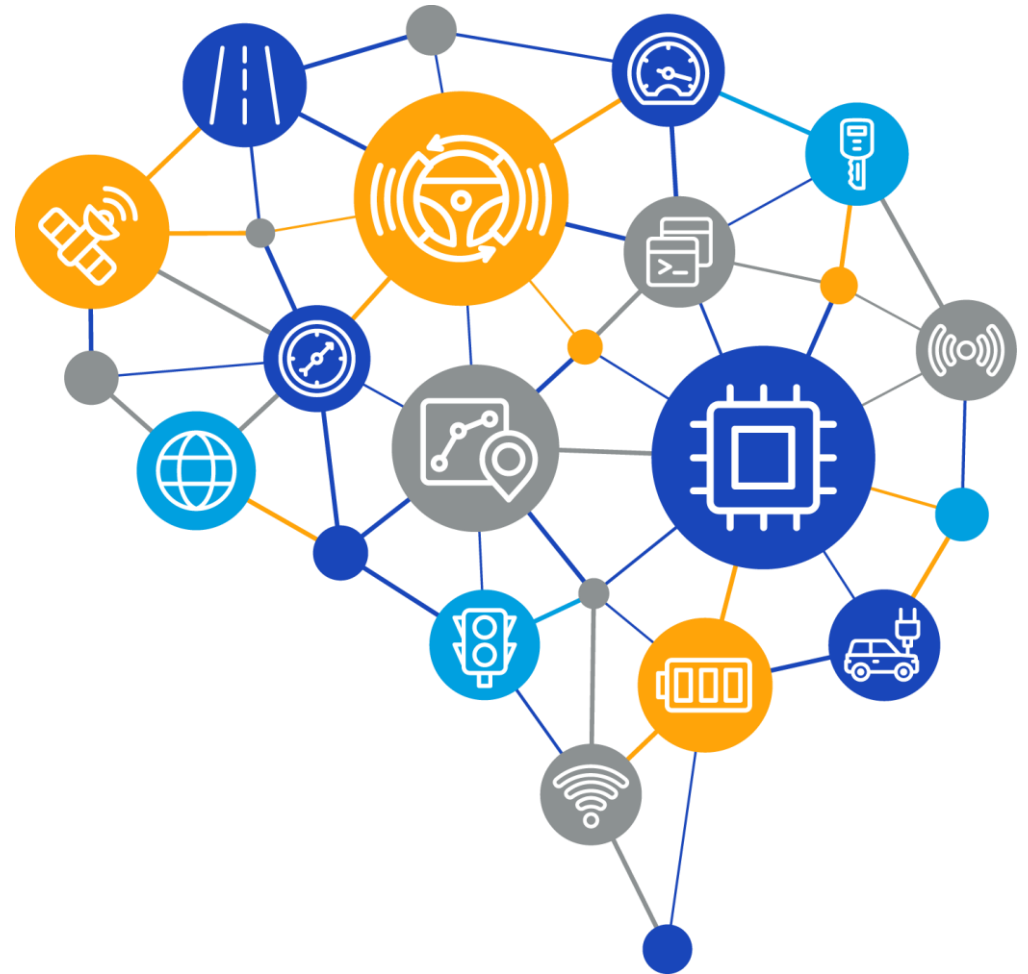


파이참 가상환경 설정

Development for Python

2022.08

강환수 교수

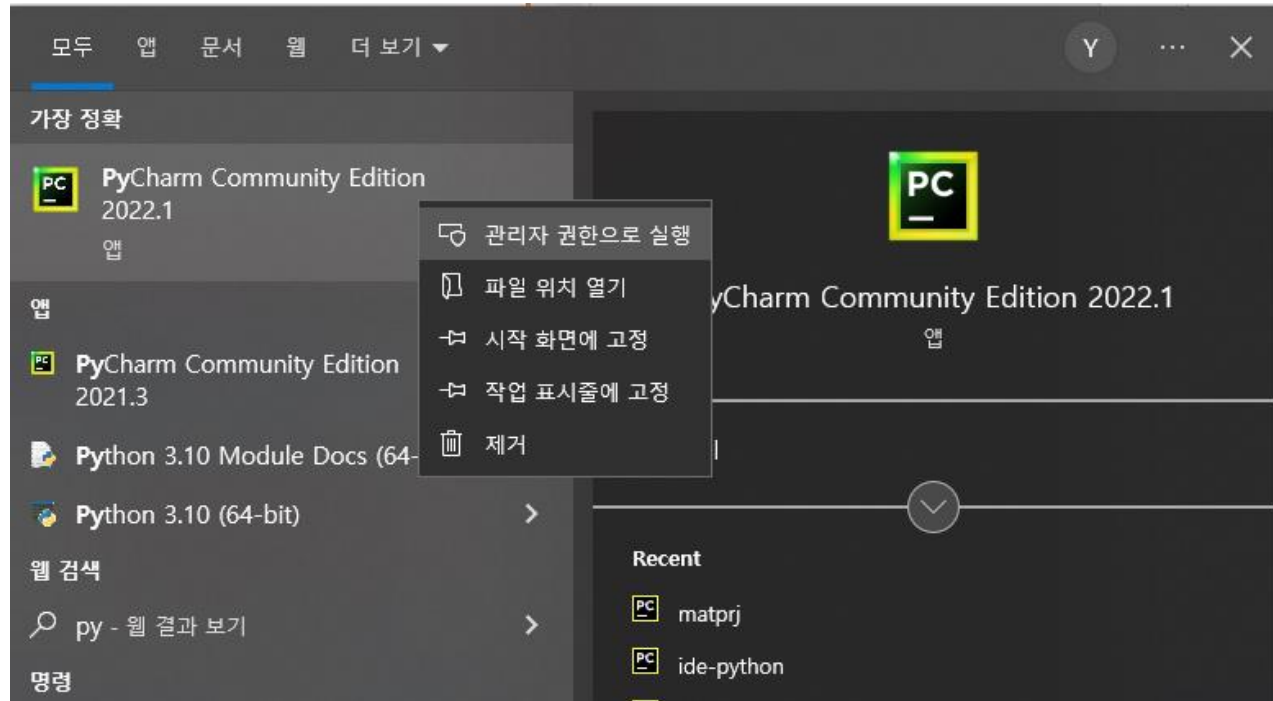


AI Experts
Who Lead
The Future

01

파이참에서 이미 생성된 가상환경 설정

- 관리자 권한으로 실행



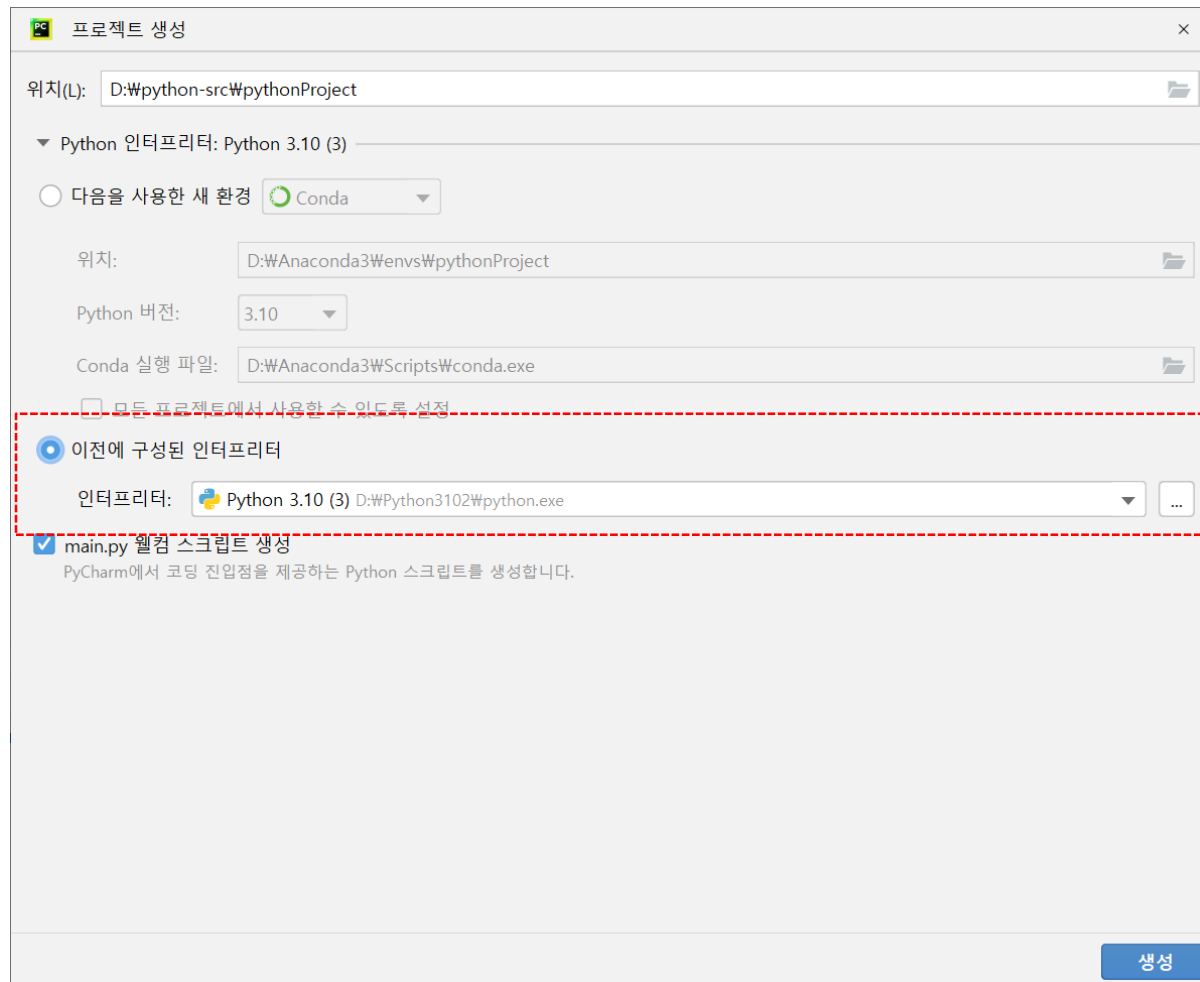
이미 생성된 Existing Interpreter 지정

파이썬 가상환경 Python language

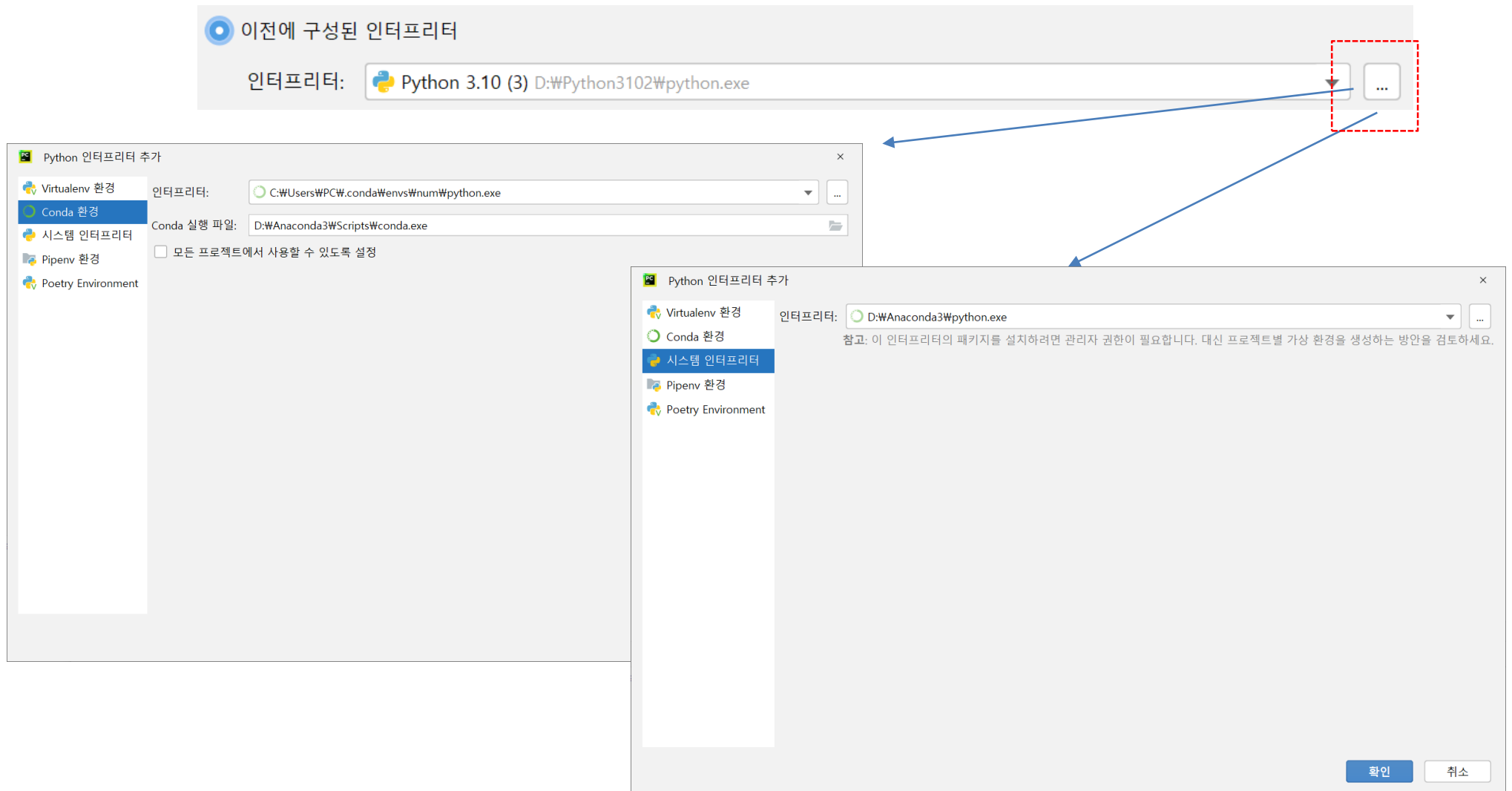
- Existing interpreter

- ... 선택

- 자신이 만든 가상환경 폴더의 scripts 하부 python.exe를 지정



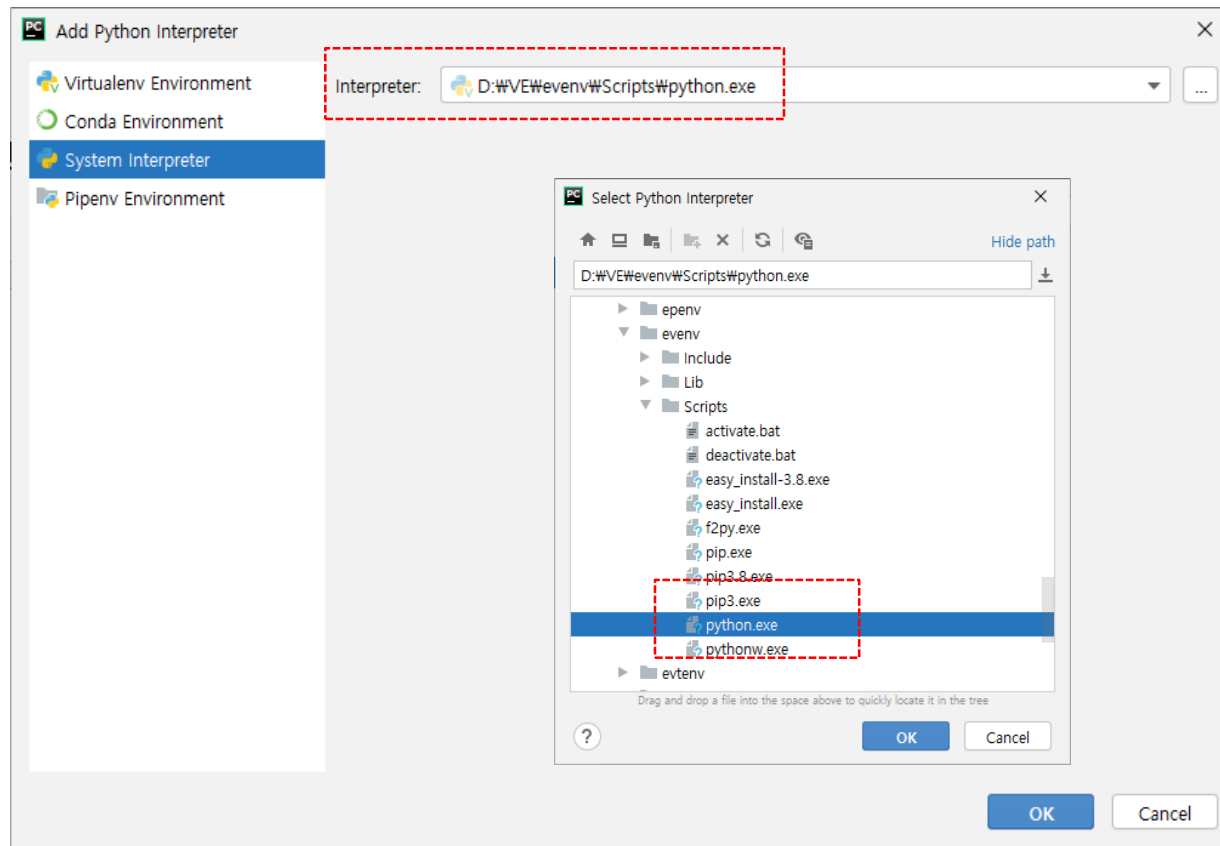
- Conda 또는 시스템 인터프리터



파이썬 프로젝트 pyc prj01 만들기(1)

파이썬 가상환경 Python language

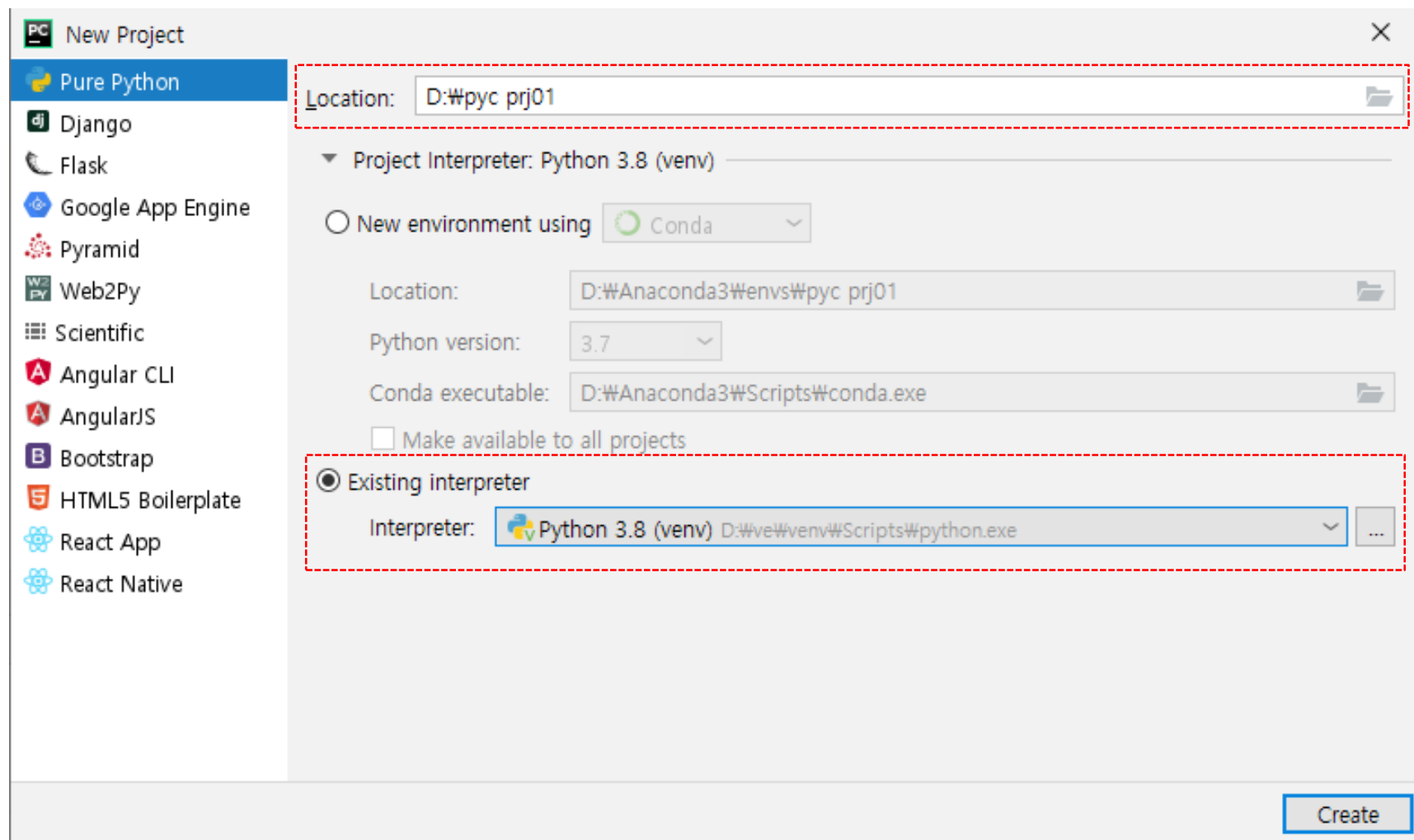
- D:\Wpyc prj01
 - 가상환경 venv의 인터프리터 지정
 - venv로 만든 evenv 지정
 - ...을 눌러 자신이 virtualenv로 직접 만든 가상환경의 scripts 폴더의 python.exe를 지정



파이참 프로젝트 pyc prj01 만들기(2)

파이썬 가상환경 Python language

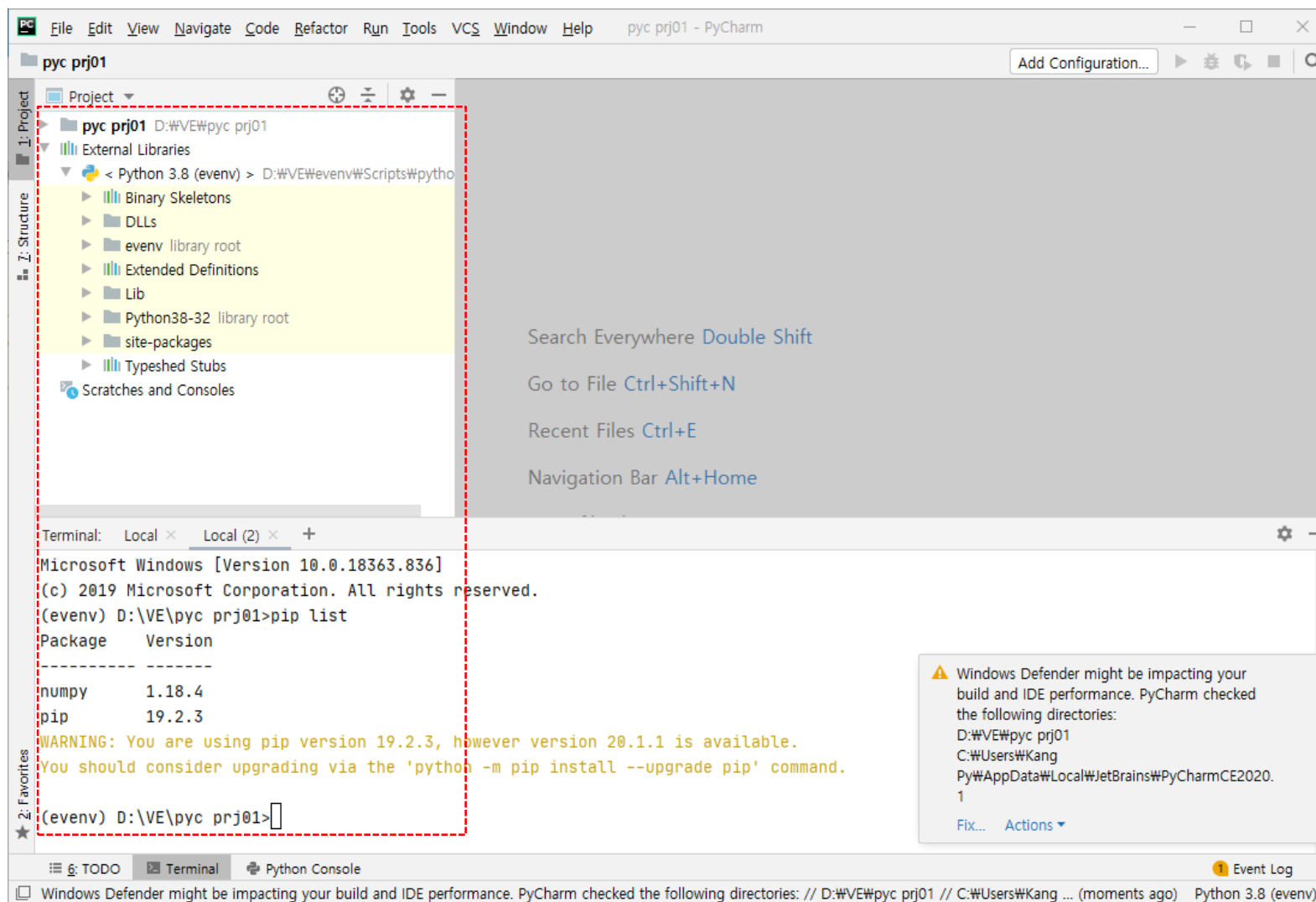
- 자신이 만든 가상환경이 지정된 New Project 대화상자



파이참 프로젝트 pyc prj01 만들기(3)

파이썬 가상환경 Python language

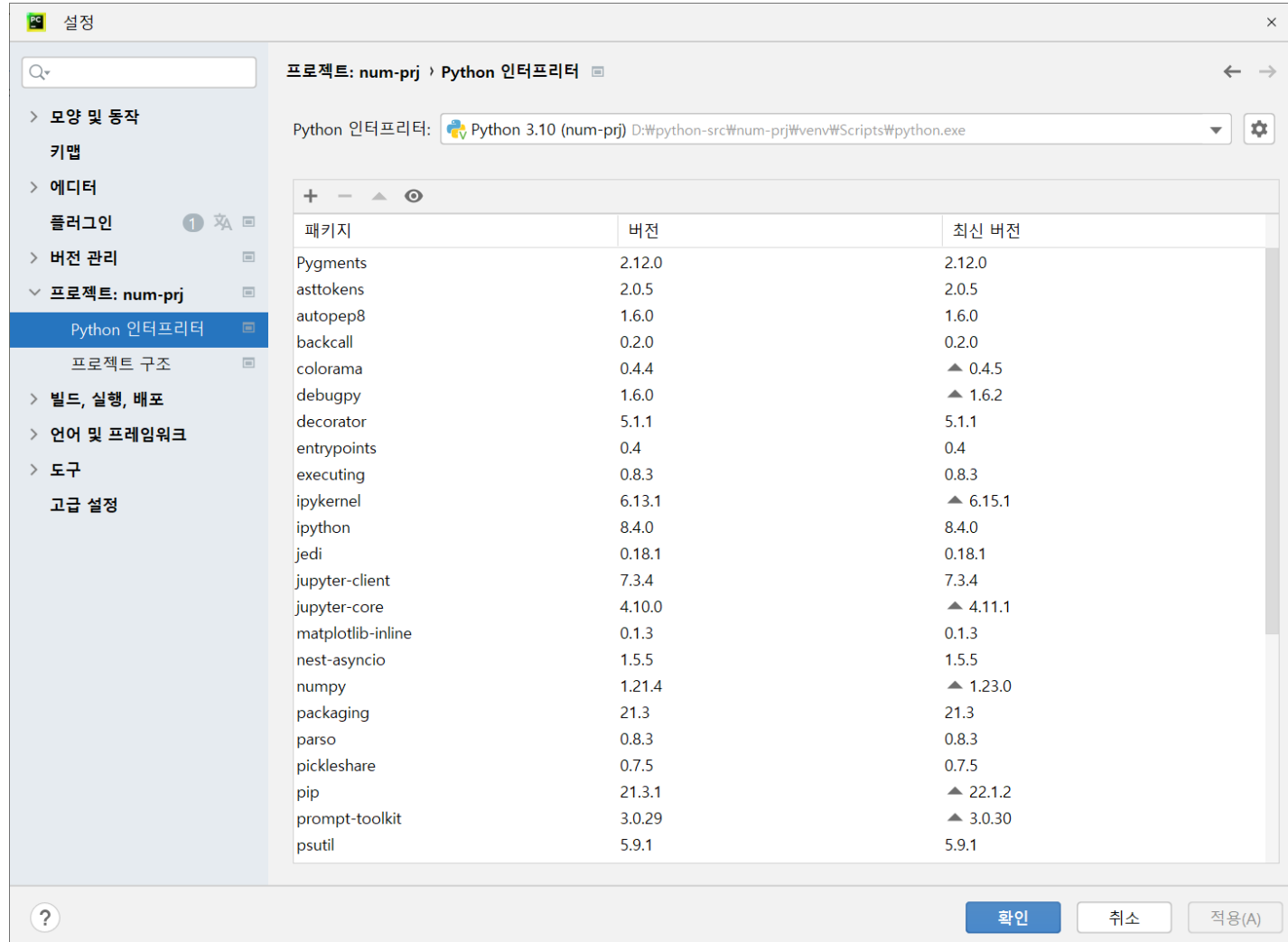
- 가상환경이 설정된 프로젝트



설정에서 지정하는 방법

파이썬 가상환경 Python language

- 메뉴 파일 | 설정
 - Ctrl + alt + s



AI Experts
Who Lead
The Future

02

파이참 프로젝트 생성 시
직접 가상환경 만들어 설정

3개의 프로젝트 생성

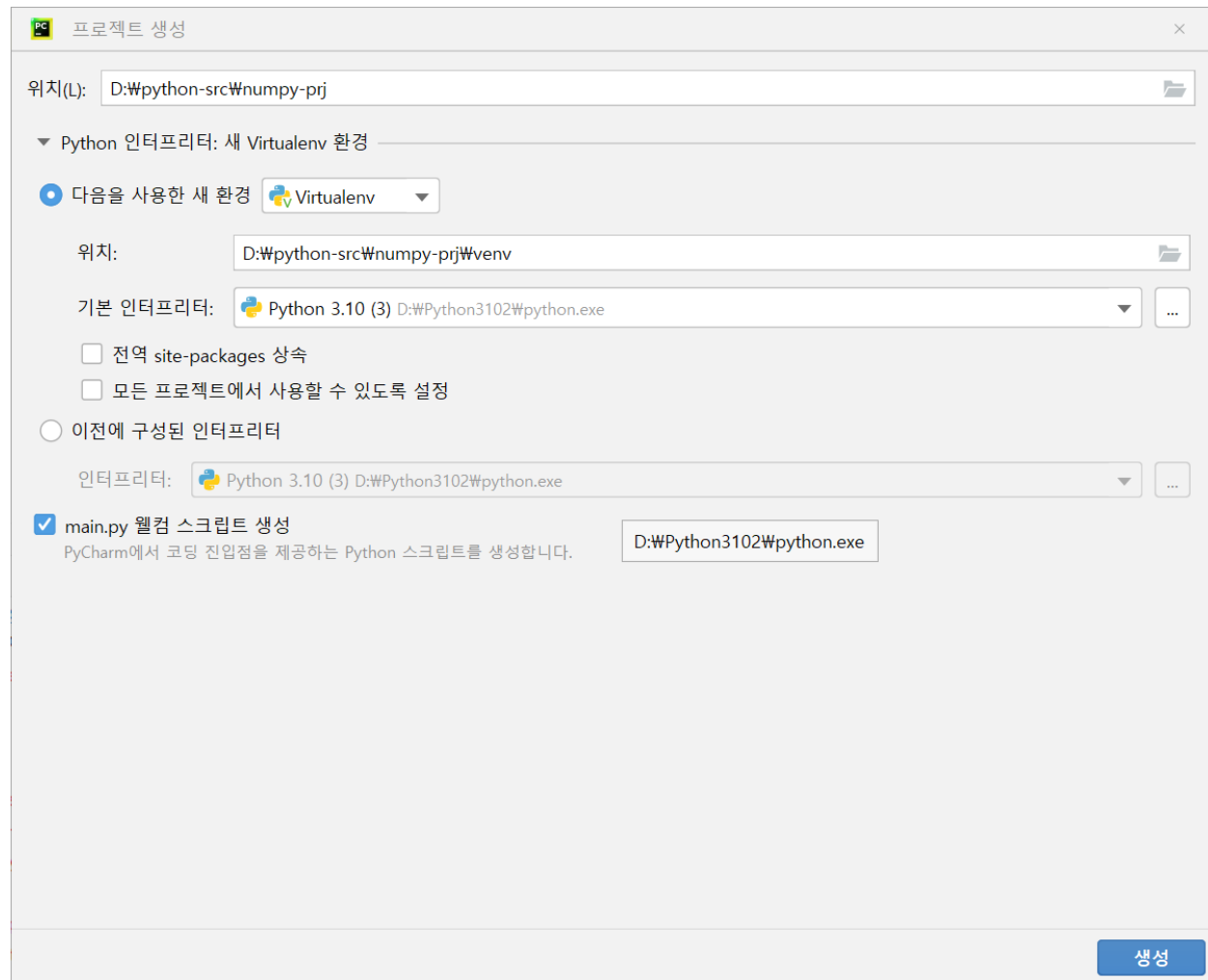
파이썬 가상환경 Python language

- 프로젝트 **numpy-prj**
 - Virtualenv 도구 사용
- 프로젝트 **conda-prj**
 - Conda 도구 사용

파이참에서 직접 가상환경 만들기

파이썬 가상환경 Python language

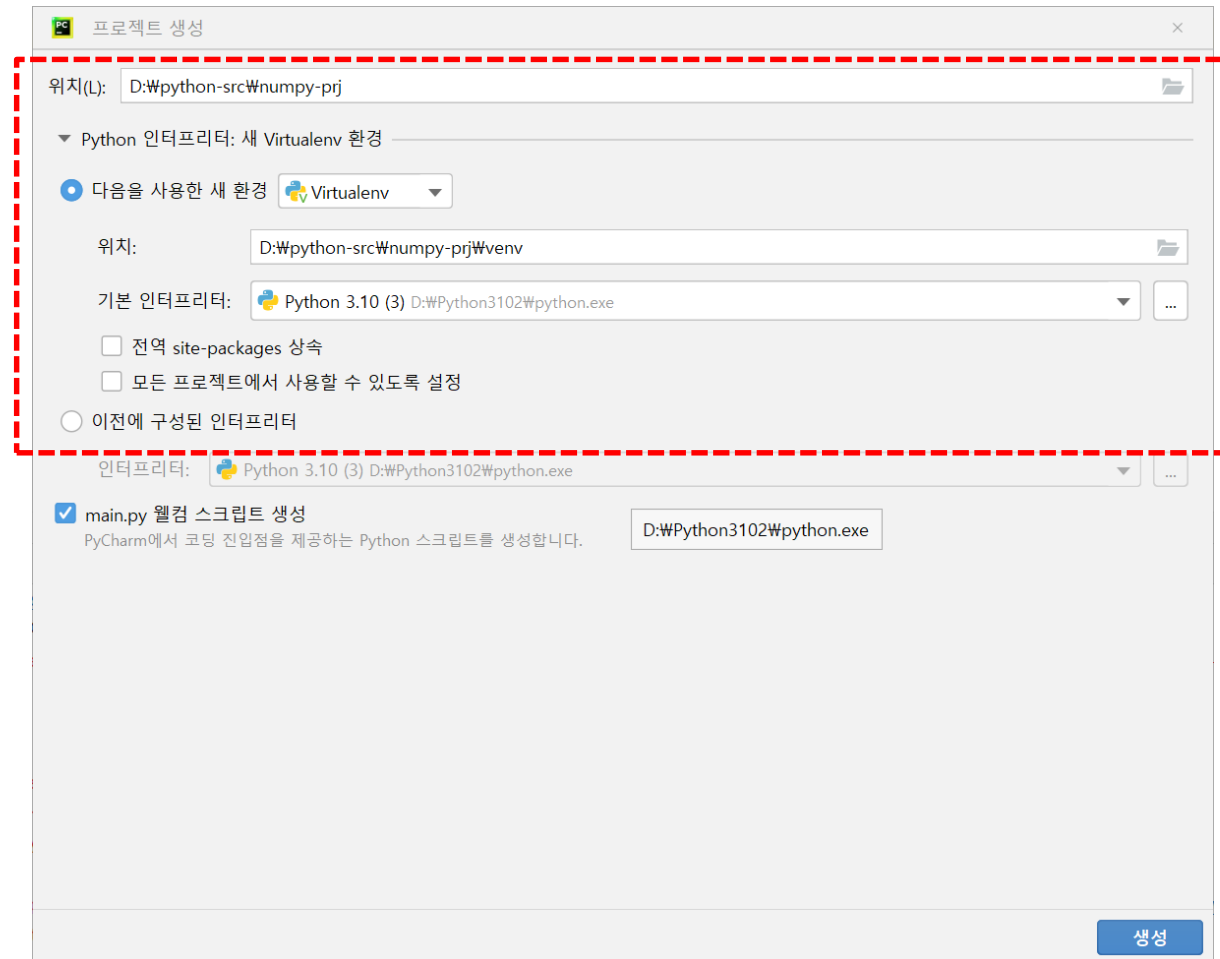
- New Project에서
 - New environment using 종류 3가지
 - Virtualenv
 - virtualenv 설치
 - Conda
 - 아나콘다 또는 미니콘다 설치



파이참 Virtualenv으로 생성

파이썬 가상환경 Python language

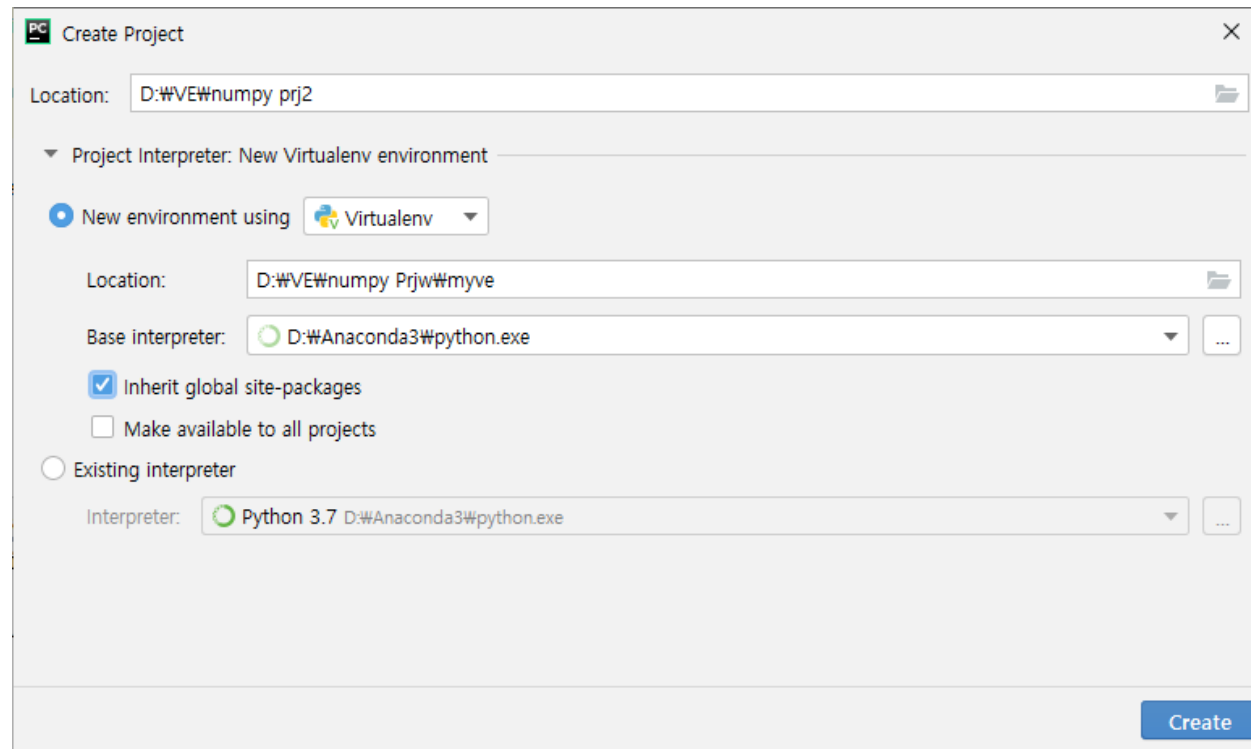
- **Location:**
 - 프로젝트 이름
- **New Environment using**
 - Virtualenv



옵션

파이썬 가상환경 Python language

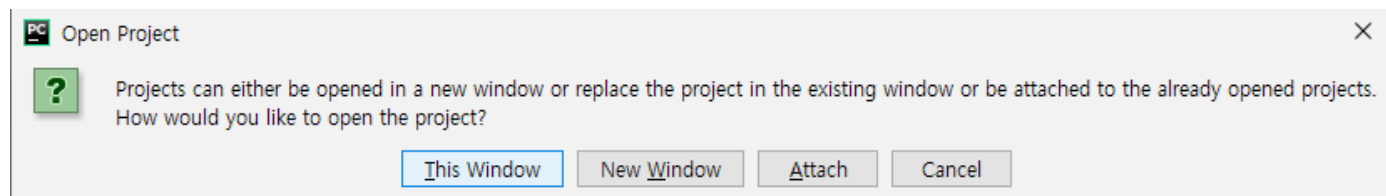
- 가상환경 위치 변경도 가능
 - 수정도 가능
 - 프로젝트 폴더와 가상환경이름
- Inherit global site-packages
 - 베이스 인터프리터의 설치 모듈과 같이 설치



프로젝트 창 생성 방법

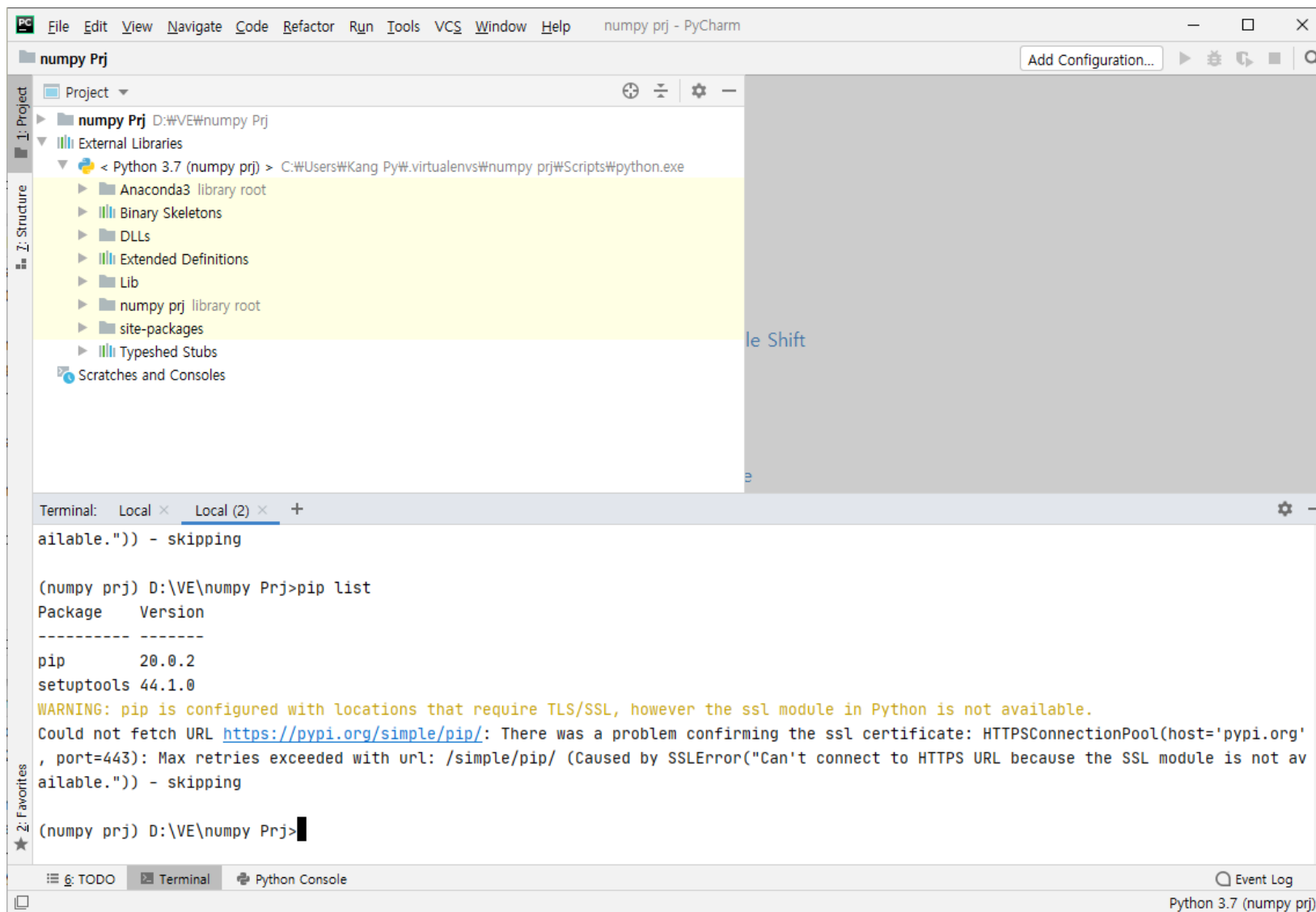
파이썬 가상환경 Python language

- **This Window**
 - 현재 프로젝트는 없어지고 선택한 프로젝트가 창에 보임
 - 이전의 프로젝트는 사라짐
- **New Window(가장 선호)**
 - 현재 프로젝트도 계속 유지
 - 다른 독립적인 창에 선택한 프로젝트가 보임
 - 여러 파이참 윈도우가 표시되며 각각의 파이참을 처리 가능
- **Attach**
 - 현재 프로젝트는 하부에 선택한 프로젝트가 함께 보임
 - 다만 기존의 가상환경을 그대로 사용
 - 설정에서 지정한 환경이 무시됨
 - 기존 가상환경을 그대로 사용하는 프로젝트를 만들 때 사용



numpy prj 확인

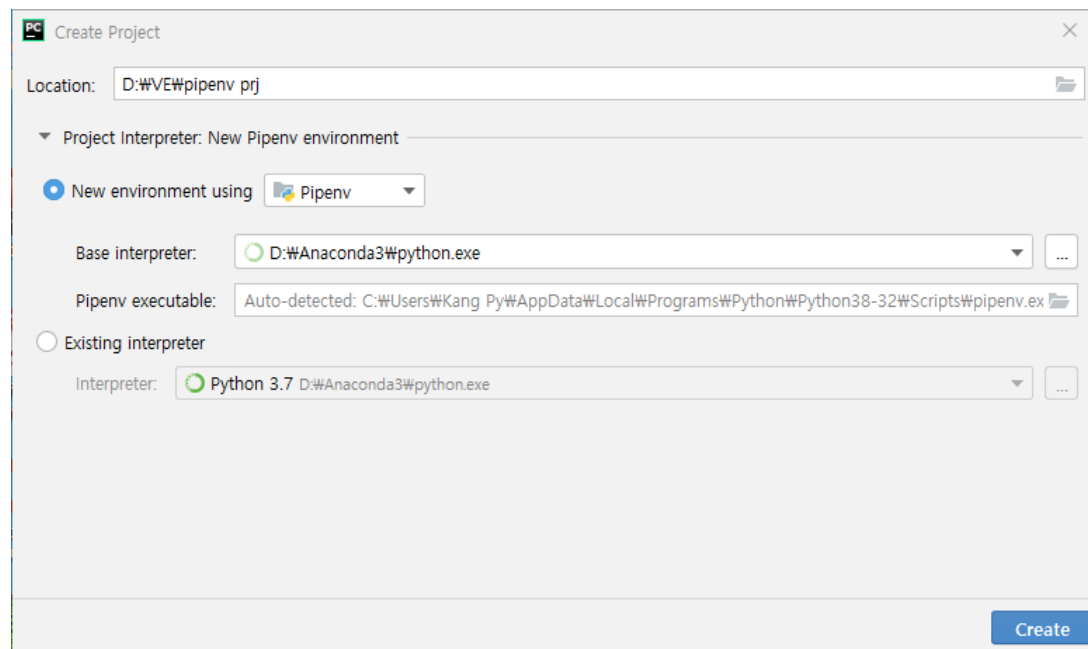
파이썬 가상환경 Python language



파이참 Pipenv로 생성

파이썬 가상환경 Python language

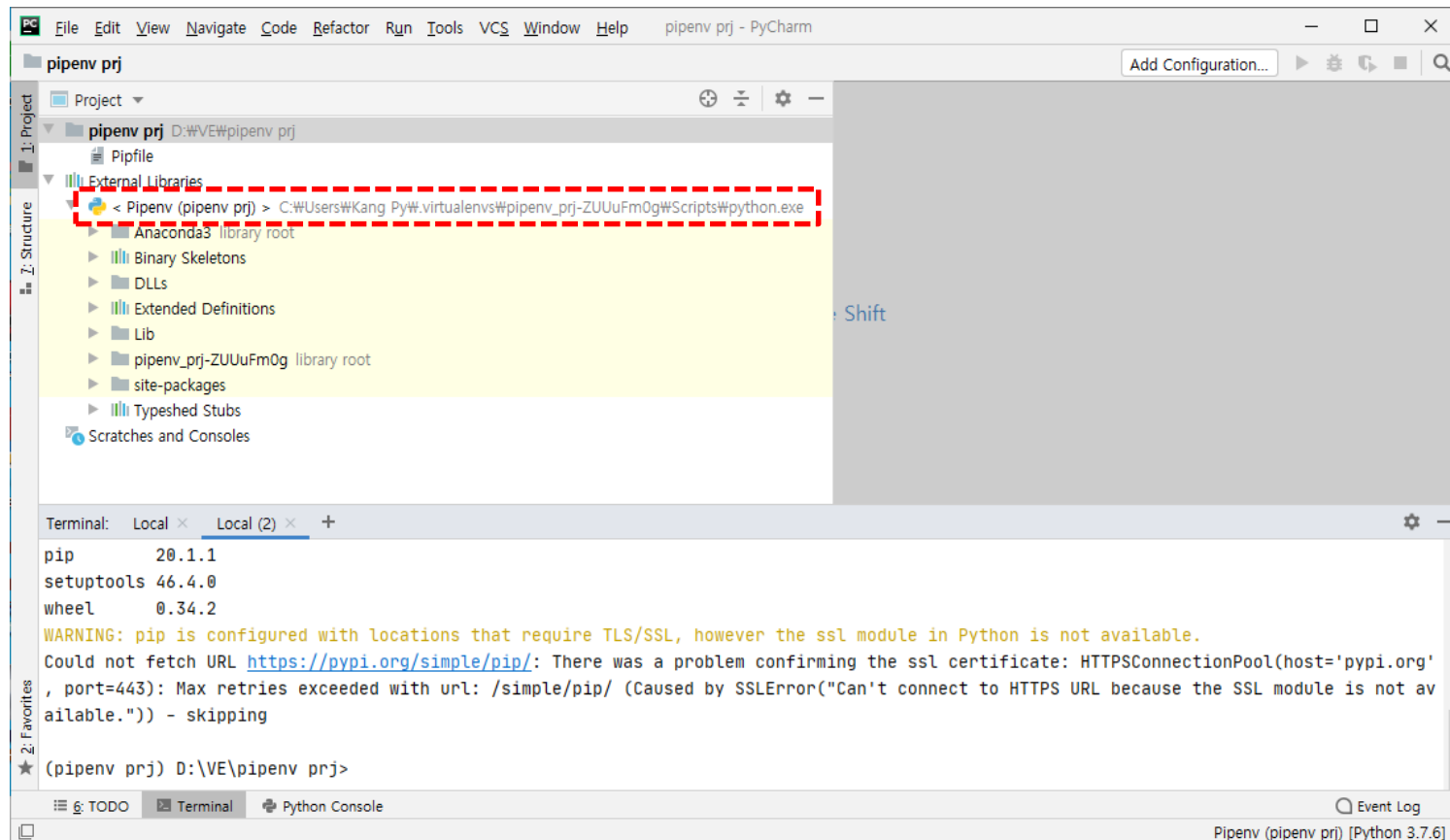
- **Location:**
 - 프로젝트 이름
- **New Environment using**
 - Pipenv
 - Base interpreter
 - **적정한 인터프리터 지정**
 - 아나콘다나 파이썬 기본
 - Pipenv executable
 - **D:\W...\Scripts\pipenv.exe**
 - 이미 설치가 되어 있어야 함



pipenv prj 확인

파이썬 가상환경 Python language

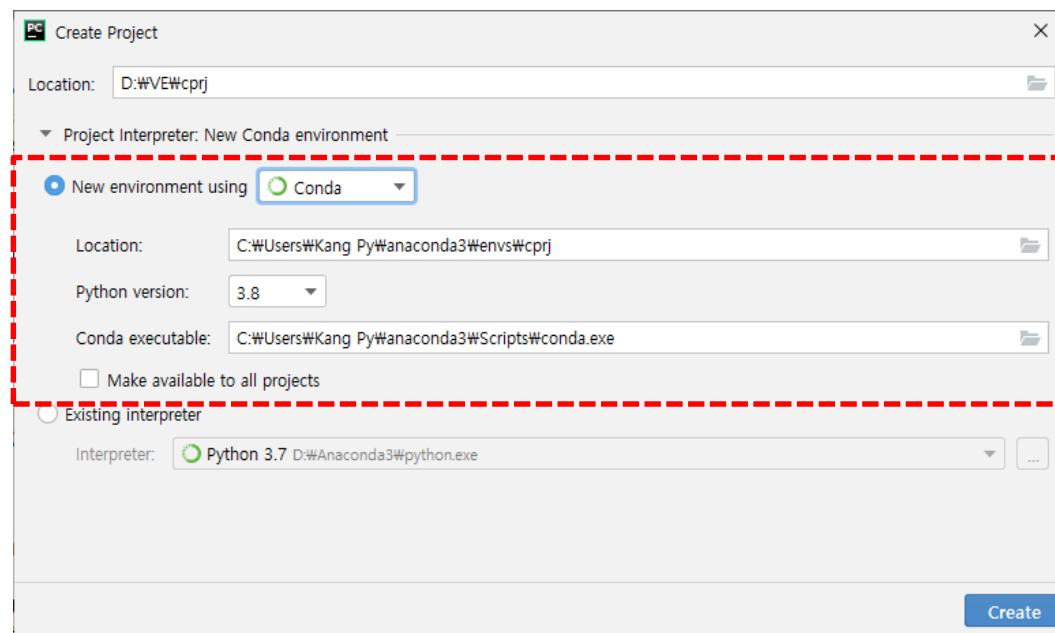
- 가상환경은 시스템이 자동으로 생성해 연결
 - 기본적으로 이름은 프로젝트폴더 이름과 유사한 이름으로 자동 지정
 - pipenv prj
 - pipenv_prj-0000



파이참 conda로 생성

파이썬 가상환경 Python language

- **Location:**
 - 프로젝트 이름
 - `conda prj`
- **New Environment using**
 - **Location**
 - 가상환경이 만들어지는 폴더
 - 기본은 `anaconda3\envs\하부`, 프로젝트 폴더 하부로 해도 가능
 - **Python version**
 - **Conda executable**
 - 콘다가 설치되어 있어야 함, 아나콘다나 미니콘다(miniconda) 설치 필요

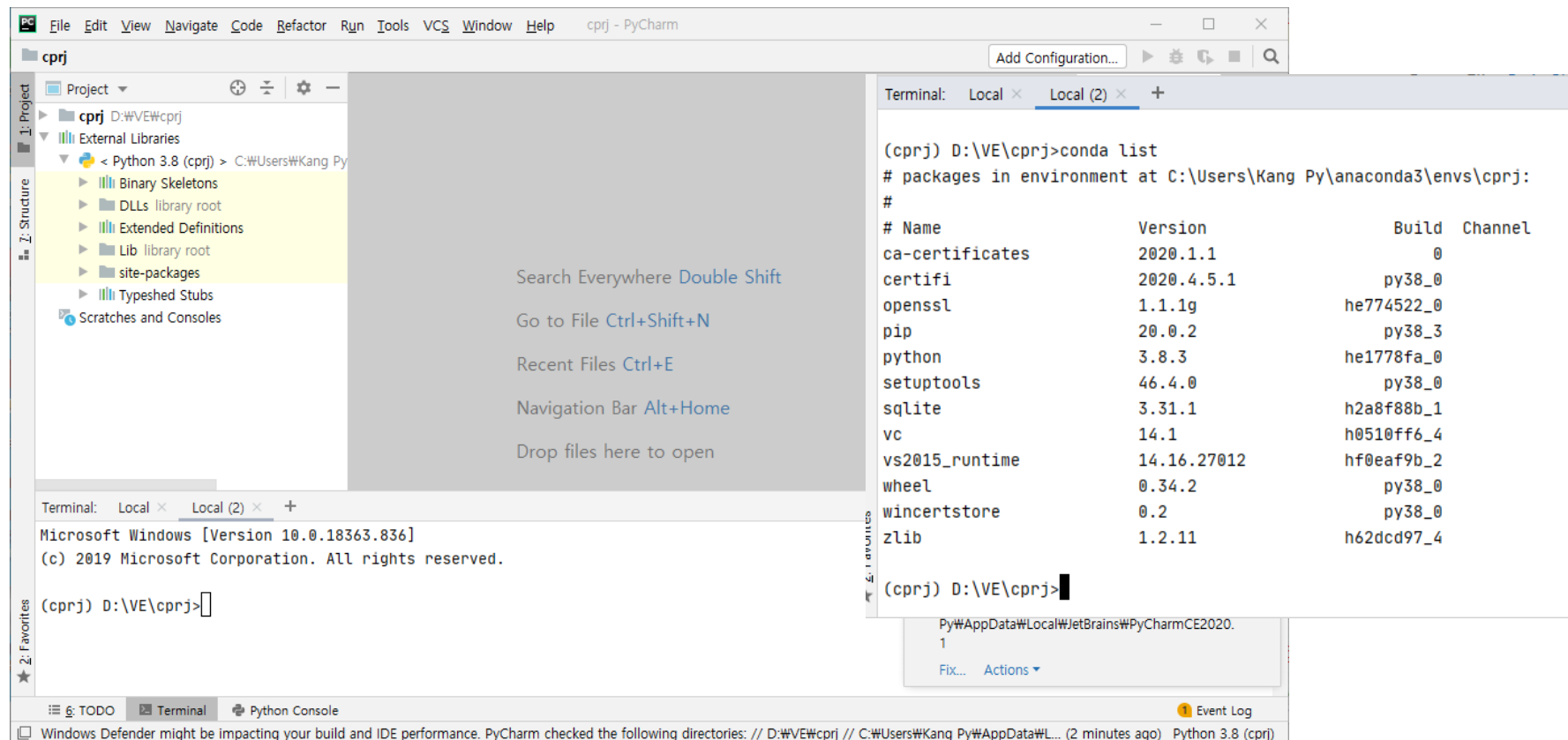


cond prj 확인

파이썬 가상환경 Python language

• Conda로 만들어지는 가상환경

- 다른 가상환경 설치와 비교해 기본적으로 설치되는 모듈이 많음
- 가상환경 이름
 - **conda prj**



추가로 패키지를 설치하는 방법 1

파이썬 가상환경 Python language

- 터미널에서 설치 가능
 - conda install numpy

```
(cprj) D:\VE\cprj>conda list
# packages in environment at C:\Users\Kang Py\anaconda3\envs\cprj:
#
# Name                        Version                        Build      Channel
blas                          1.0                            mkl
ca-certificates               2020.1.1                        0
certifi                       2020.4.5.1                      py38_0
icc_rt                         2019.0.0                       h0cc432a_1
intel-openmp                  2020.1                          216
mkl                           2020.1                          216
mkl-service                   2.3.0                          py38hb782905_0
mkl_fft                       1.0.15                         py38h14836fe_0
mkl_random                    1.1.0                          py38hf9181ef_0
numpy                         1.18.1                         py38h93ca92e_0
numpy-base                    1.18.1                         py38hc3f5095_1
openssl                       1.1.1g                          he774522_0
pip                           20.0.2                         py38_3
python                        3.8.3                          he1778fa_0
setuptools                    46.4.0                         py38_0
six                            1.14.0                         py38_0
sqlite                        3.31.1                         h2a8f88b_1
vc                             14.1                           h0510ff6_4
vs2015_runtime                14.16.27012                    hf0eaf9b_2
wheel                         0.34.2                         py38_0
wincertstore                  0.2                            py38_0
zlib                          1.2.11                         h62dcd97_4

(cprj) D:\VE\cprj>
```

```
(cprj) D:\VE\cprj>conda install numpy
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.8.2
  latest version: 4.8.3

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\Kang Py\anaconda3\envs\cprj

  added / updated specs:
    - numpy

The following packages will be downloaded:

  package | build | size
  -----|-----|-----
  intel-openmp-2020.1 | 216 | 1.6 MB
  mkl-2020.1 | 216 | 99.3 MB
  mkl-service-2.3.0 | py38hb782905_0 | 50 KB
  mkl_fft-1.0.15 | py38h14836fe_0 | 119 KB
  mkl_random-1.1.0 | py38hf9181ef_0 | 243 KB
  numpy-1.18.1 | py38h93ca92e_0 | 6 KB
  numpy-base-1.18.1 | py38hc3f5095_1 | 3.8 MB
  six-1.14.0 | py38_0 | 27 KB
  -----|-----|-----
  Total: | | 105.2 MB

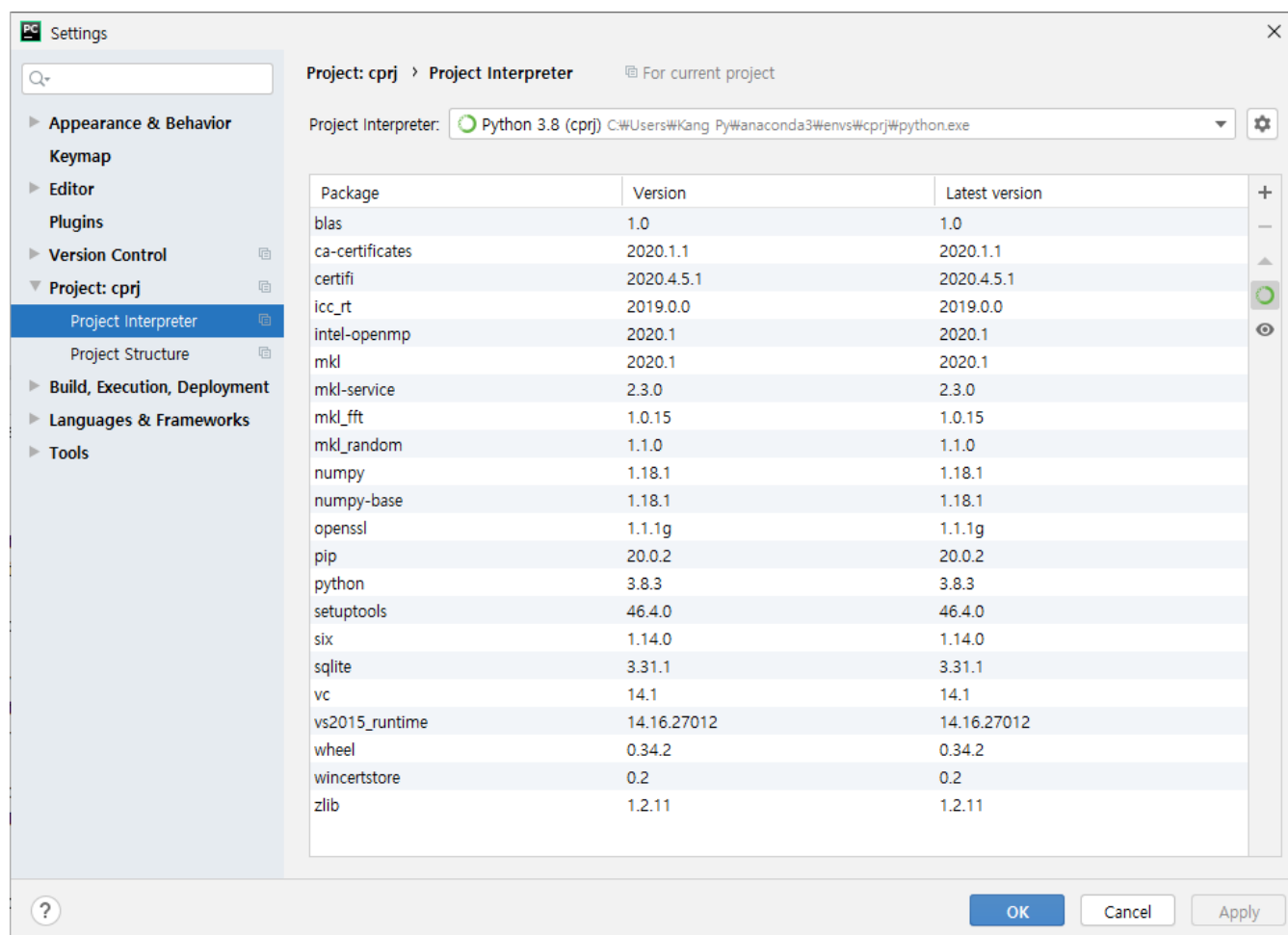
The following NEW packages will be INSTALLED:

  blas pkgs/main/win-64::blas-1.0-mkl
  icc_rt pkgs/main/win-64::icc_rt-2019.0.0-h0cc432a_1
  intel-openmp pkgs/main/win-64::intel-openmp-2020.1-216
  mkl pkgs/main/win-64::mkl-2020.1-216
  mkl-service pkgs/main/win-64::mkl-service-2.3.0-py38hb782905_0
  mkl_fft pkgs/main/win-64::mkl_fft-1.0.15-py38h14836fe_0
  mkl_random pkgs/main/win-64::mkl_random-1.1.0-py38hf9181ef_0
  numpy pkgs/main/win-64::numpy-1.18.1-py38h93ca92e_0
  numpy-base pkgs/main/win-64::numpy-base-1.18.1-py38hc3f5095_1
```

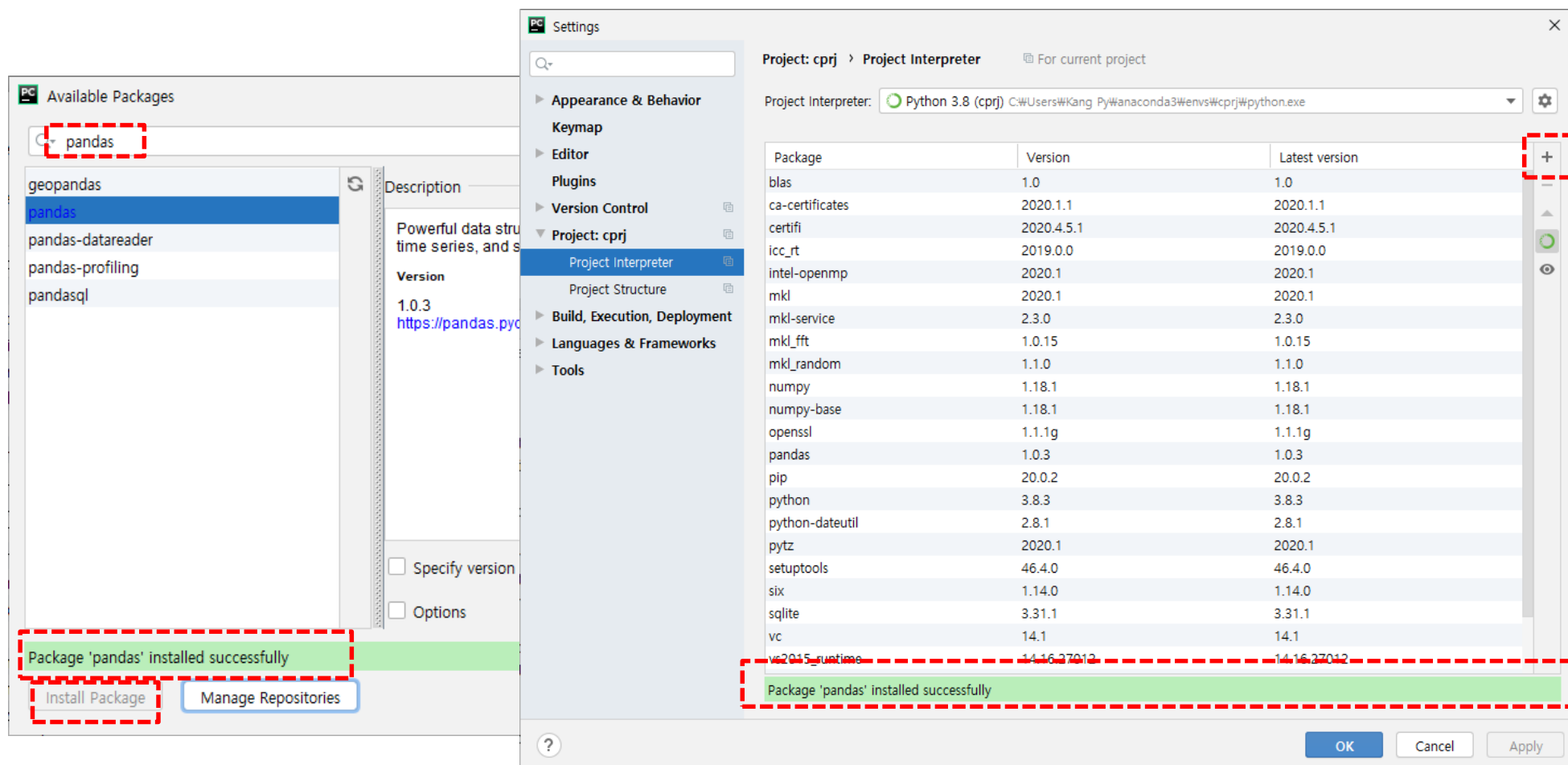
가상환경의 설치 모듈 확인

파이썬 가상환경 Python language

- File | Settings ...
 - Ctrl + alt + s



- 프로젝트 conda prj의 Settings에서 pandas 설치
 - 하단 버튼 Install Package



numpy, pandas, matplotlib 설치 관계

파이썬 가상환경 Python language

- Pandas 설치하면 numpy도 함께 설치
 - 터미널에서 pip 실행
 - `pip install pandas`

```
(pipenv prj) D:\pipenv prj>pip install pandas
Collecting pandas
  Downloading https://files.pythonhosted.org/packages/78/b9/a304328ea14cd172a5cf681b634b99e24a5b4e24de83204b713b088f02d5/pandas-0.25.3-cp38-cp38-win32.w
```

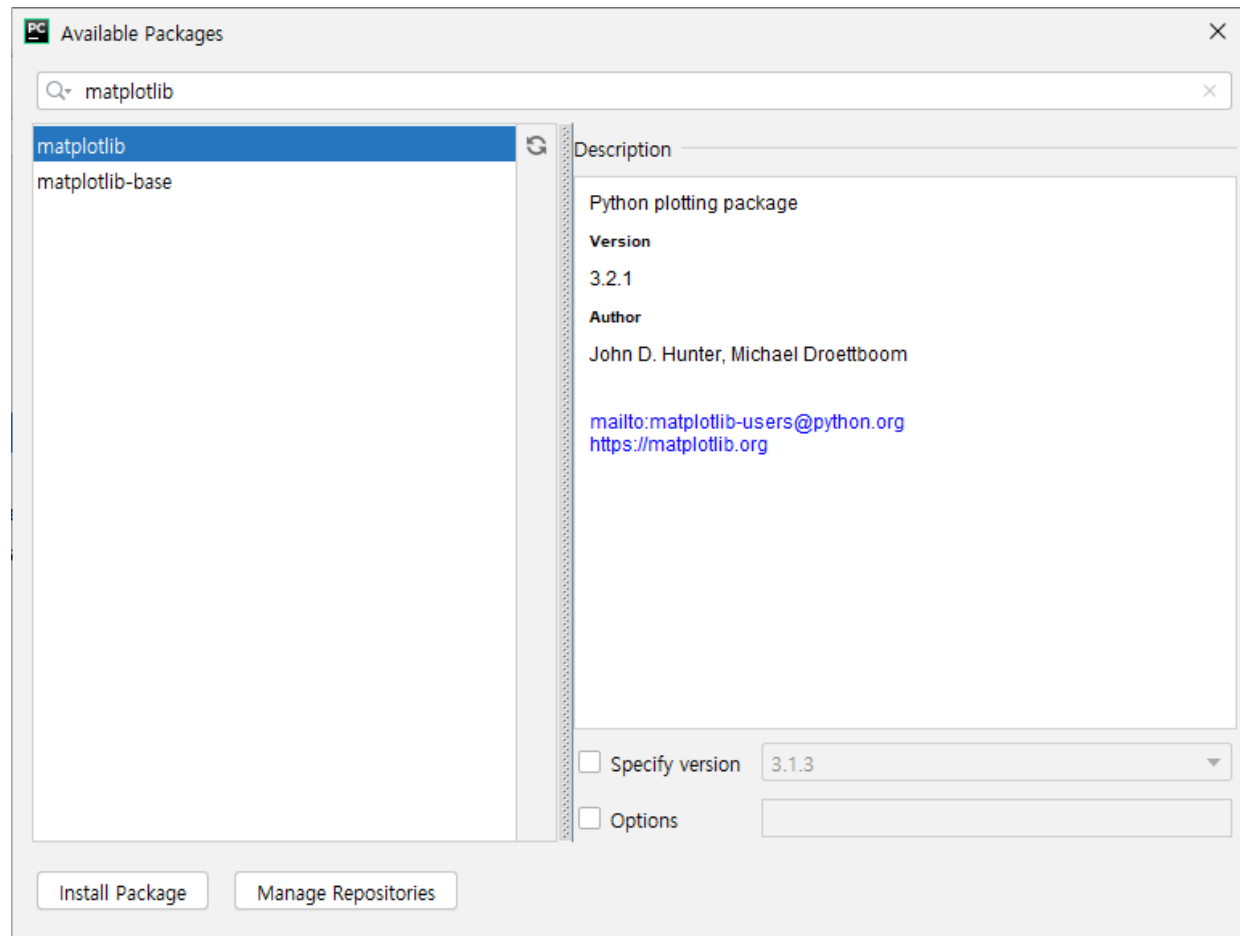
- 반대로 numpy도 pandas와 함께 설치

- Matplotlib를 설치해도 numpy 설치
 - 그 반대는 안됨
- Matplotlib와 pandas는 각각 설치 필요

모듈 matplotlib 설치

파이썬 가상환경 Python language

- Settings의 project interpreter에서 설치
 - 모듈 matplotlib를 settings에서 설치 전

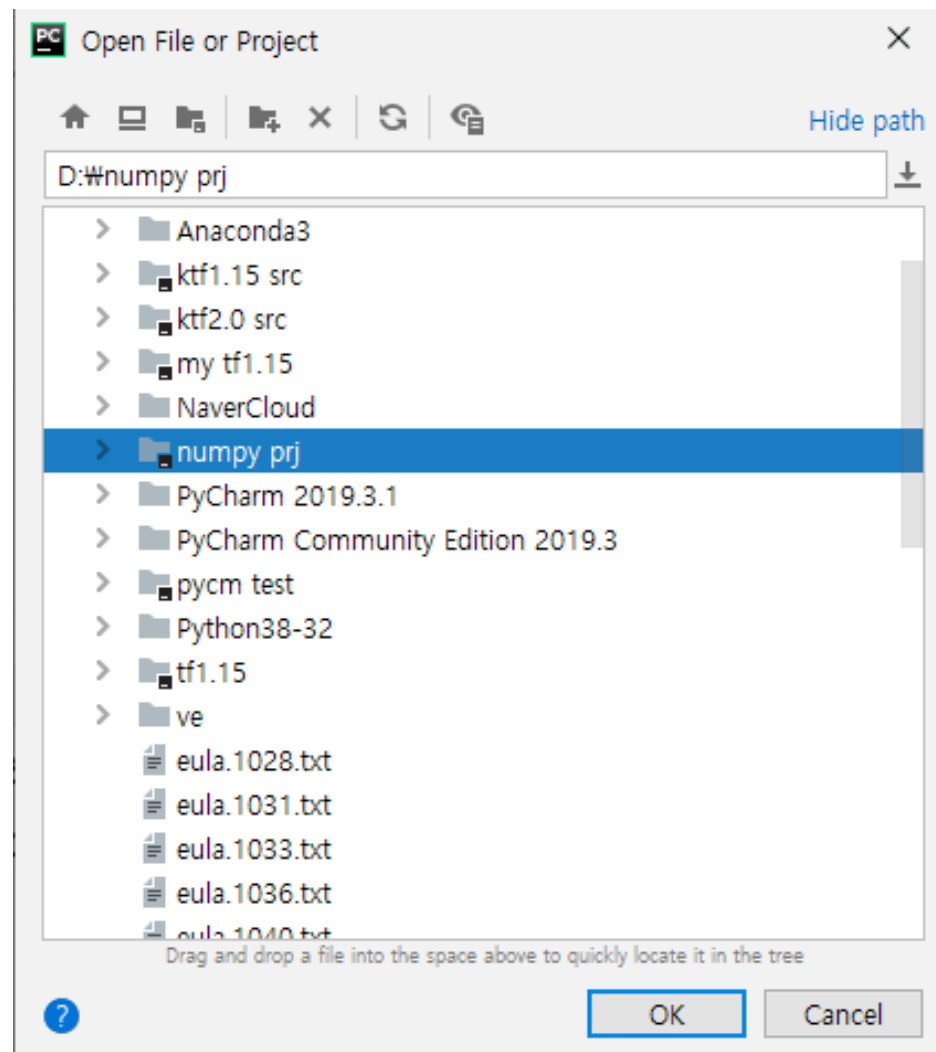


파이썬 프로젝트 열기와
모듈 설치 시
프로젝트 Terminal 활용과
Settings에서 설치

기존 프로젝트 열기

파이썬 가상환경 Python language

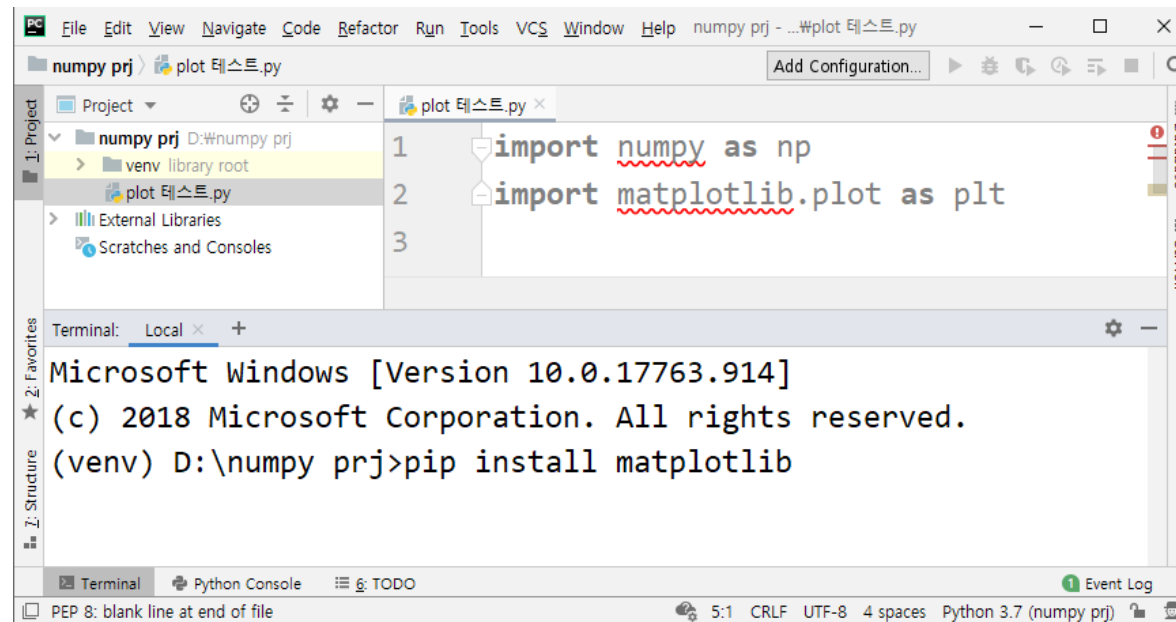
- Open ...



필요한 모듈 설치 방법

파이썬 가상환경 Python language

- 방법 2가지
 - 직접 터미널 창에서 pip 또는 conda로 설치
 - 하단 왼쪽 Terminal 클릭



- Settings...(ctrl + alt + s)에서 설치
 - 메뉴 file / settings...
 - Settings... 창의 왼쪽 메뉴에서 다음 선택
 - Project: 프로젝트_이름 / Project interpreter

터미널에서 matplotlib 설치 후 코딩 실행

파이썬 가상환경 Python language

- pip install matplotlib

> pip show matplotlib

설치 유무 확인 방법

The screenshot shows the PyCharm IDE interface. The main editor displays a Python script named 'plot 테스트.py' with the following code:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.random.random(10)
5 print(x)
6
7 plt.plot(x)
8 plt.show()
9
```

The SciView panel on the right shows a line plot of the generated data. The x-axis ranges from 0 to 5, and the y-axis ranges from 0.2 to 1.0. The plot shows a blue line with several peaks and valleys.

The Terminal panel at the bottom shows the output of the command 'pip install matplotlib' in a virtual environment:

```
(c) 2018 Microsoft Corporation. All rights reserved.
(venv) D:\numpy prj>pip install matplotlib
Collecting matplotlib
  Using cached https://files.pythonhosted.org/packages/dd/73/d
c25ca27a9960539ef984921b0d42368445b856ae0861c3acba542b9a39c/ma
```

현 가상환경에서 jupyter 실행

파이썬 가상환경 Python language

- **현 가상환경에서 확인**
 - Jupyter가 있는 폴더가 현재 가상환경 확인
 - **아니면**
 - **Where jupyter**
 - 또는 conda list jupyter로 확인
 - **없다면 설치 필요**
- **설치**
 - 명령어나 파이참에서 jupyter 설치