에서 [이 앱이 디바이스를 변경할 수 있도록 허용하시겠습니까?] 라는 대화 상자가 나타나면 [예]를 선택한다.



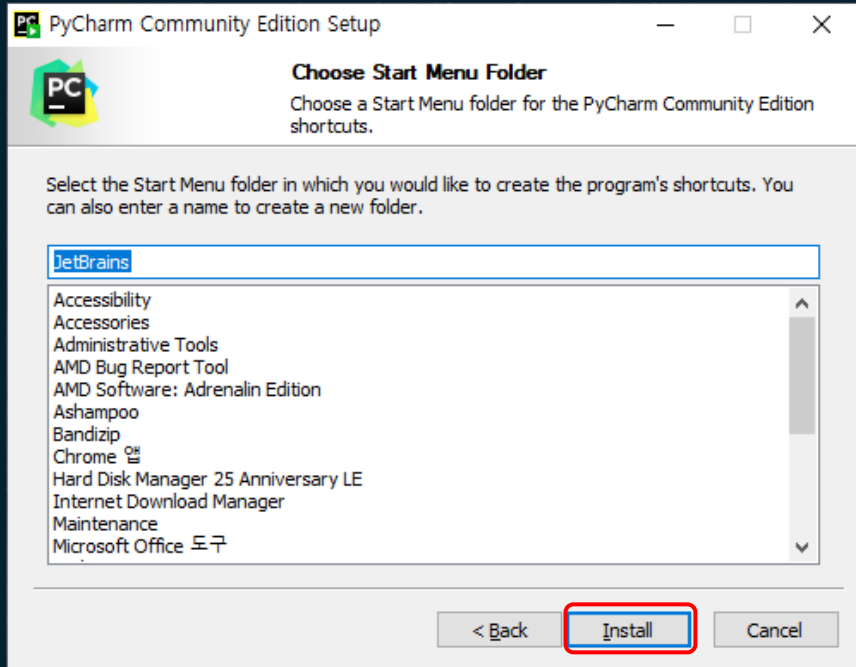
4. 파이참이 설치될 경로를 지정하는 화면에서 [Next]를 클릭한다. 원하는 위치를 지정해도 된다.



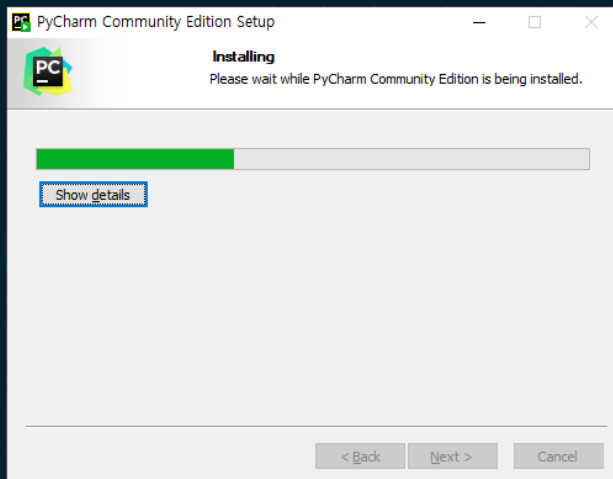
5. 다음 단계에는 화면과 같이 **모든 항목을 선택**하고 [Next]를 클릭한다.



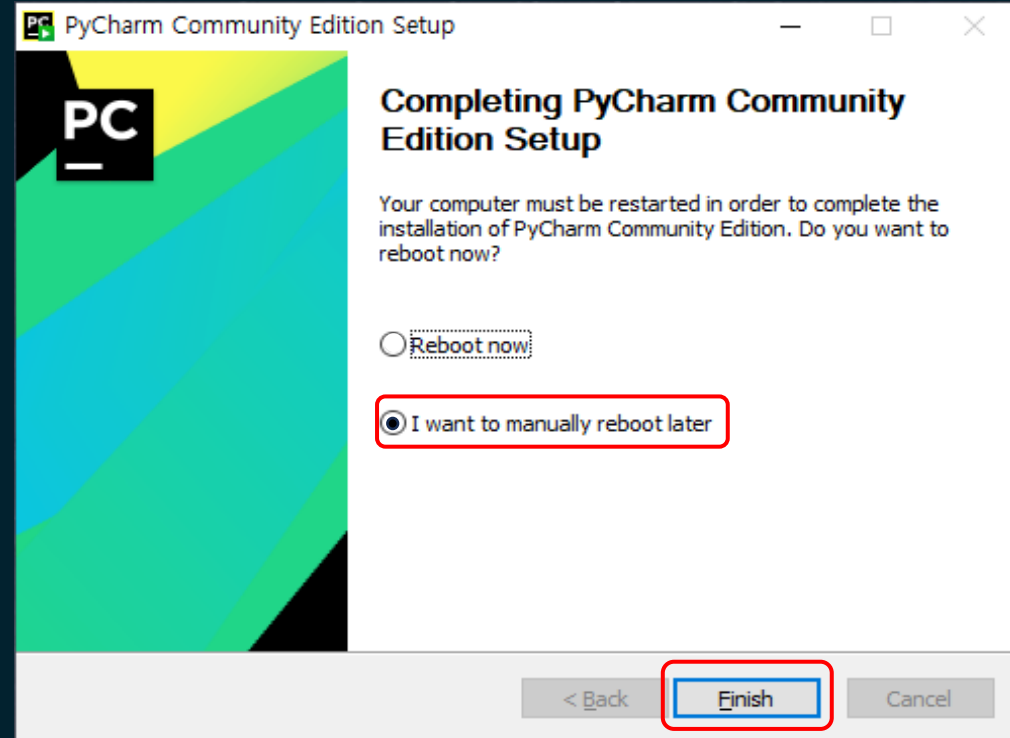
6. 다음 화면에서 [Install]을 클릭한다.



설치가 진행된다.



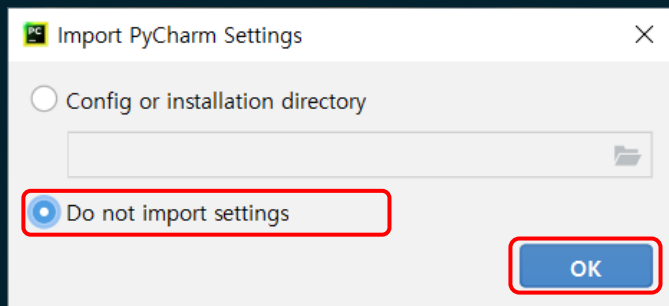
7. 다음 화면에서 반드시 [I want to manually reboot later]를 선택하고 [Finish]를 클릭한다.



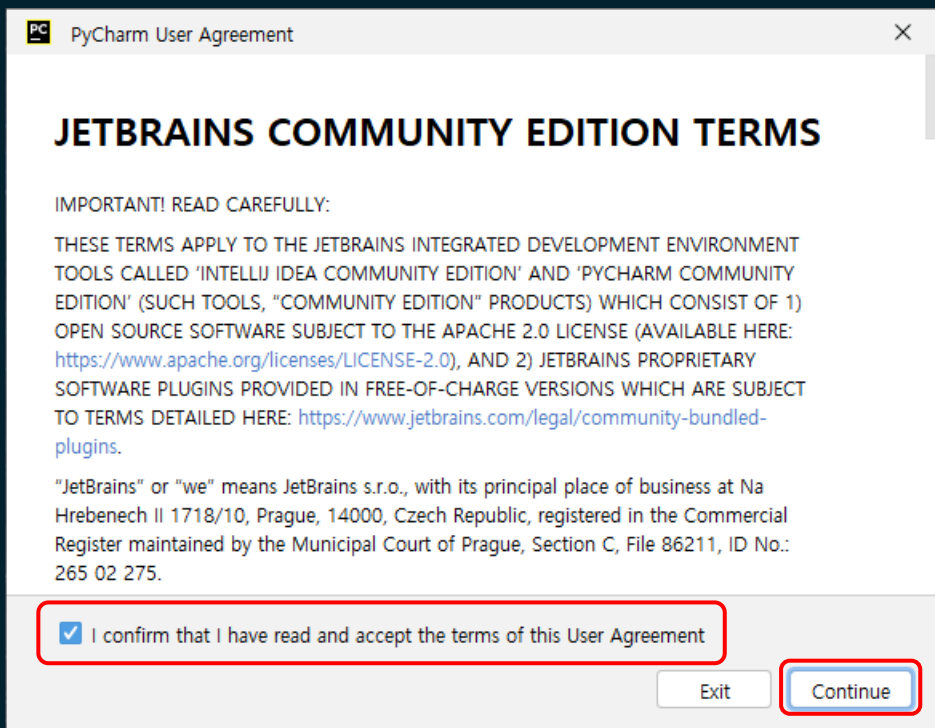
Reboot now를 선택하는 것을 추천한다.

9. 바탕화면에 [PyCharm] 아이콘이 나타난다.

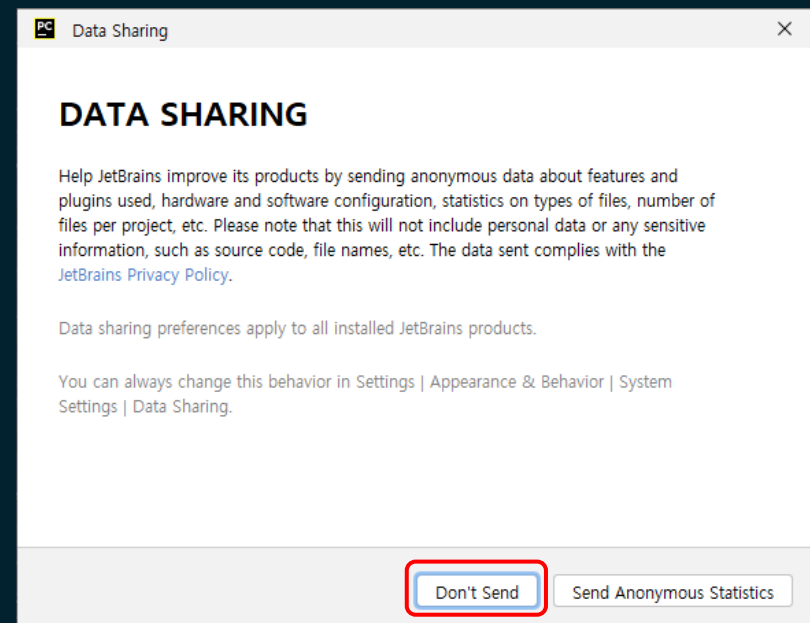
PyCharm을 실행했을 때 아래 화면이 표시되면 "Do not import settings"를 선택하고 "OK"클릭.



10. 첫 화면에서 [동의]를 선택하고 [Continue]를 클릭한다.



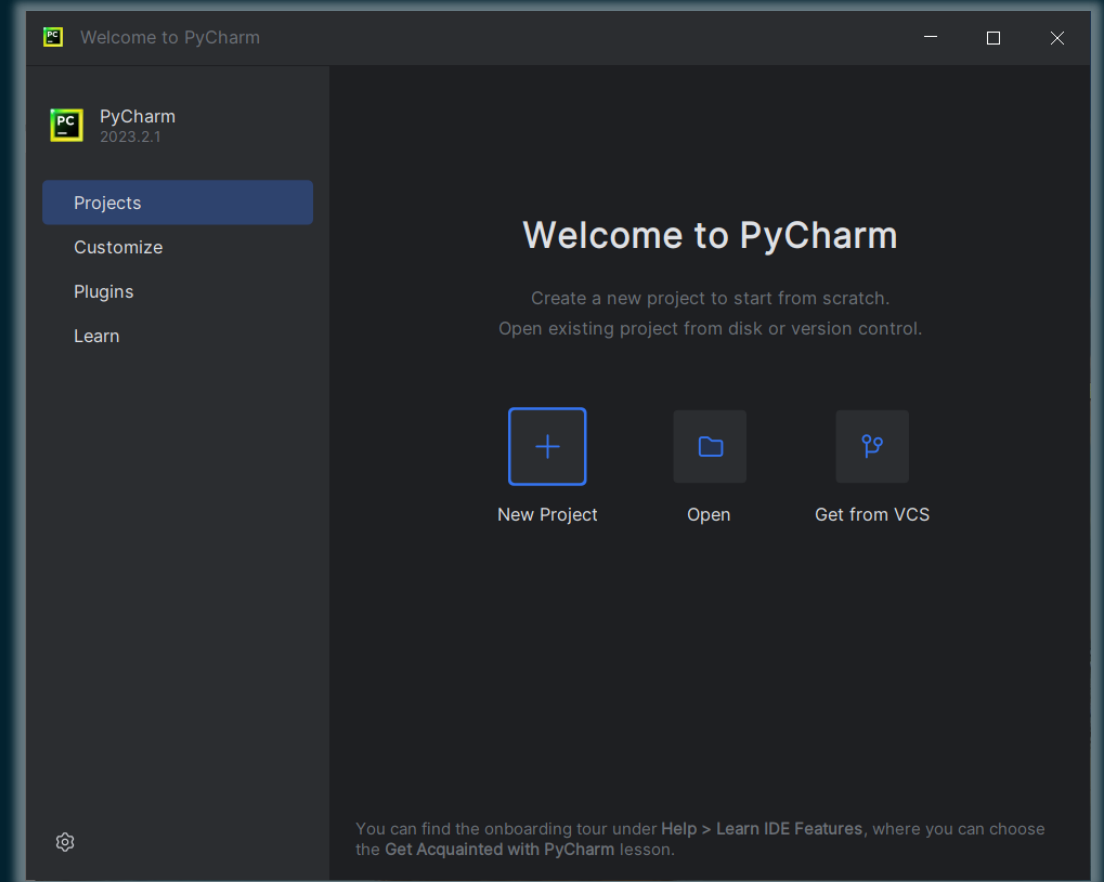
다음 단계에서는
사용자의 데이터를
수집한다는
것이므로 [Don't
Send]를 클릭한다.



아래와 같은 파이참 시작 화면이 나타난다.



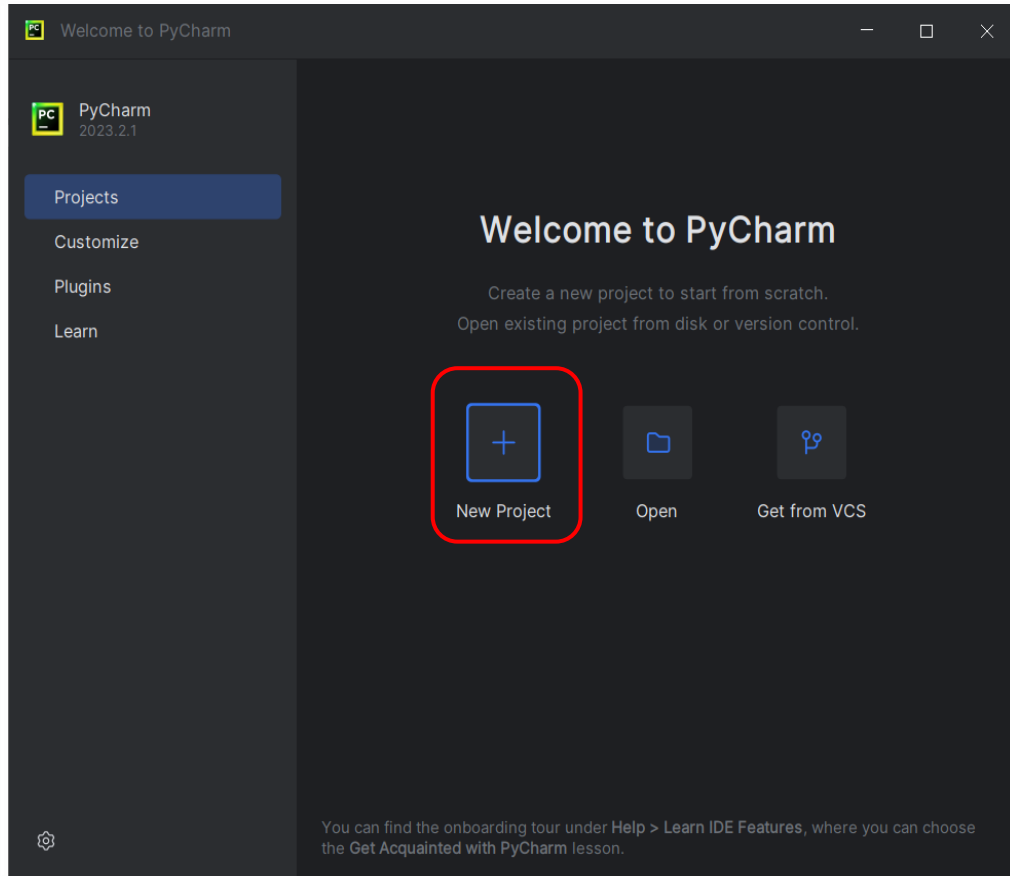
아래 화면이 나타나면 PyCharm이 정상적으로 시작된 것이다.



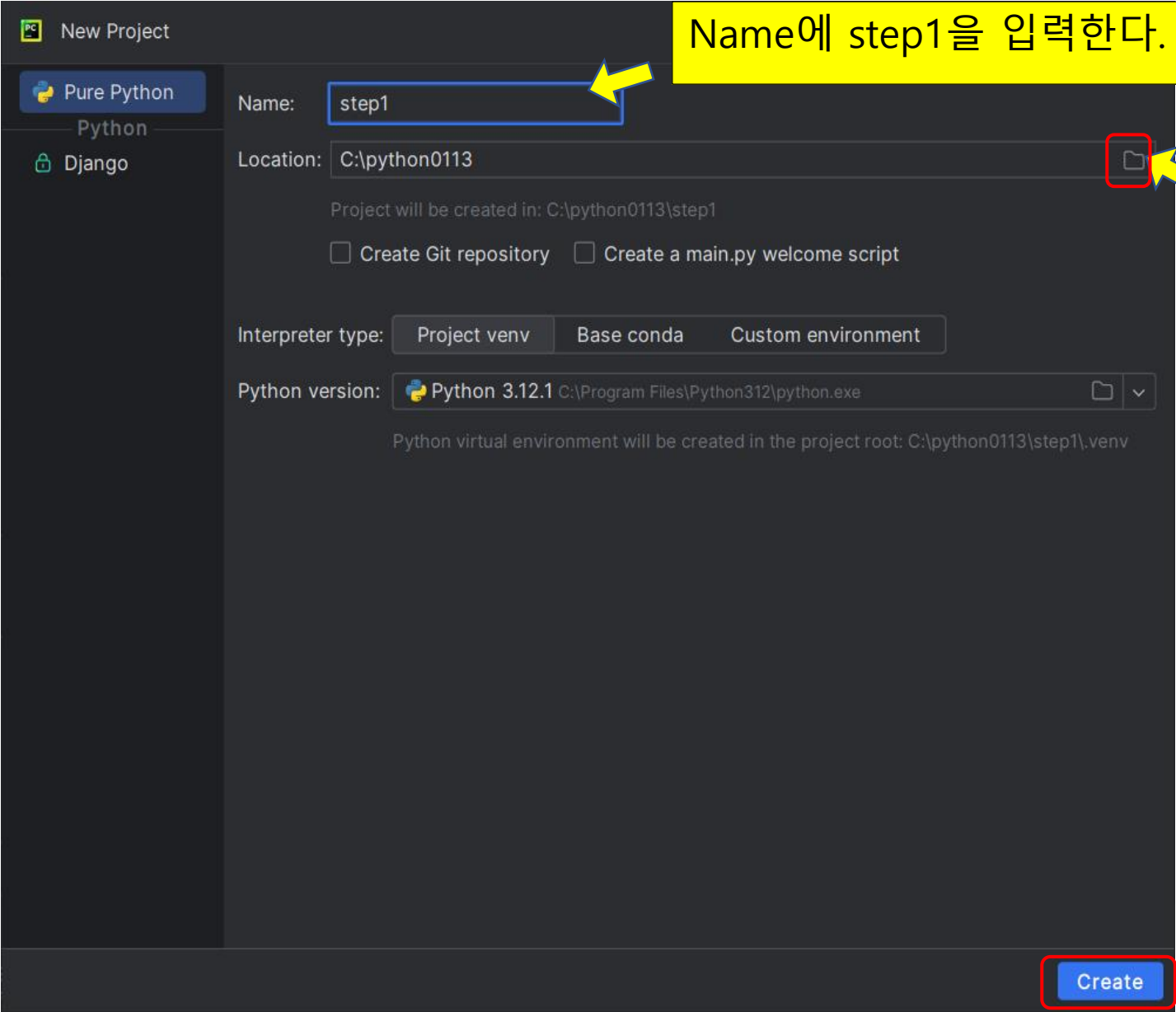
LESSON

프로젝트 만들기

1. [New Project]를 클릭한다.



2. Location과 Name을 다음과 같이 설정하고 Create를 클릭한다..



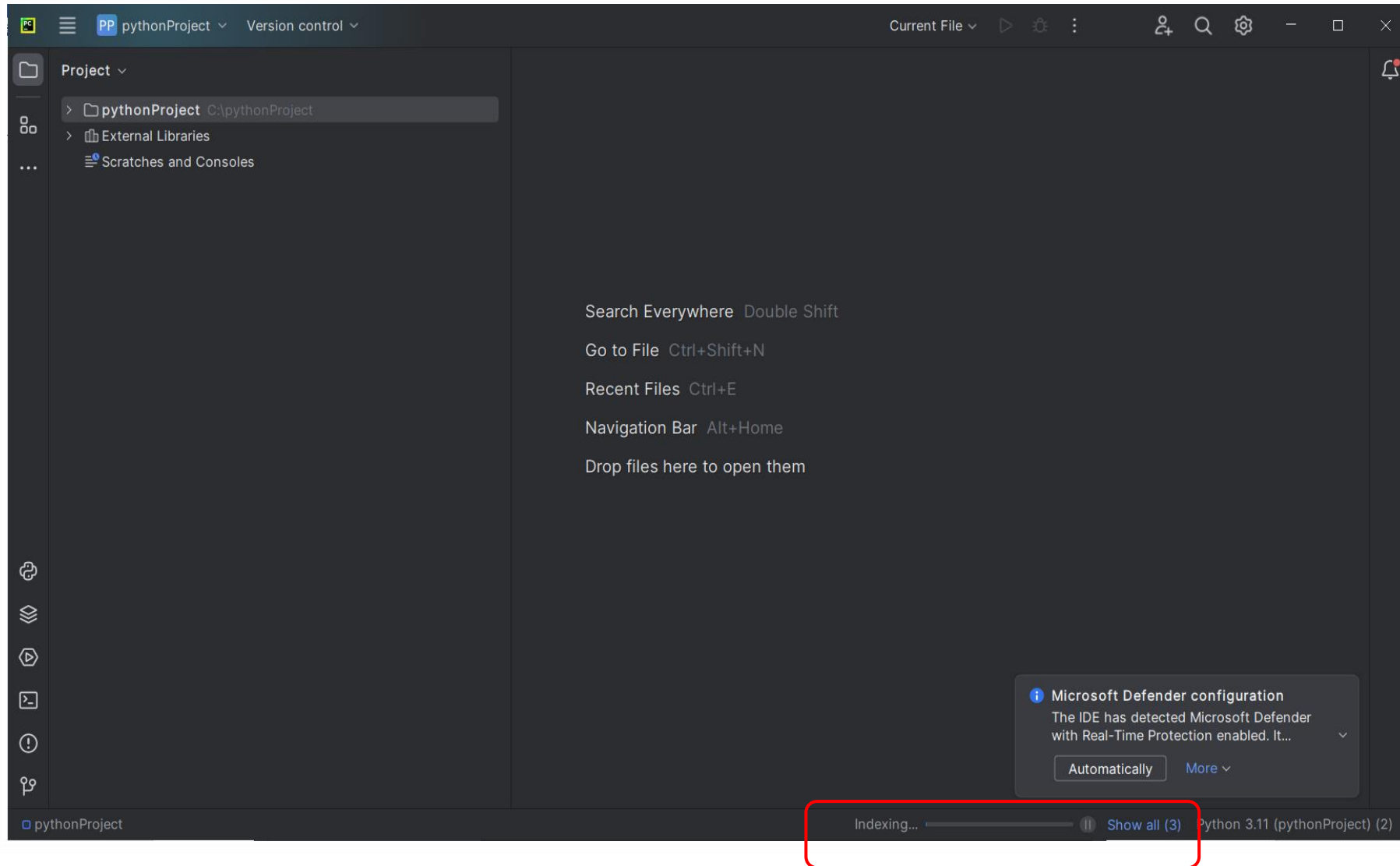
여기를 클릭하고 C:\w아래에 폴더를 만든다.

폴더명: python0113



폴더명, 프로젝트명은 반드시 영어로 작성한다. 한글은 문제가 발생하는 경우가 있다.

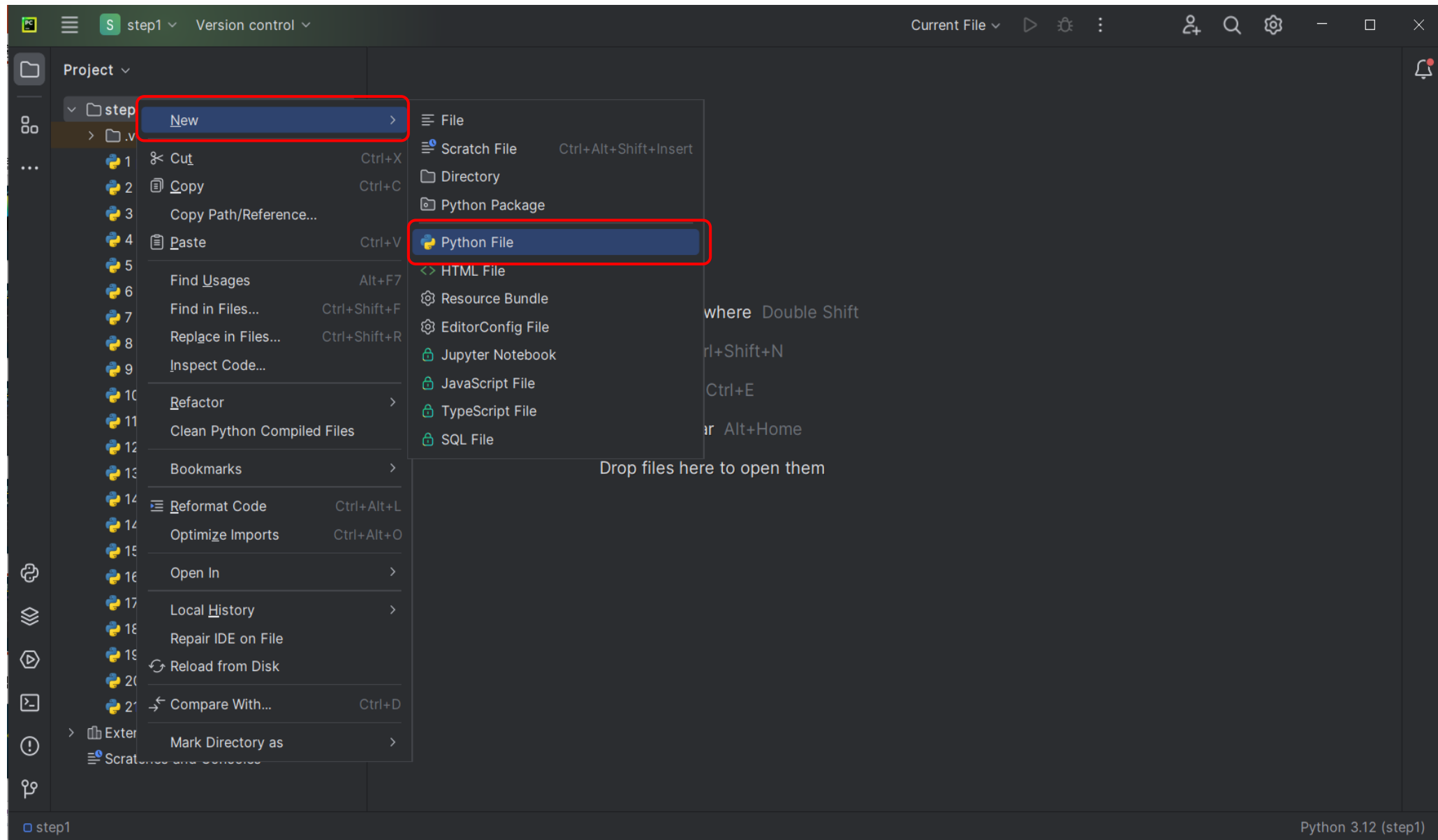
PyCharm은 시작할 때 필요한 정보를 화면 우측 하단에 표시한다. 단, 뭔가 설치나 로드가 진행 중인 경우에는 모두 완료될 때까지 기다려야 한다.



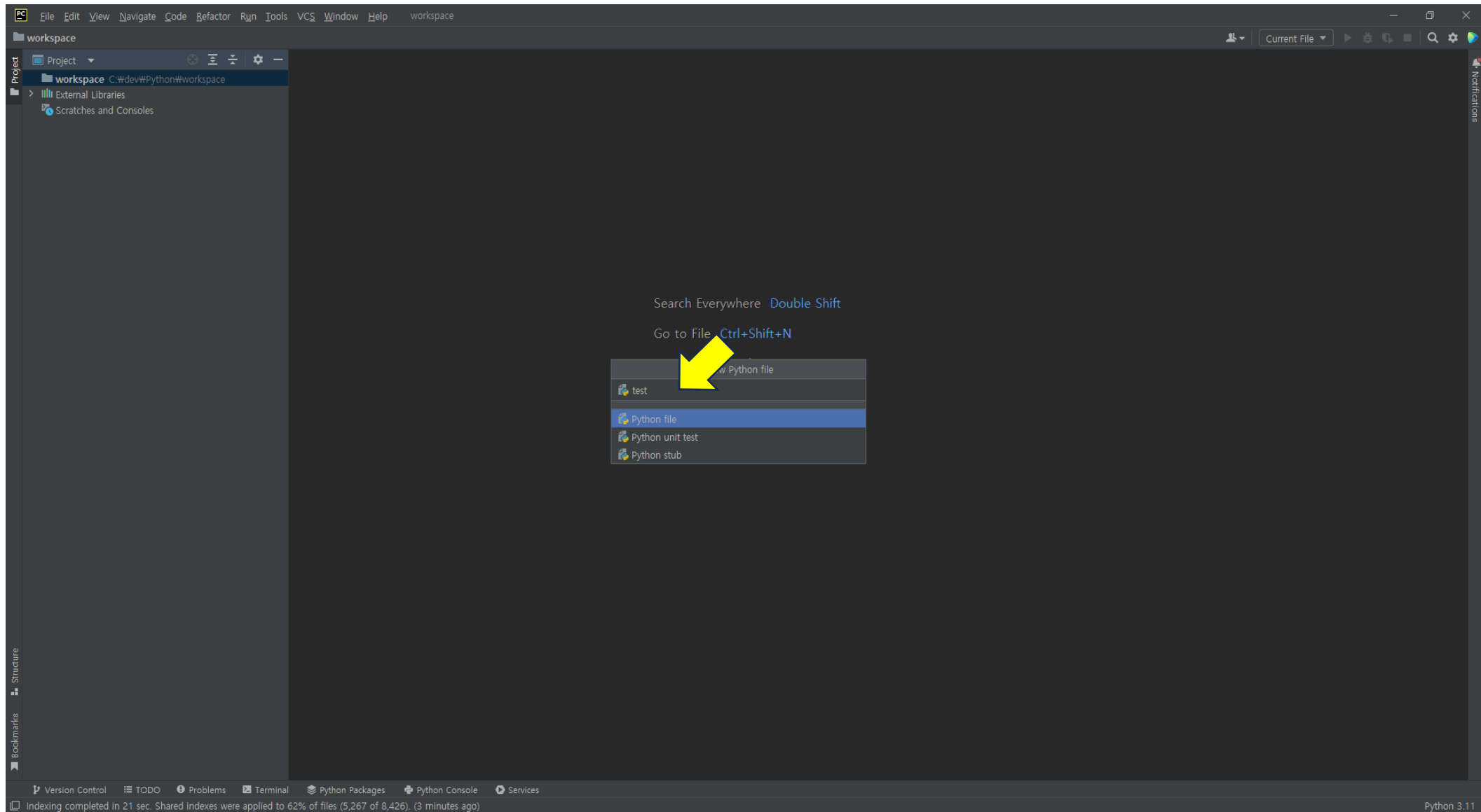
LESSON

파일 만들기

1. 화면 좌측의 [Project] 탭의 "**step1**"에서 마우스 오른쪽을 클릭하고 [New – Python File]을 선택한다.



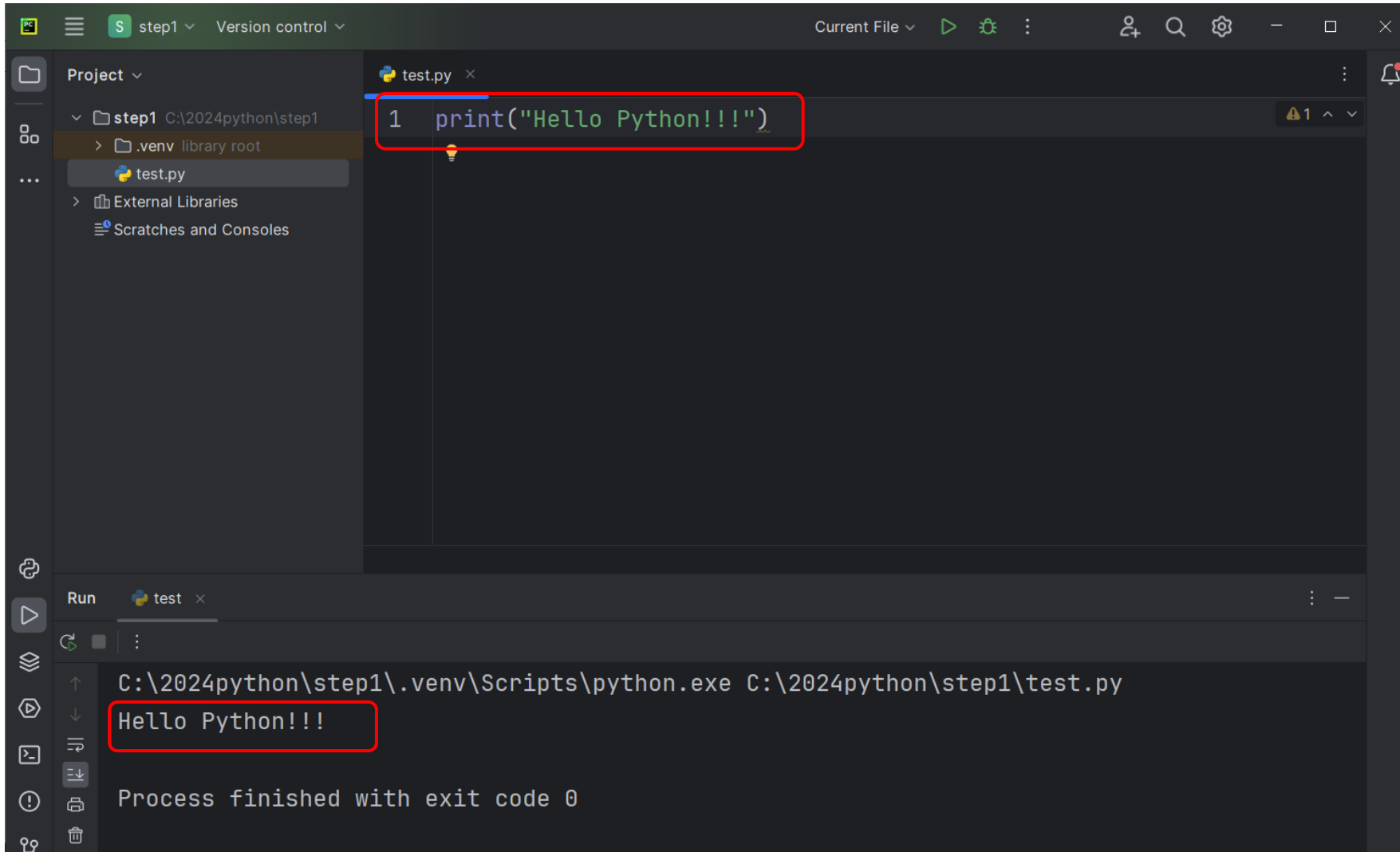
2. 다음 단계에서 파일의 이름을 입력하고 [Enter]를 누른다.



3. 아래 코드를 입력한다.

```
print("Hello Python!!!")
```

4. 단축키 [**Shift** + **F10**]을 누르면 실행 결과가 화면 하단에 표시된다.



LESSON

가상환경(Virtual Environment)

파이참은 프로젝트를 생성하면 자동으로 가상환경이 만들어진다.

가상환경 개요 및 필요성

➤ 가상환경 개요

- 파이썬 프로젝트를 위한 독립적인 환경을 만들어주는 도구이다.
- 각각의 가상환경은 프로젝트에 필요한 특정한 버전의 파이썬 및 라이브러리를 포함하며, 다른 프로젝트에 영향을 미치지 않도록 격리된 환경을 제공한다.

➤ 가상환경이 필요한 이유

- 1.버전 관리:** 각 프로젝트는 필요한 파이썬 버전 및 라이브러리의 특정 버전을 요구할 수 있습니다. 가상환경을 사용하면 각 프로젝트에 맞게 환경을 구성할 수 있습니다.
- 2.독립성:** 가상환경은 프로젝트 간에 독립성을 제공합니다. 한 프로젝트에서 사용하는 라이브러리나 패키지의 버전이 다른 프로젝트에 영향을 미치지 않습니다.
- 3.종속성 관리:** 특정 프로젝트에 필요한 라이브러리들을 명시적으로 관리할 수 있으며, 이를 통해 프로젝트의 종속성을 명확하게 알 수 있습니다.

터미널에서 가상환경 만들기

1. 프로젝트 생성

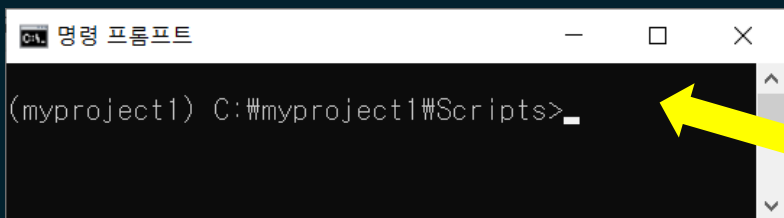
`py -m venv 프로젝트명`

2. 가상환경 활성화

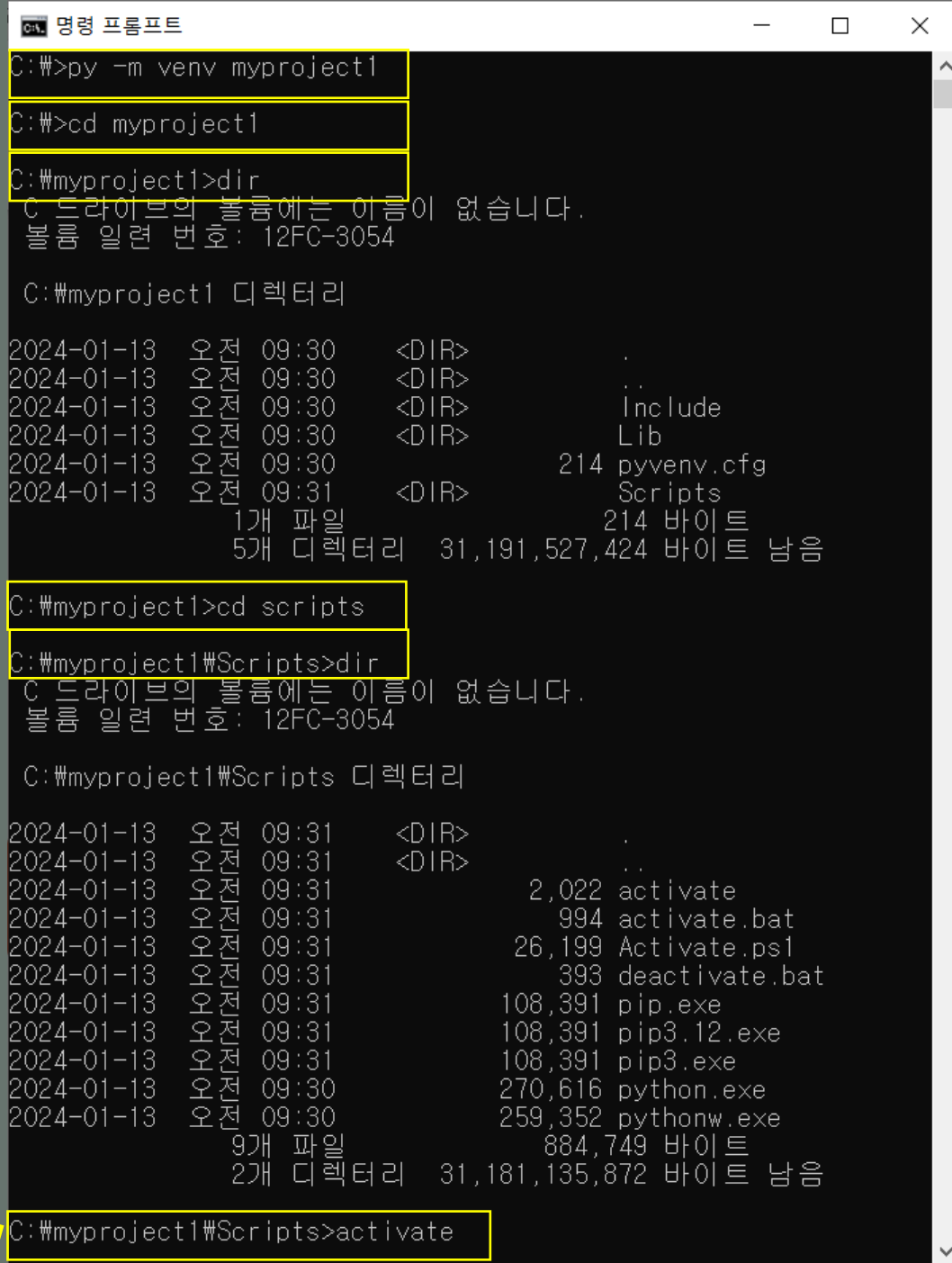
해당 프로젝트의 scripts 폴더로 이동해서 activate 명령을 입력하면 된다.

가상환경이 활성화되면 "(프로젝트명)"이 프롬프트 앞에 표시된다.

`cd 프로젝트명\scripts`
`activate`



```
(myproject1) C:\myproject1\Scripts>
```



```
C:\>py -m venv myproject1
C:\>cd myproject1
C:\myproject1>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 12FC-3054

C:\myproject1 디렉터리

2024-01-13 오전 09:30 <DIR> .
2024-01-13 오전 09:30 <DIR> ..
2024-01-13 오전 09:30 <DIR> Include
2024-01-13 오전 09:30 <DIR> Lib
2024-01-13 오전 09:30 214 pyvenv.cfg
2024-01-13 오전 09:31 <DIR> Scripts
1개 파일 214 바이트
5개 디렉터리 31,191,527,424 바이트 남음

C:\myproject1>cd scripts
C:\myproject1\Scripts>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 12FC-3054

C:\myproject1\Scripts 디렉터리

2024-01-13 오전 09:31 <DIR> .
2024-01-13 오전 09:31 <DIR> ..
2024-01-13 오전 09:31 2,022 activate
2024-01-13 오전 09:31 994 activate.bat
2024-01-13 오전 09:31 26,199 Activate.ps1
2024-01-13 오전 09:31 393 deactivate.bat
2024-01-13 오전 09:31 108,391 pip.exe
2024-01-13 오전 09:31 108,391 pip3.12.exe
2024-01-13 오전 09:31 108,391 pip3.exe
2024-01-13 오전 09:30 270,616 python.exe
2024-01-13 오전 09:30 259,352 pythonw.exe
9개 파일 884,749 바이트
2개 디렉터리 31,181,135,872 바이트 남음

C:\myproject1\Scripts>activate
```

LESSON

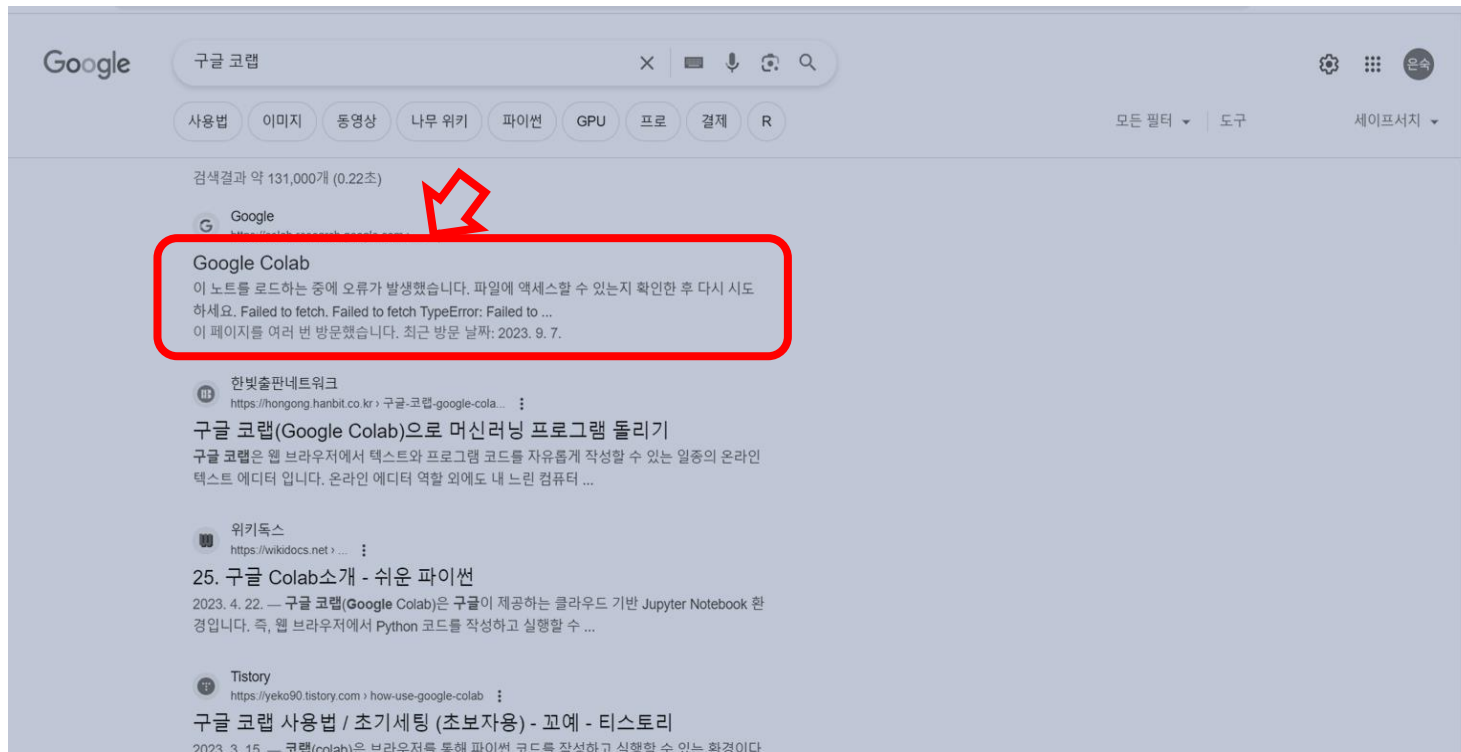
Google의 Colab

구글 로그인

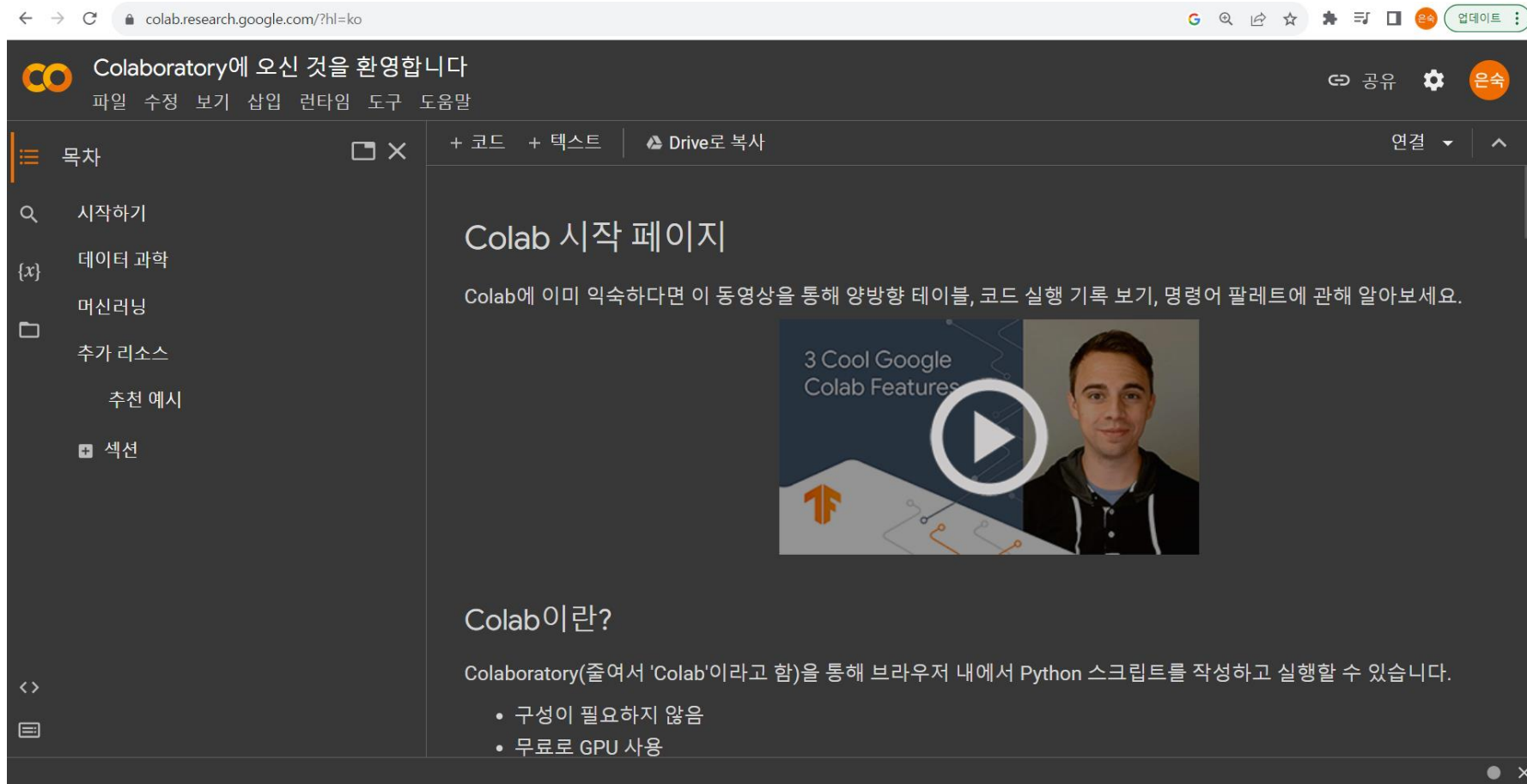
- 각자 자신의 계정으로 로그인한다.

Colab 접속

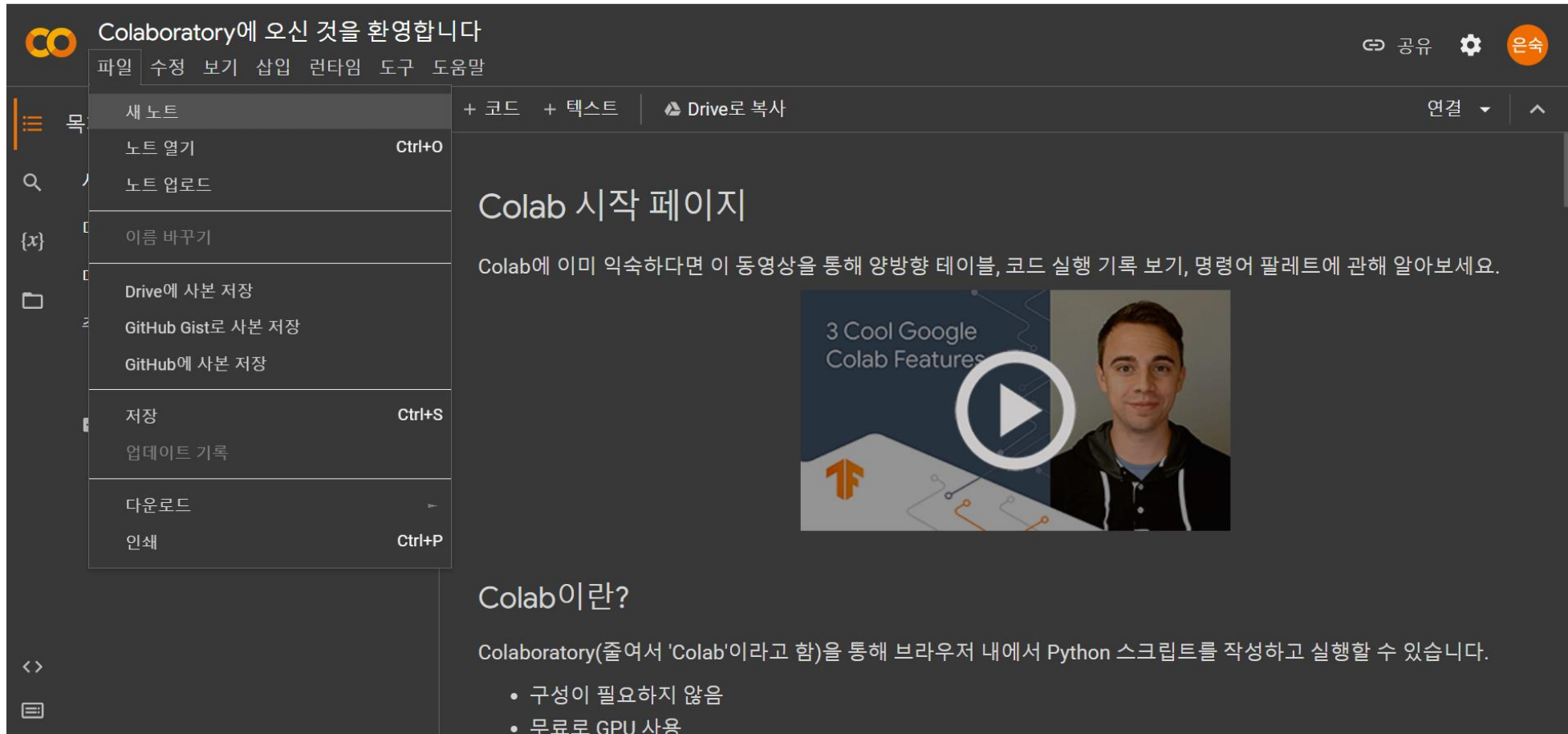
- 구글 검색창에 "구글 Colab" 입력 후 아래 클릭



Colab 접속후 화면



새로운 파일 만들기



Colaboratory에 오신 것을 환영합니다

파일 수정 보기 삽입 런타임 도구 도움말


새 노트
노트 열기 Ctrl+O
노트 업로드
이름 바꾸기
Drive에 사본 저장
GitHub Gist로 사본 저장
GitHub에 사본 저장
저장 Ctrl+S
업데이트 기록
다운로드
인쇄 Ctrl+P

+ 코드 + 텍스트 | Drive로 복사

연결 ^

Colab 시작 페이지

Colab에 이미 익숙하다면 이 동영상상을 통해 양방향 테이블, 코드 실행 기록 보기, 명령어 팔레트에 관해 알아보세요.

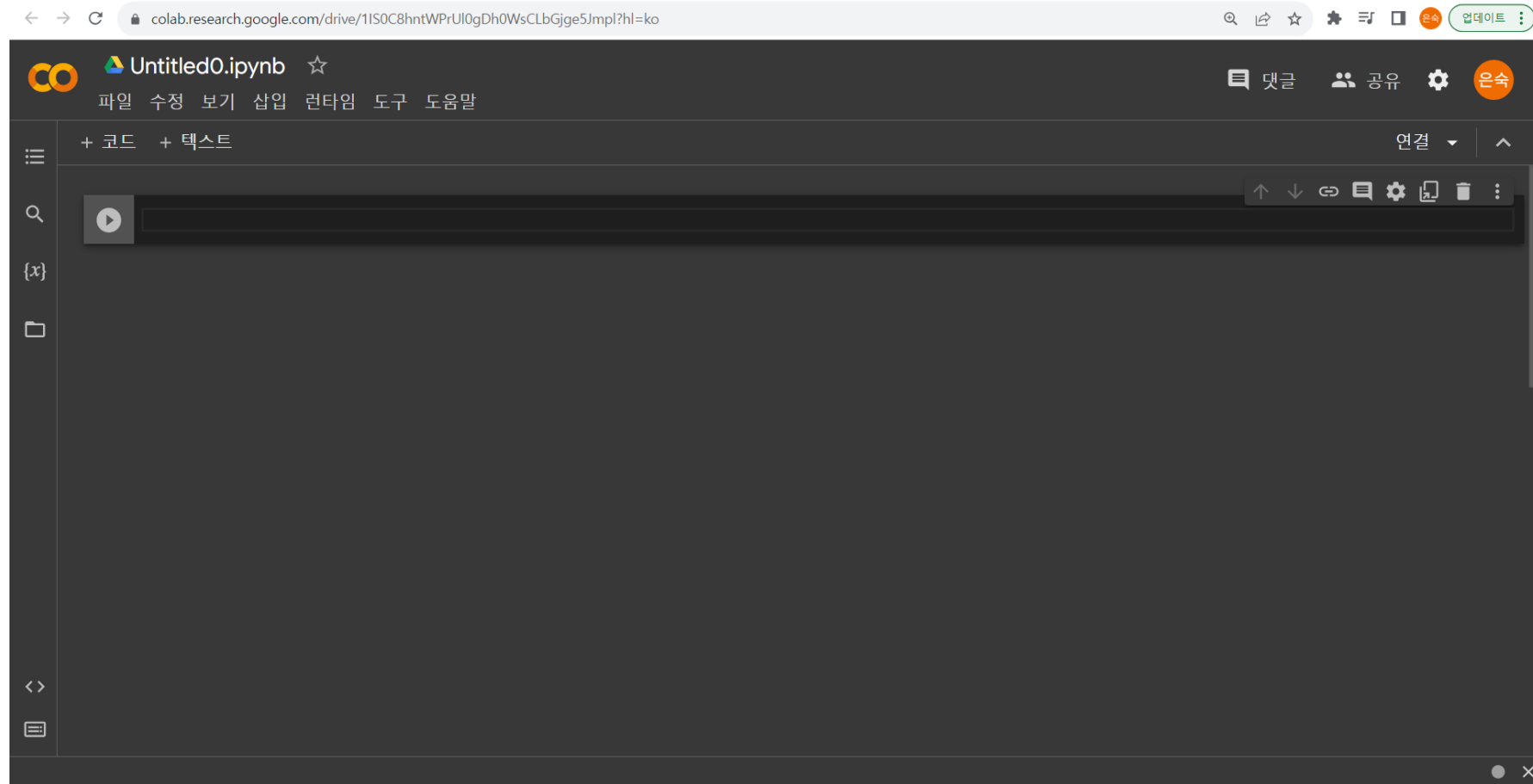


Colab이란?

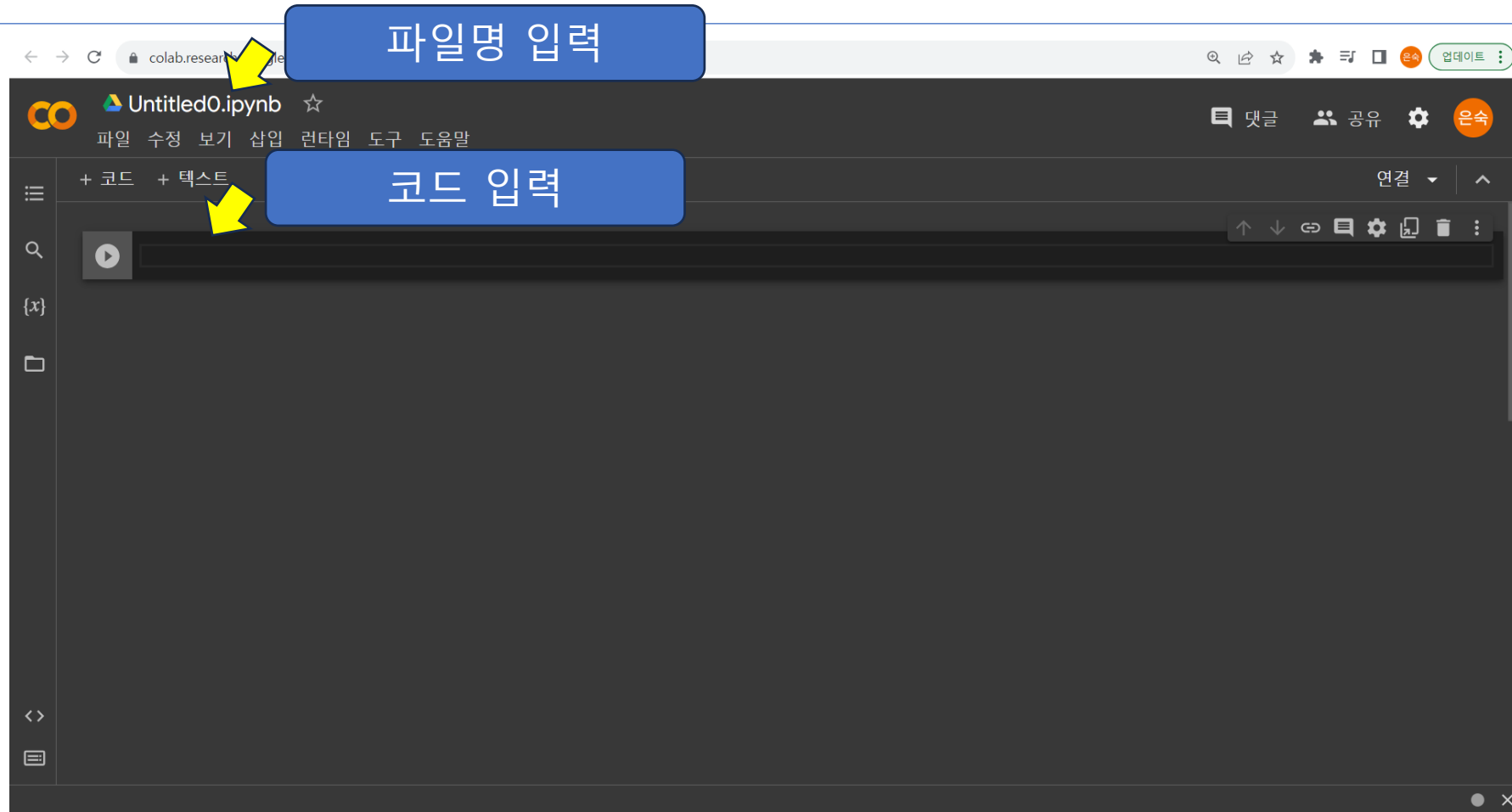
Colaboratory(줄여서 'Colab'이라고 함)을 통해 브라우저 내에서 Python 스크립트를 작성하고 실행할 수 있습니다.

- 구성이 필요하지 않음
- 무료로 GPU 사용

새로 만들기후 화면



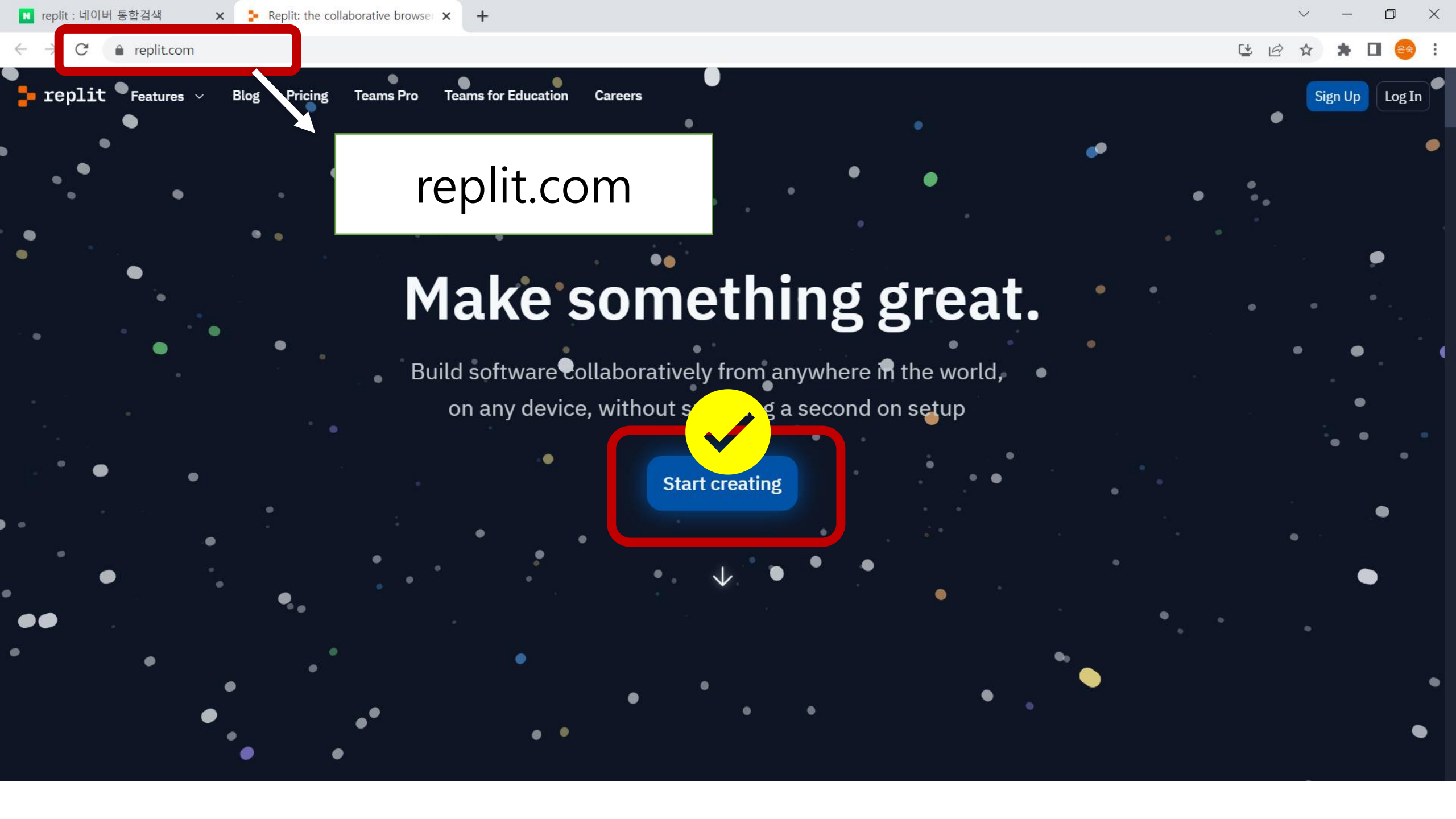
새로 만들기후 화면



LESSON

파이썬 웹 IDE

replit.com



replit.com

replit.com

Make something great.

Build software collaboratively from anywhere in the world,
on any device, without setting up a second machine

Start creating



Create a Replit account

Teacher? Sign up for our education product


+ Create Account





Have an account? [Log In](#)


Trouble signing up? [Get help](#)





By continuing, you agree to Replit's [Terms of Service](#) and [Privacy Policy](#), and to receiving emails with updates.

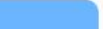


 Continue with Google



 Continue with Github

 Continue with Facebook

 Continue with Apple







Search & run commands

Ctrl .



Please verify your email.

Resend verification link

Home

Templates

My Repls

My Cycles 0

Community

Bounties

Learn

Teams

Curriculum

Use Replit App

Blog
About
Careers
Pricing
Discord
Terms

? Help



@hollykes, your AI pair programmer is waiting for you.

Learn more

Get started

0/3

See what you can accomplish on Replit

Learn

→ Explore tutorials

Share

→ See what people are building

Code

→ Create your first Repl

My Repls

+ Create

See all Repls

Explore Tutorials

repl: 네이버 통합검색

Home - Replit

replit.com/~

Search & run commands

Ctrl .

Please verify your email. [Resend verification link](#)

+ Create

Power Up

Home

Templates

My Repls

My Cycles 0

Community

Bounties

Learn

Teams

Curriculum

Use Replit App

Blog

About

Careers

Pricing

Discord

Terms

Help

Create a Repl

Import from GitHub

Template

Python

Python

Python is a high-level, interpreted, general-purpose programming language.

replit 2K + 16.6M

Title

ClassicCandidDowngrade

Public

Anyone can view and fork this Repl.

Power Up to make private

+ Create Repl

원하는 이름 입력

🔍 Search

📁 Files

🐍 main.py

Packager files

🐍 poetry.lock

🐍 pyproject.toml

📁 Tools

📄 Docs

💬 Chat

📧 Threads

📦 Packages

🔗 Git

🔍 Debugger

🐚

🔒

🧑🏻 Try Ghostwriter

CPU

RAM

Storage

? Help

🐍 main.py x +

1 Not sure what to do? Run some [examples](#) or [generate code with Ghostwriter](#) (start typing to dismiss)

Line 1 : Col 1

>_ Console x 🐚 Shell x +

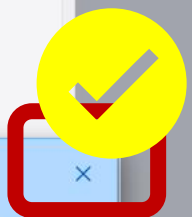
History 🕒

Welcome to the Workspace

The Workspace is where you create projects on Replit. These are called **Repls**.
Let's explore around!

● ● ● ● ● ● ●

> Next



main.py - ClassicCandidDowngrade x +

replit.com/@hollykes/ClassicCandidDowngrade#main.py

ClassicCandidDowngrade hollykes

Run

Search

Files

- main.py
- introduce.txt

Package files

- poetry.lock
- pyproject.toml

Tools

- Docs
- Chat
- Threads
- Packages
- Git
- Debugger
- Shell
- Terminal
- Lock

Try Ghostwriter

CPU RAM Storage

? Help

main.py x +

```
1 f = open('introduce.txt', 'w')
2 f.write('안녕하세요.\n')
3 f.write('python을 공부중입니다.')
4 f.close()
5
```

Console x

Shell x

탭을 드래그하면 탭을 분리할 수 있다.

Line 5 : Col 1

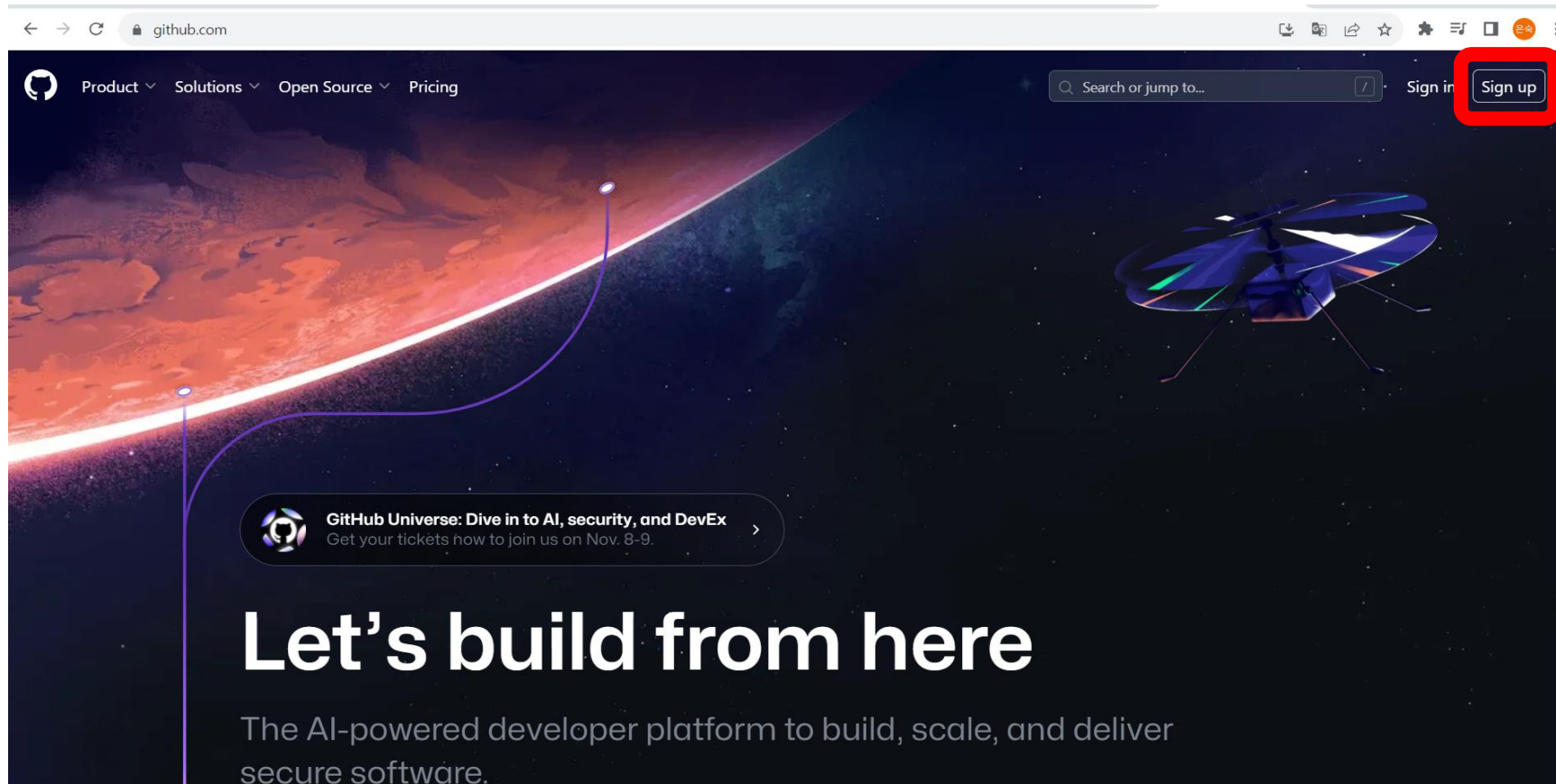
History

LESSON

웹 저장소 Github

github 가입하기

github.com



C언어로 두 수 계산

```
#include <stdio.h>
```

```
int main() {  
    // 변수 선언
```

```
    double a, b;
```

```
    a = 10
```

```
    b = 20
```

```
    // 덧셈, 뺄셈, 곱셈, 나눗셈 결과 출력
```

```
    printf("덧셈 결과: %.2f\n", a + b);
```

```
    printf("뺄셈 결과: %.2f\n", a - b);
```

```
    printf("곱셈 결과: %.2f\n", a * b);
```

```
    printf("나눗셈 결과: %.2f\n", a / b);
```

```
    return 0;
```

```
}
```

파이썬 언어로 두 수 계산

```
# 변수 선언
```

```
a = 10
```

```
b = 20
```

```
// 덧셈, 뺄셈, 곱셈, 나눗셈 결과 출력
```

```
print(a, '+', b, '=', a + b)
```

```
print(a, '-', b, '=', a - b)
```

```
print(a, '*', b, '=', a * b)
```

```
print(a, '/', b, '=', a / b)
```

