Lecture 01

Coding environment with Python

Dr. Heesuk Kim

Python 기반의 프로그래밍 환경

1 Installation using Python's Official homepage

2 Installation with Anaconda

Jupyter Notebook / Pycharm / Spyder / Visual Studio Code etc.

Python 공식 홈페이지를 활용하여 설치

- 파이썬 공식 홈페이지 http://www.python.org/downloads 에서 파이썬 설치 실행 파일을 다운로드 후 설치하는 방법

2021. 8. 29.
Python 3.9.6

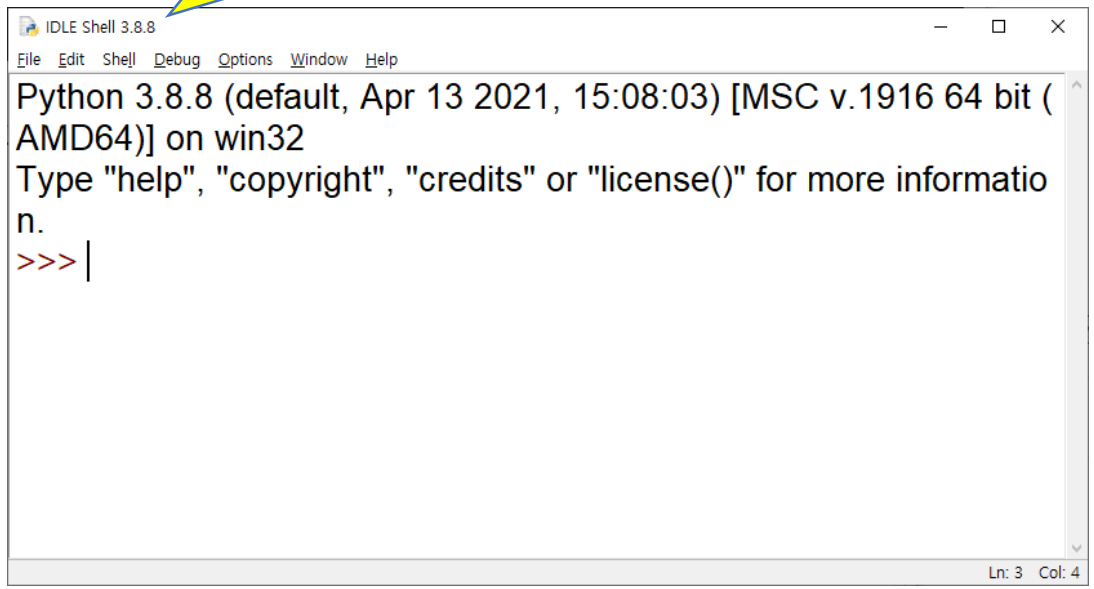
python-3.9.6-amd64.exe

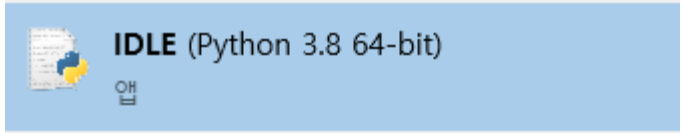
Python 공식 홈페이지를 활용하여 설치

- 파이썬 공식 홈페이지를 이용하여 파이썬을 설치한 경우, IDLE 환경에서 프로그래밍

파이썬 버전 확인

(참고) 캡처된 화면은 Python 3.8.8 기반으로 설치된 경우이다.





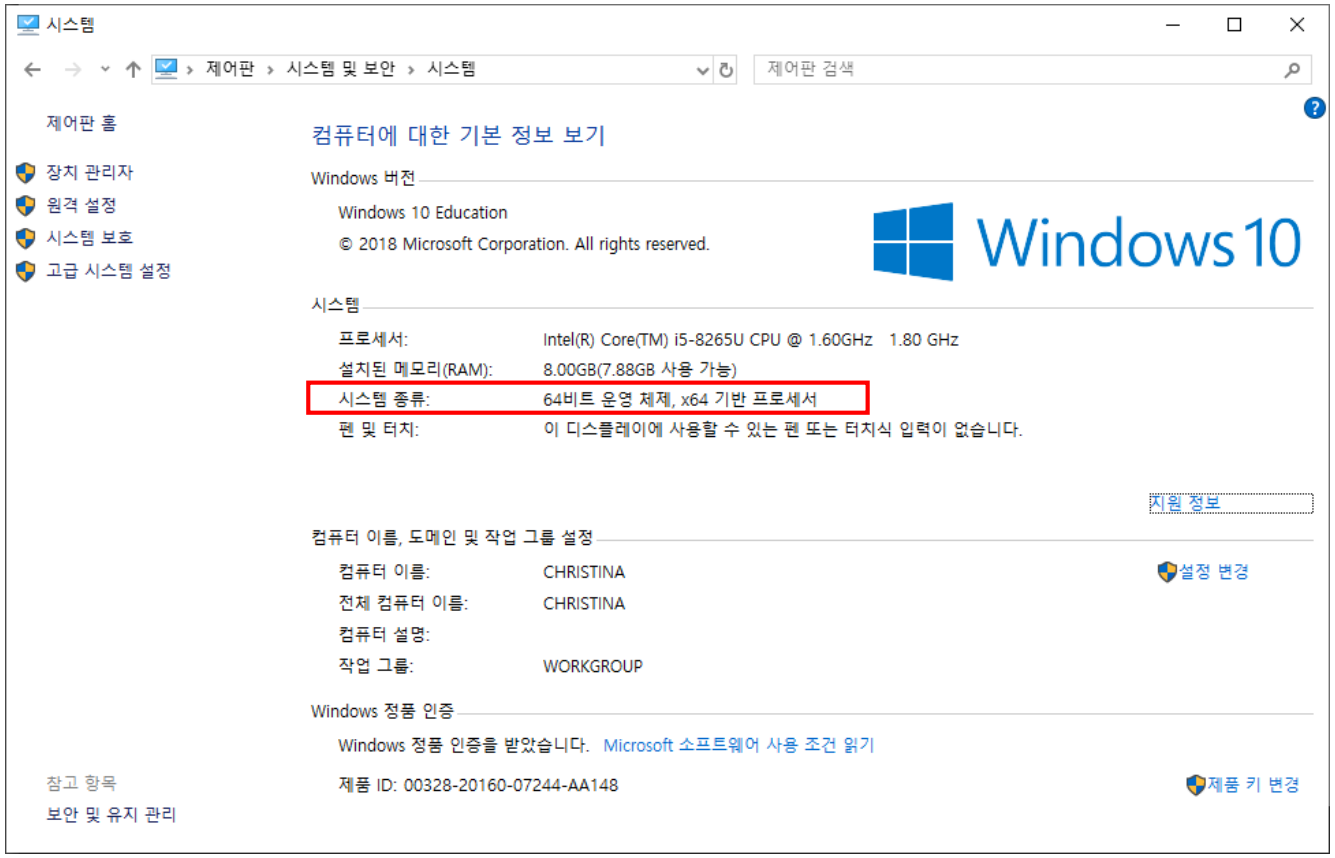
아나콘다를 활용하여 Python 설치

Anaconda

- 아나콘다는 컨티눔(Continuum)이라는 회사에서 만든 파이썬 배포판
- Anaconda는 패키지 관리 및 배포를 단순화하는 것을 목표로 함
- Big Data 분석 및 과학 컴퓨팅을 위한 Python과 R 프로그래밍 언어의 배포판
- 아나콘다 배포판은 1300만 명 이상의 사용자들이 사용함
- Anaconda를 설치하면 데이터 분석을 위한 기본 패키지가 내장되어 있어 편리하게 사용
- 윈도우, 리눅스, macOS에 적합한 1,400개 이상의 유명 데이터 과학 패키지가 포함
- Anaconda에는 conda라는 패키지 및 환경을 관리 할 수 있는 프로그램이 있으므로 편리
- Python의 용도가 데이터 분석 및 기계학습이라면 Anaconda 설치를 권장
- 여러 가지 파이썬 배포판 중 가장 후발주자이지만 가장 뛰어난 완성도로 현재는 사실상의 파이썬 표준 배포판으로 인정

아나콘다를 활용하여 Python 설치

Checklist before installing Anaconda



아나콘다를 활용하여 Python 설치

- 아나콘다 공식 홈페이지 https://www.Anaconda.com/products/individual 에서 다운로드 후 설치

1

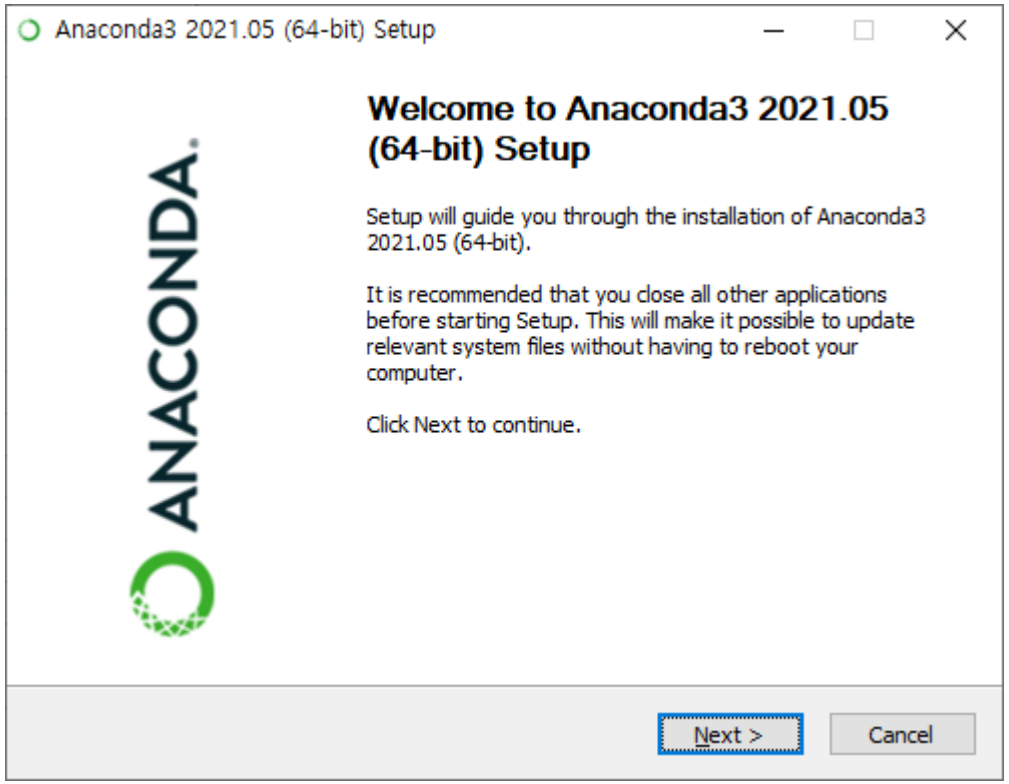
2021. 8. 29 버전

Anaconda3-2021.05-Windows-x86_64.exe

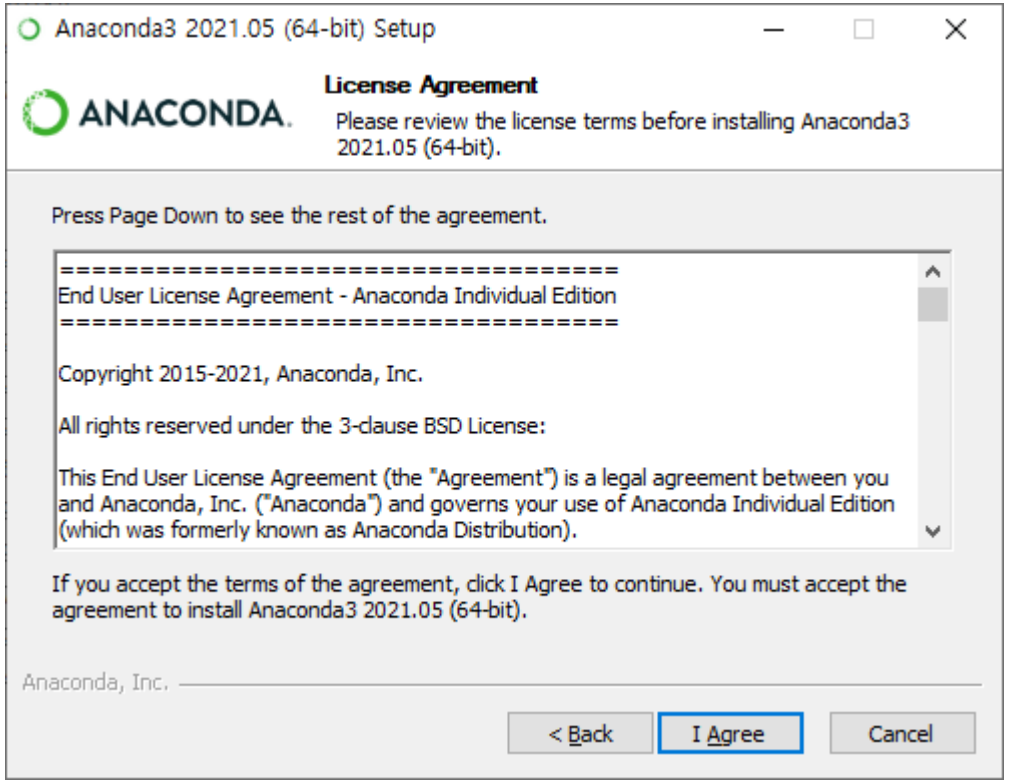
다운 받은 Anaconda 파일을 더블클릭하여 설치합니다.

아나콘다를 활용하여 Python 설치

2

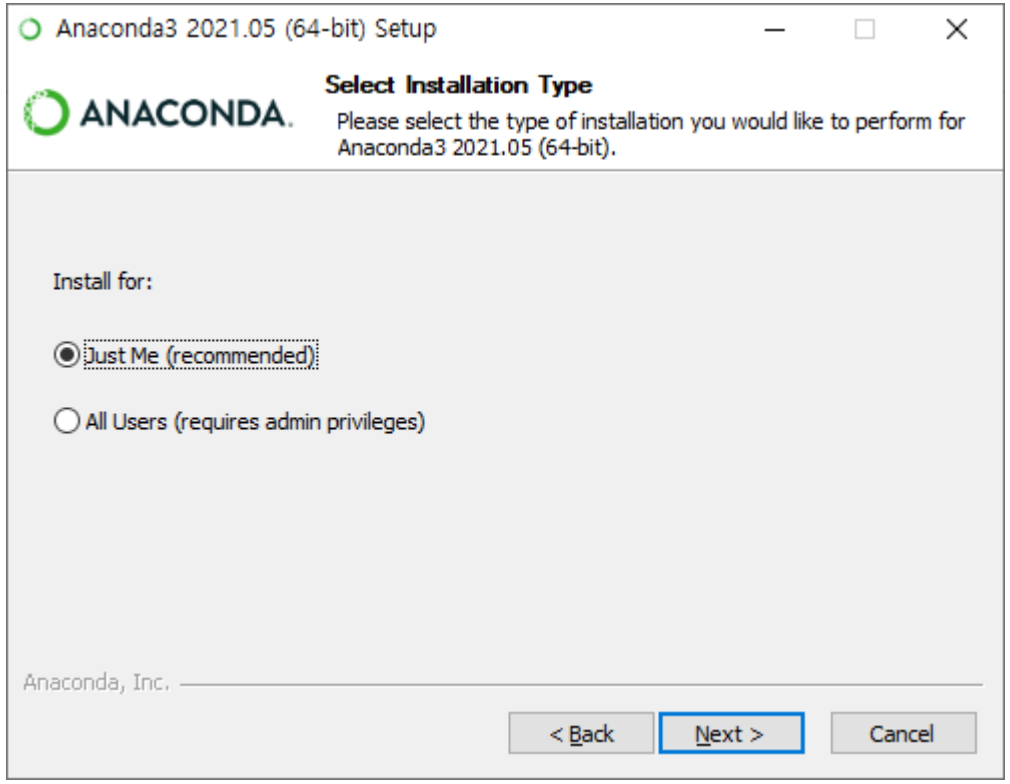


3

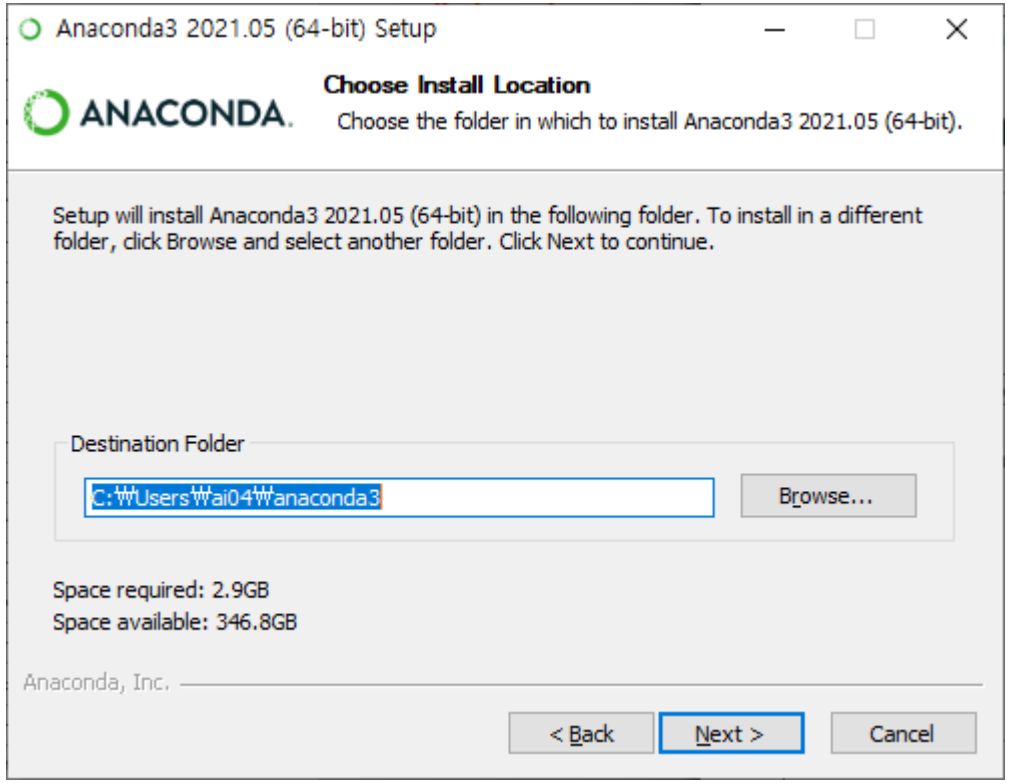


아나콘다를 활용하여 Python 설치

4



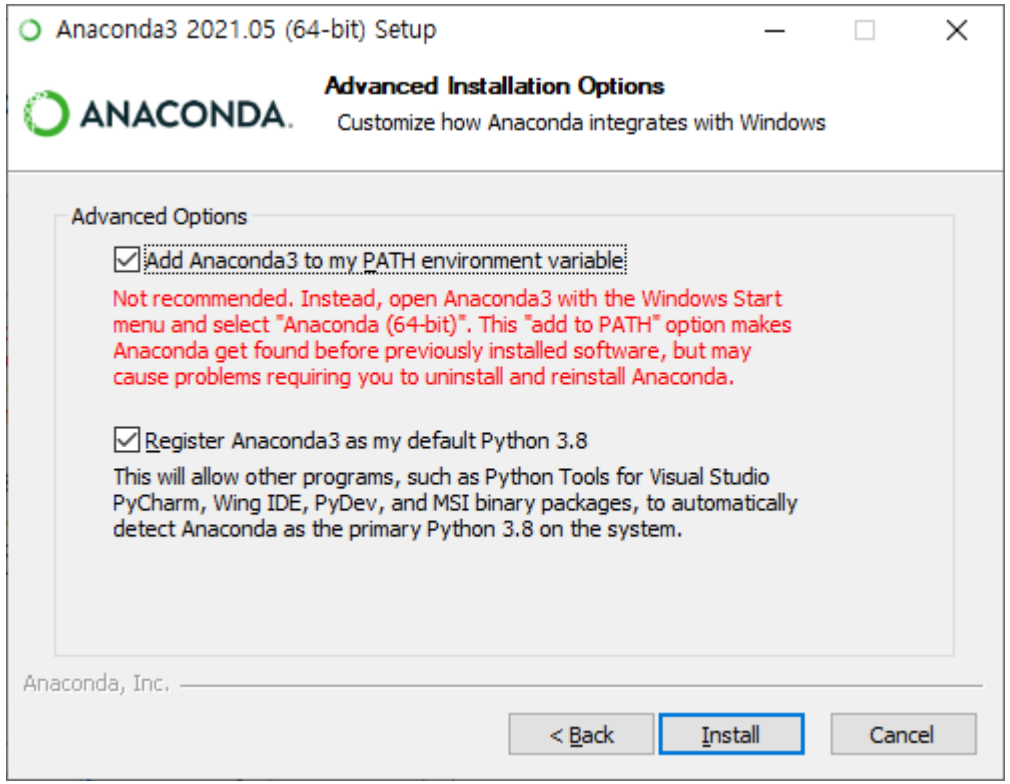
5

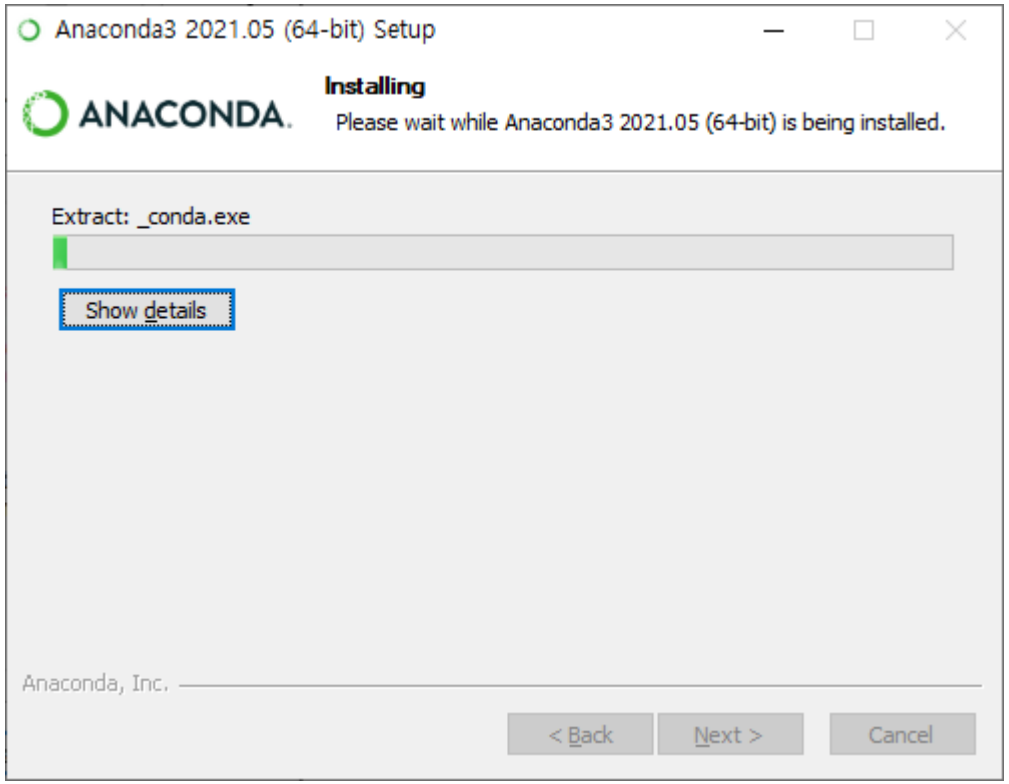


All Users를 선택할 경우 Just ME와 설치되는 경로가 다르며 패키지 설치, 삭제, 업그레이드 시 CMD창을 관리자 권한으로 열어 실행해야 하는 경우가 있습니다.

아나콘다를 활용하여 Python 설치

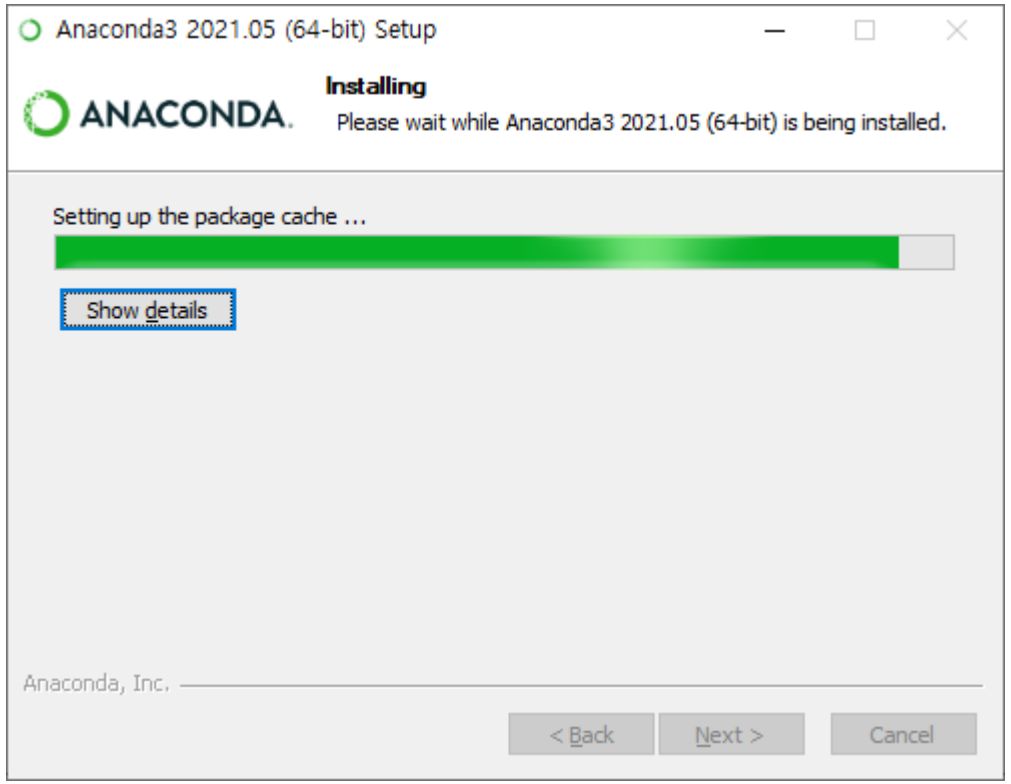
6

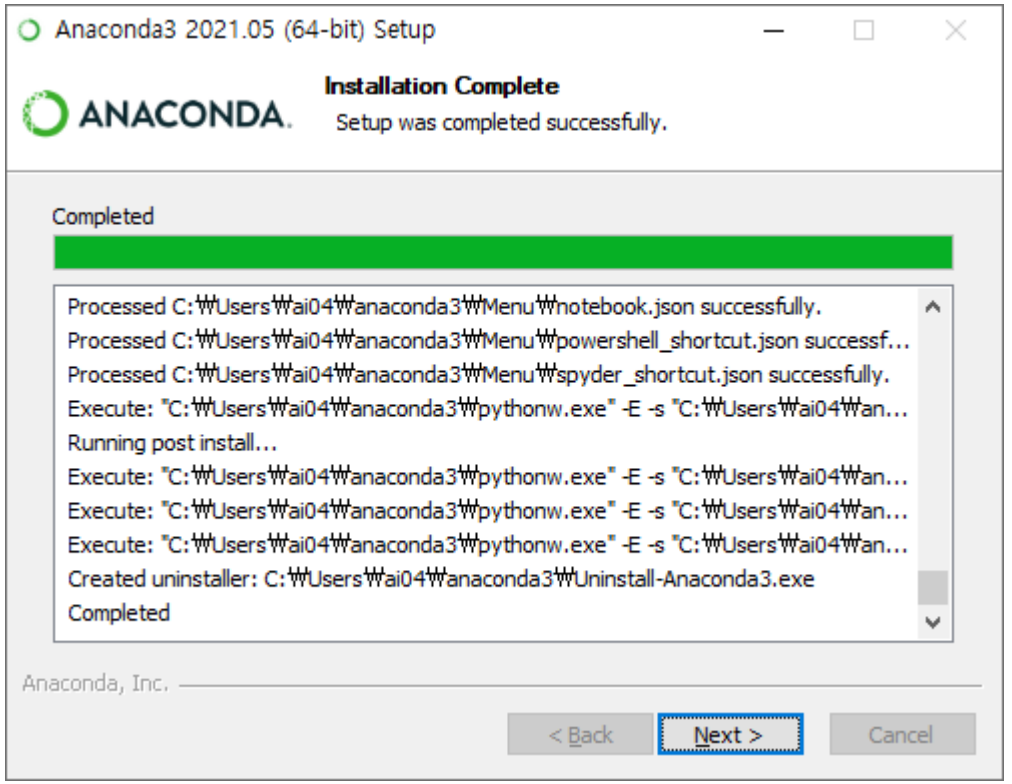




아나콘다를 활용하여 Python 설치

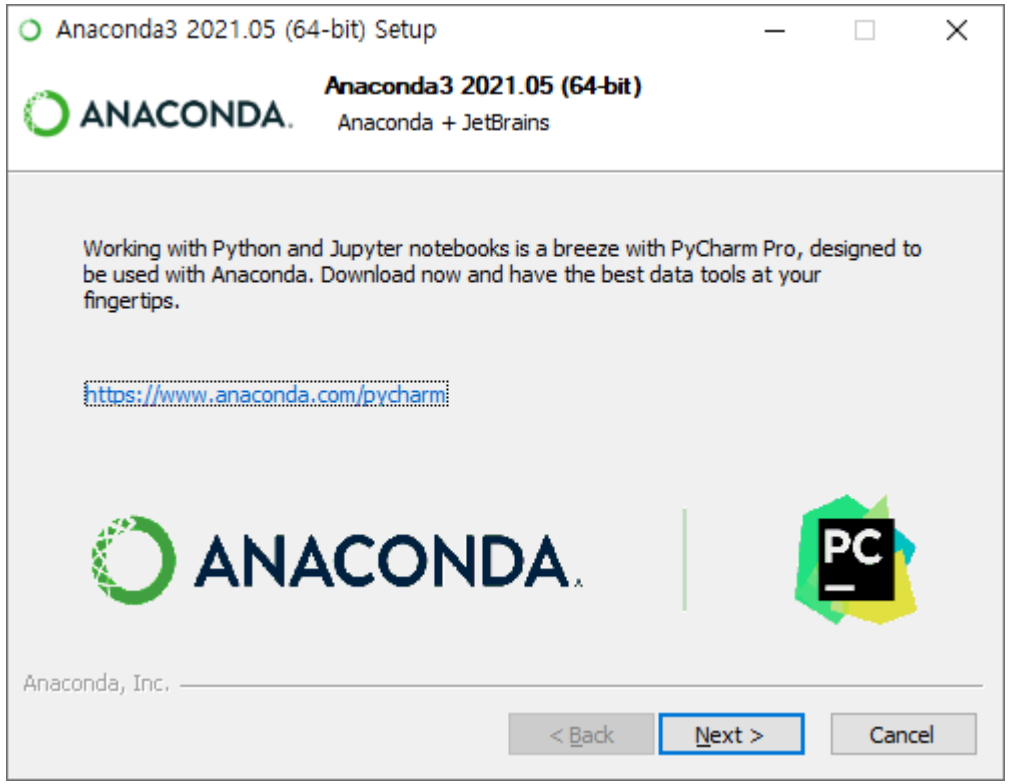
7



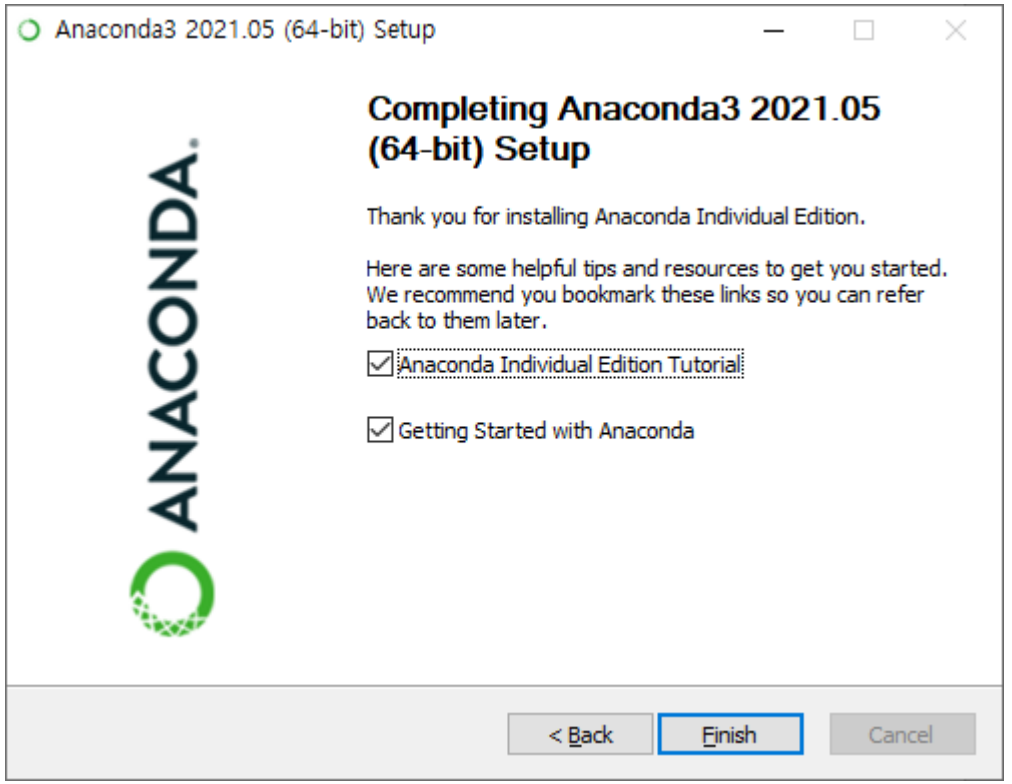


아나콘다를 활용하여 Python 설치

8

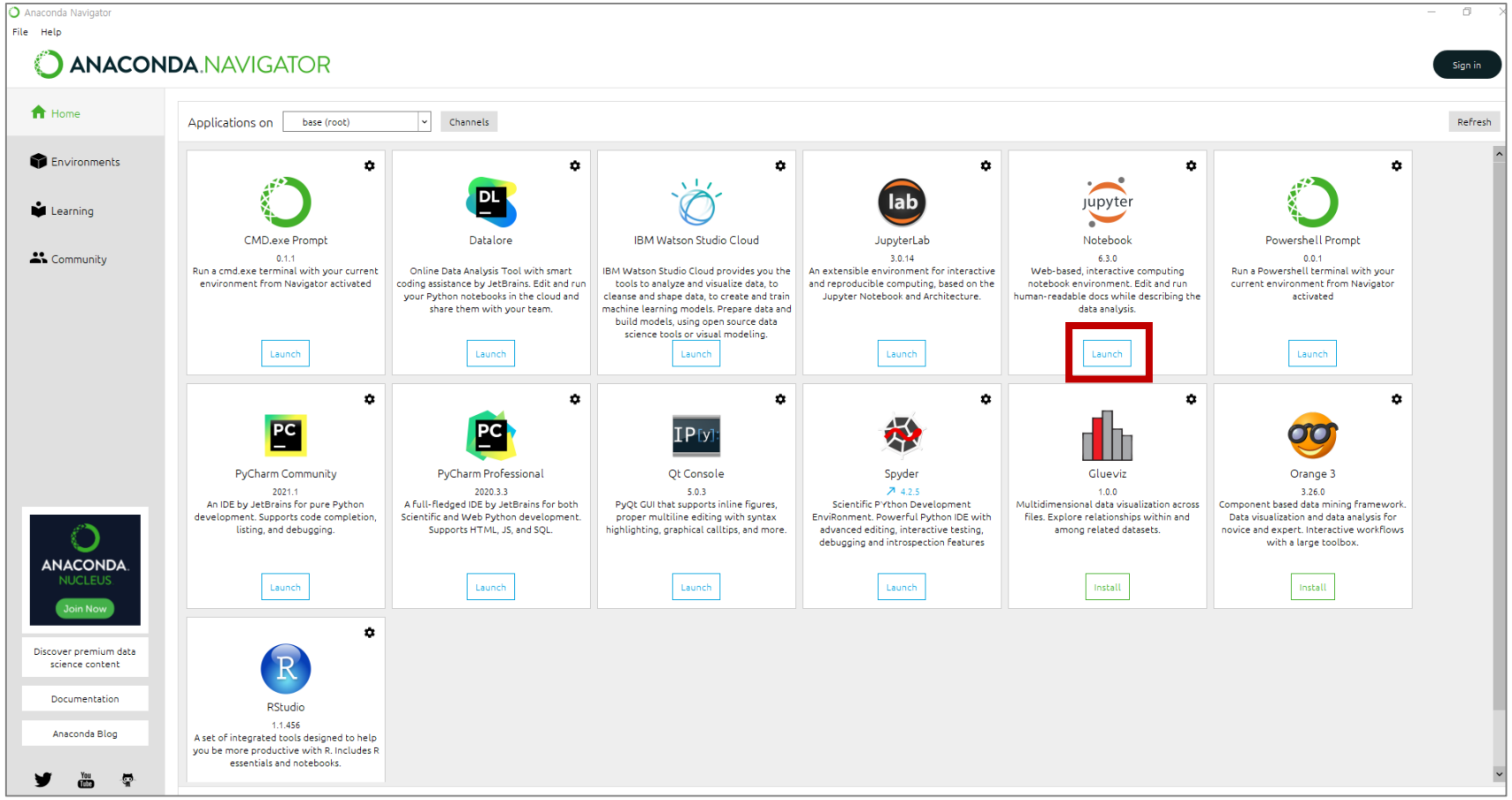


9



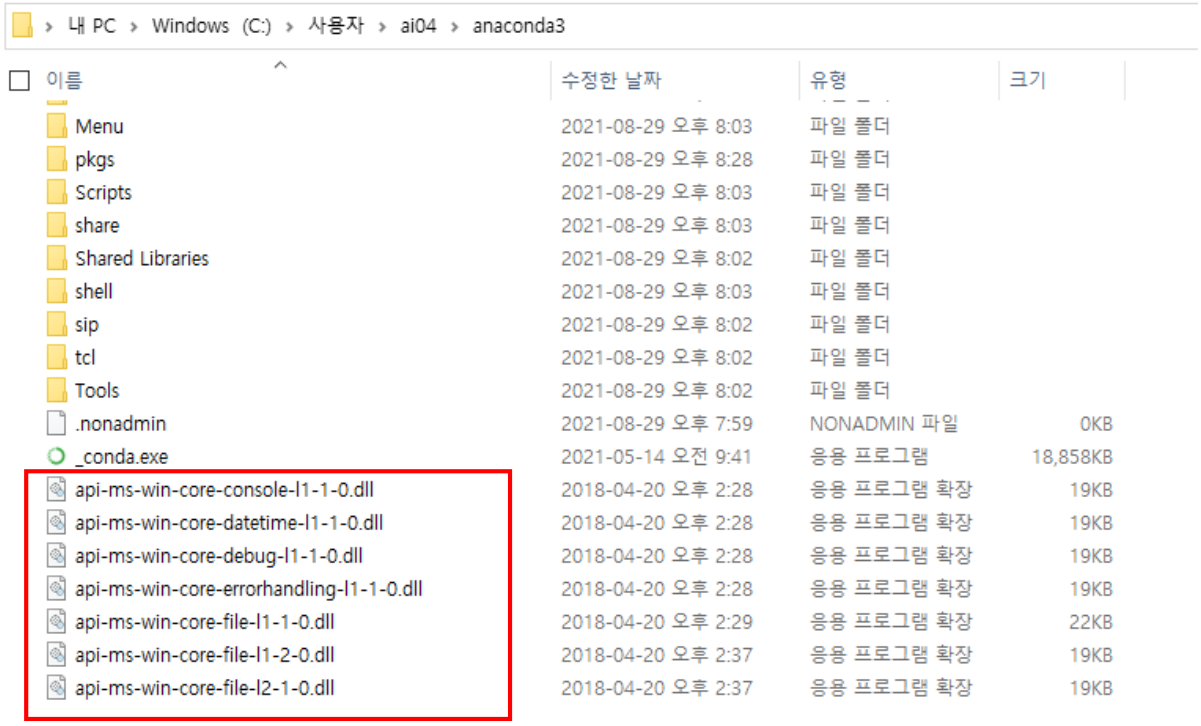
설치된 아나콘다 환경 확인

Anacodna Navigator를 실행



설치된 아나콘다 환경 확인

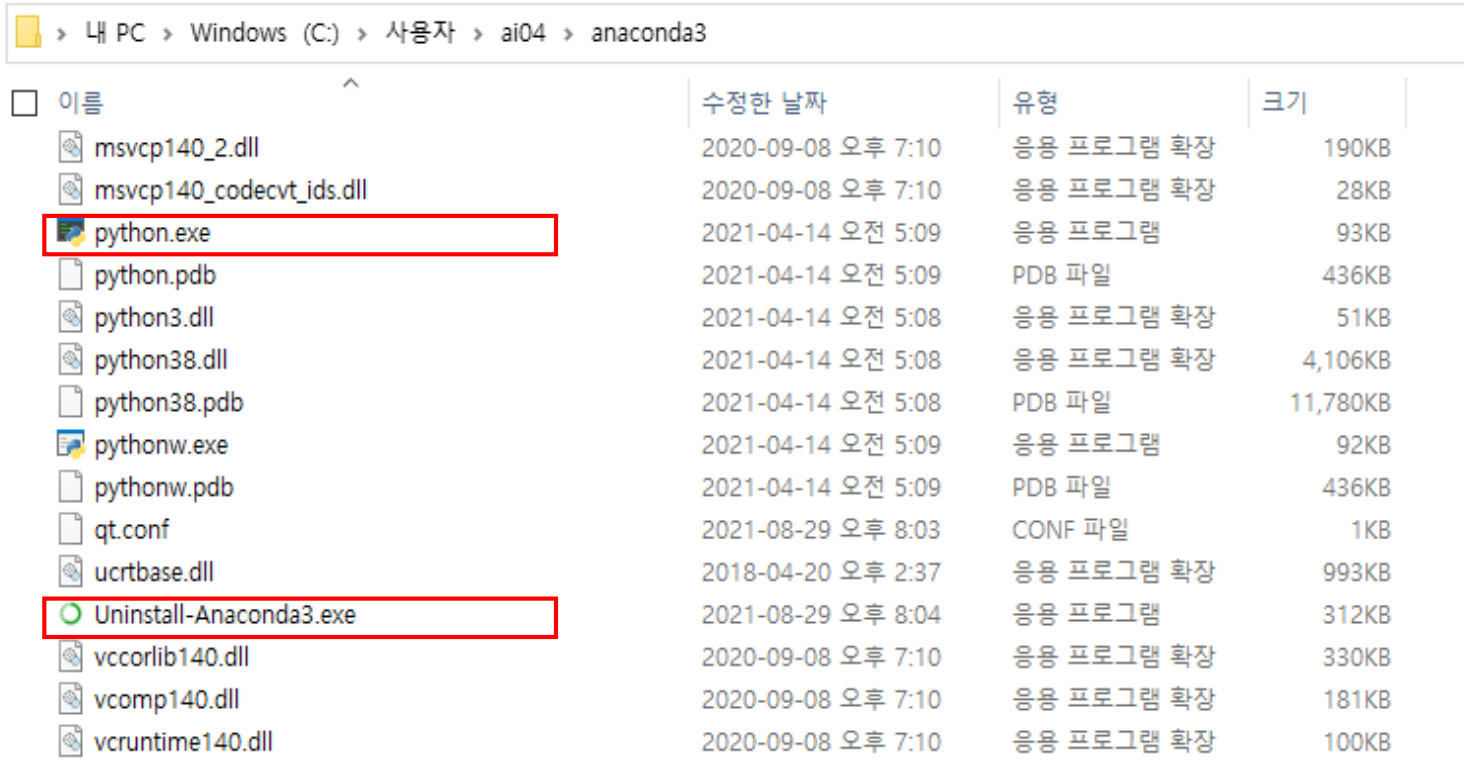
- **DLL(Dynamic Link Library)**
  - 동적 링크라고 하며 실행 파일에서 해당 라이브러리의 기능을 사용시 라이브러리 파일을 참조하여 (혹은 다운로드받아) 기능을 호출한다.
  - 정적 링크와는 다르게 컴파일 시점에 실행 파일에 함수를 복사하지 않고, 함수의 위치 정보만 갖고 그 함수를 호출할 수 있게 한다.



설치된 아나콘다 환경 확인

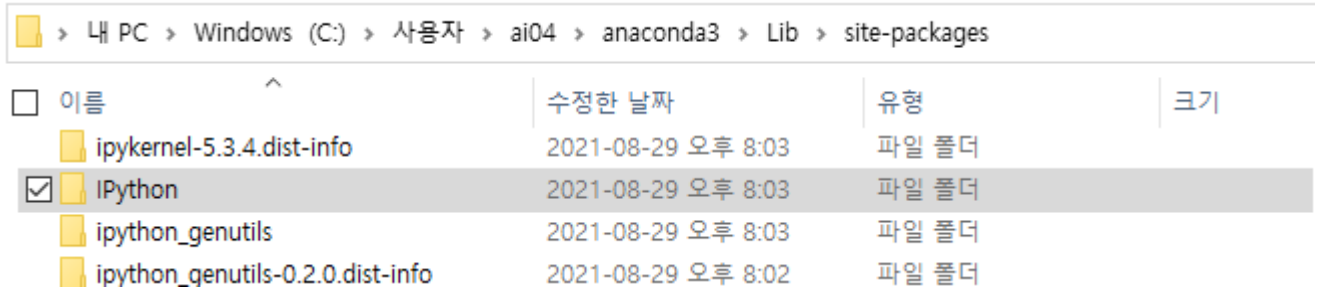
- **python.exe**

- **Uninstall-Anaconda3.**

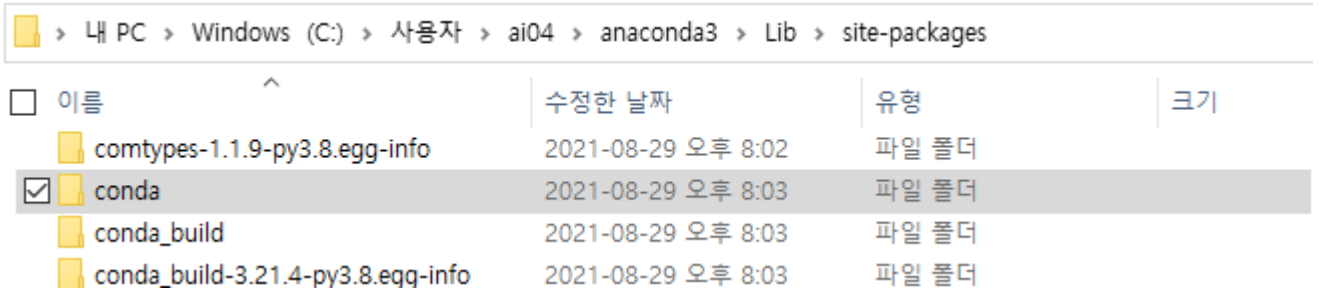


설치된 아나콘다 환경 확인

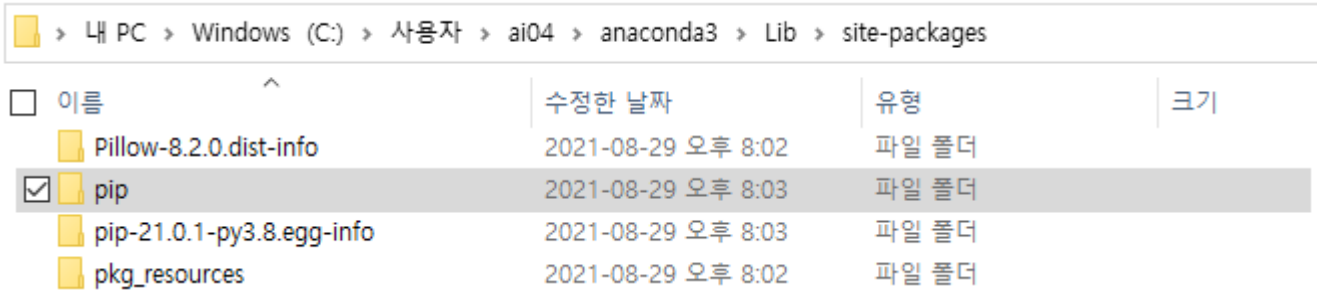
- **ipython**



- **conda**

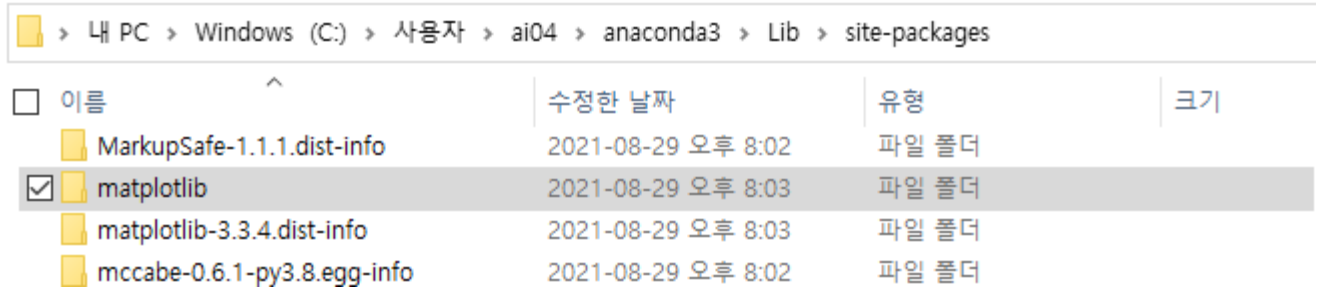


- **pip**

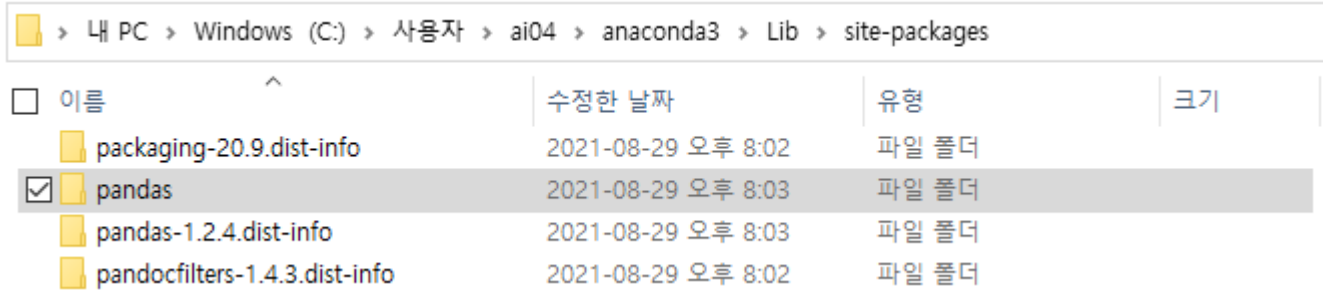


설치된 아나콘다 환경 확인

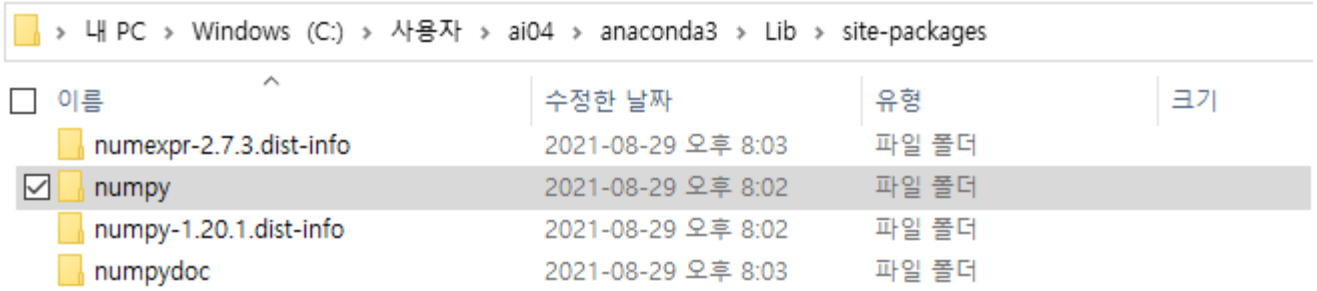
- **matplotlib**



- **pandas**

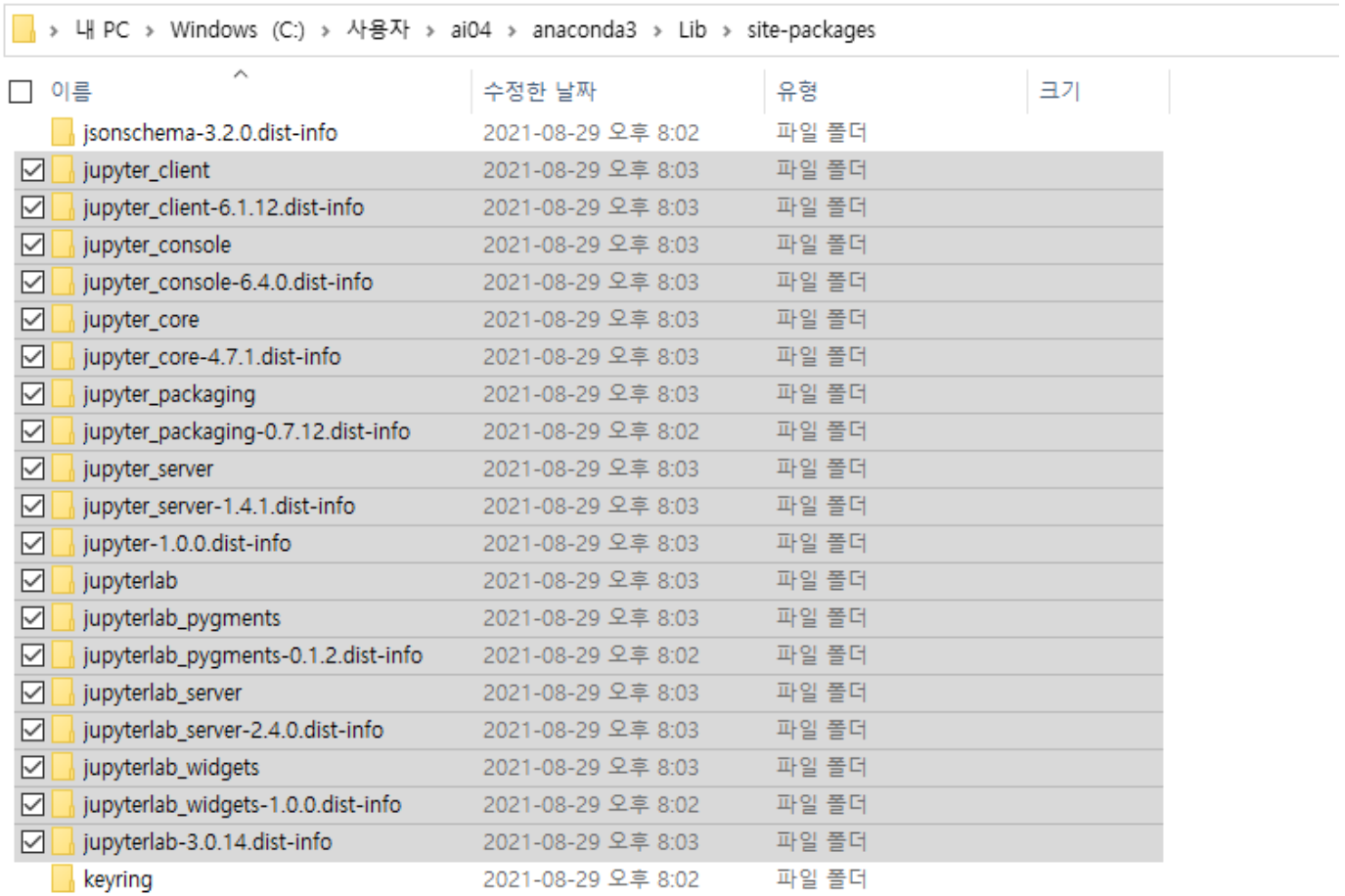


- **numpy**



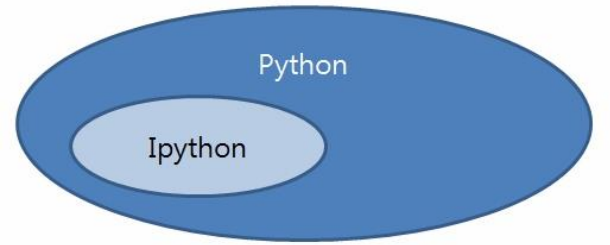
설치된 아나콘다 환경 확인

- **jupyter**



주피터 노트북

- **Anaconda를 설치하면**
  - **Ipython, jupyter** 등은 기본적으로 설치되어 있다.



  - 파이썬 공식 홈페이지에서 파이썬을 다운받아서 설치한 경우는 아래와 같은 방식으로 직접 Ipython과 Jupyter를 설치해야 한다.

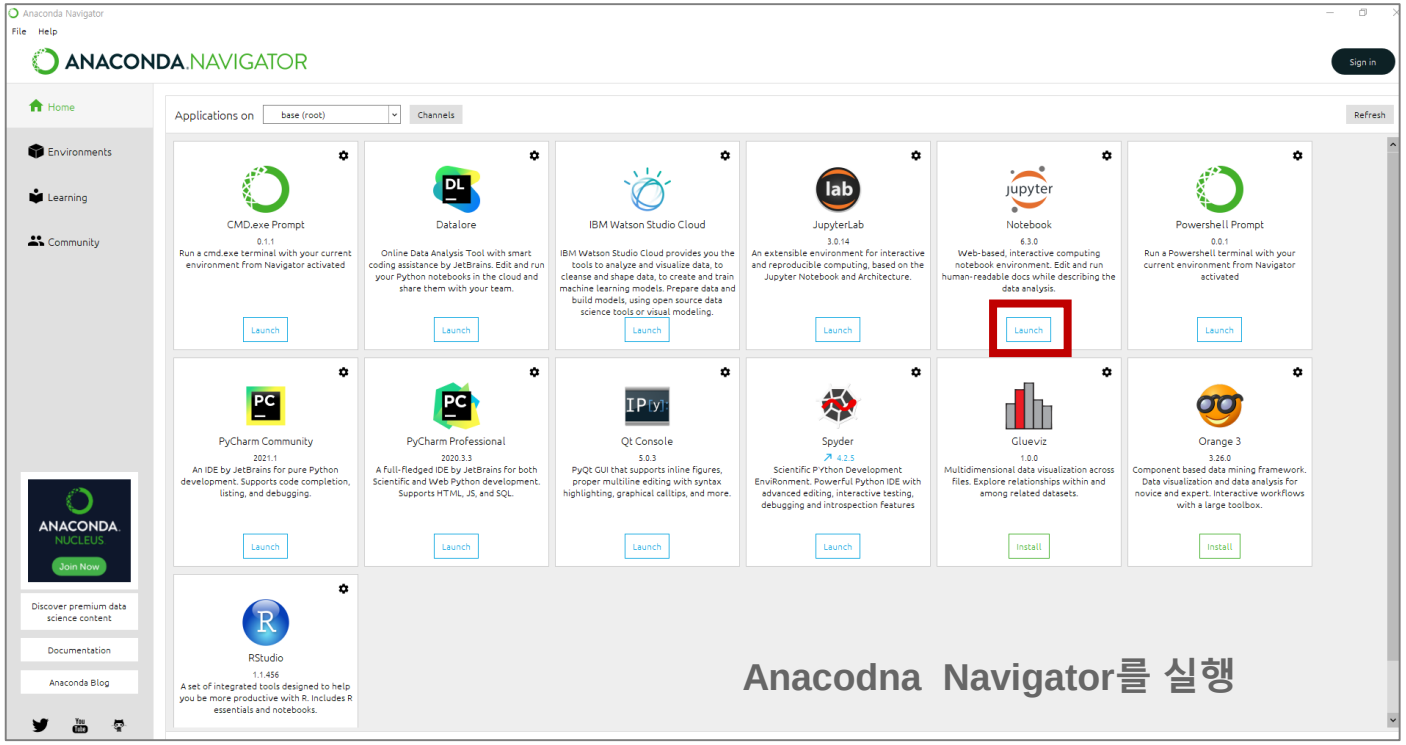
```
C:\test> pip install ipython

C:\test> pip install jupyter
```

주피터 노트북

주피터 노트북 실행하기

1 **Anaconda Navigator를 이용하여 Jupyter notebook을 실행하기**

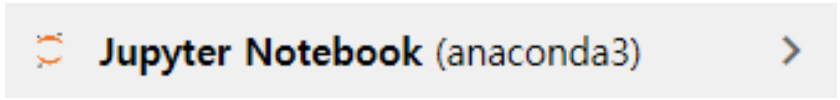


Anacodna Navigator를 실행

주피터 노트북

- **주피터 노트북 실행하기**

2 **Anaconda Navigator를 이용하지 않고, 바로 Jupyter notebook을 실행하기**



주피터 노트북

- **프로젝트 주피터 (Project Jupyter)**
  - 프로젝트 주피터(Project Jupyter)는 "오픈 소스 소프트웨어, 개방형 표준, 그리고 여러 개의 프로그래밍 언어에 걸쳐 인터랙티브 컴퓨팅을 위한 서비스 개발"을 위해 설립된 **비영리 단체**이다.
  - 프로젝트 주피터의 이름은 주피터가 지원하는 세 개의 핵심 언어인 Julia, Python 그리고 R에서 유래한다.
  - 2014년에 페르난도 페레즈(Fernando Perez)에 의해 IPython으로부터 파생된 프로젝트 주피터는 여러 개의 언어를 통한 실행 환경을 지원한다.
  - 프로젝트 주피터는 인터랙티브 컴퓨팅 제품인 주피터 노트북, 주피터허브, 그리고 주피터 노트북의 차세대 버전인 주피터랩을 개발하고 지원해왔다.

주피터 노트북

- **프로젝트 주피터 (Project Jupyter)**
  - Ipython
    - ✓ 2001년에 페르난도 페레즈(Fernando Perez)가 파이썬을 과학 플랫폼에 쉽게 사용할 수 있도록 command line 기반의 Interactive Python 인터페이스로 개발
  - Jupyter Notebook
    - ✓ 웹 브라우저에서 실행되며 코드, 텍스트, 수학식, 도표, 그래프, 대화형 그래픽 콘트롤러 등과 같은 인터페이스를 제공
    - ✓ 과학 분야 뿐만 아니라 교육, 소프트웨어 문서, 책 집필 등에도 널리 사용됨

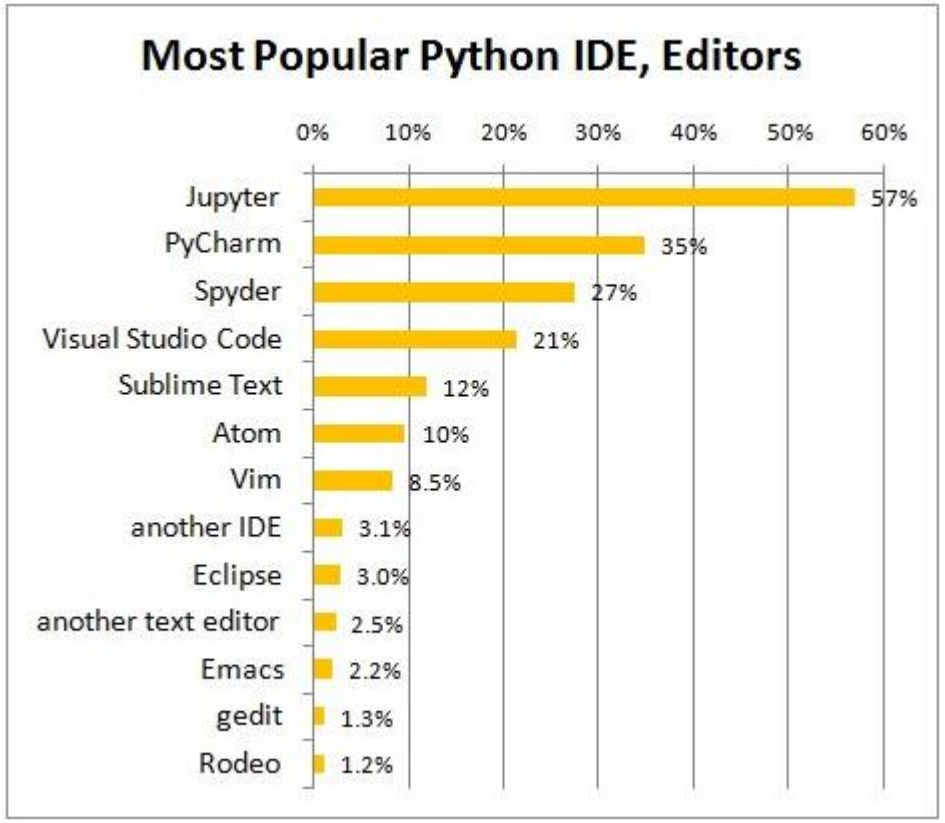
주피터 노트북

- **프로젝트 주피터 (Project Jupyter)**
  - **Jupyter Notebook**
    - ✓ 언어에 독립적으로 다양한 커널과 연동됨
    - ✓ Jupyter Notebook의 Python 커널 이름은 IPython
    - ✓ Jupyter Notebook의 R 커널의 이름은 IR
    - ✓ Julia는 IJulia

Jupyter Notebook 특징

- Coding 결과를 실시간으로 확인이 가능하다.
- 자동 완성 기능이 있다.
- 다양한 언어를 지원(R, Scala, Julia)한다.
- Markdown을 지원하여 문서화가 가능하다.
- Web 접근이 가능하면 접속 가능하다.
- 파이썬 코드를 저장하면 ipynb 파일이 생성됨
- 완성된 파일을 Download시 다양한 포맷 (py, pdf, html etc.) 으로 저장 가능

- **An integrated development environment**

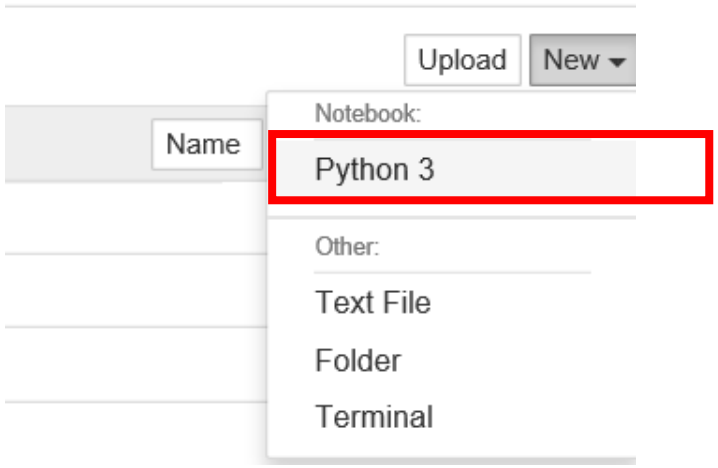


이미지 출처 https://www.kdnuggets.com/2018/12/most-popular-python-ide-editor.html

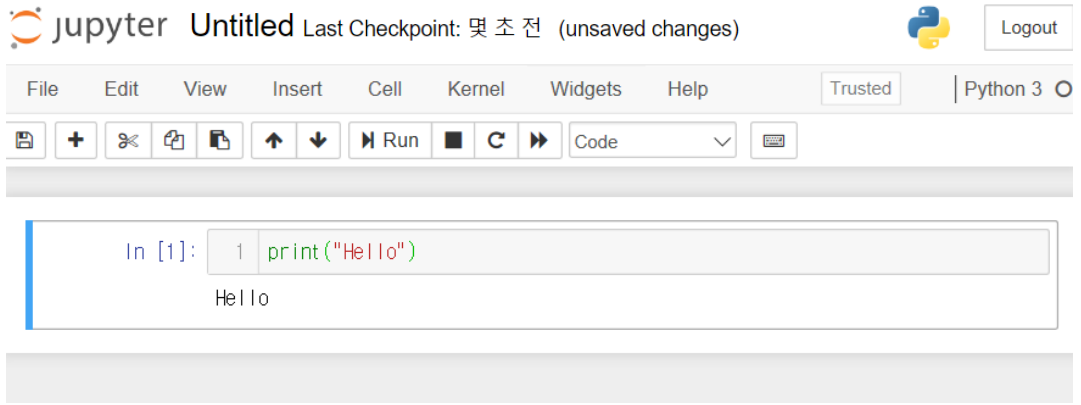
주피터 노트북 기본 사용법

- **Jupyter Notebook을 이용한 코드 입력 테스트**

1



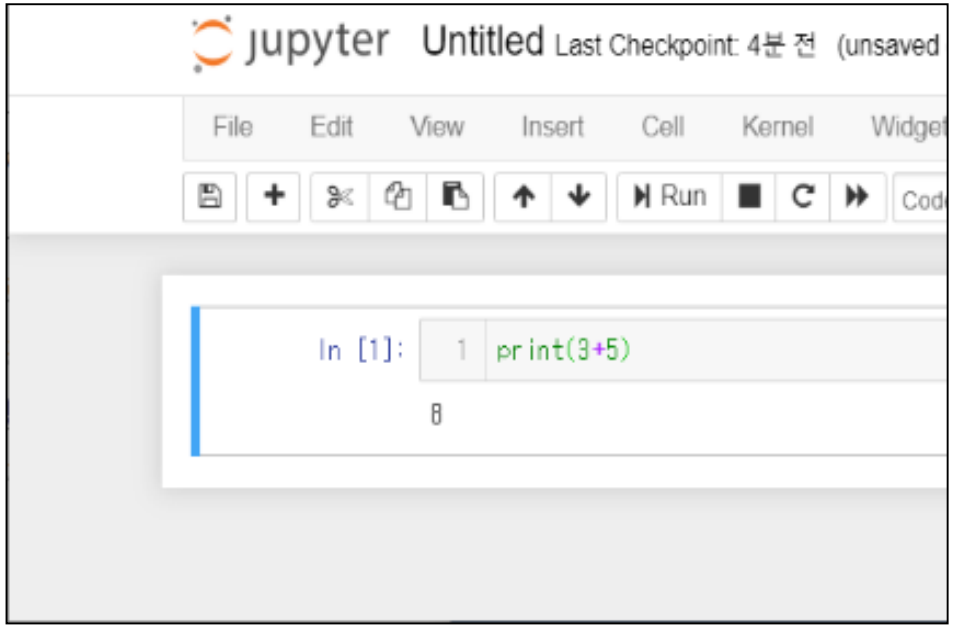
2



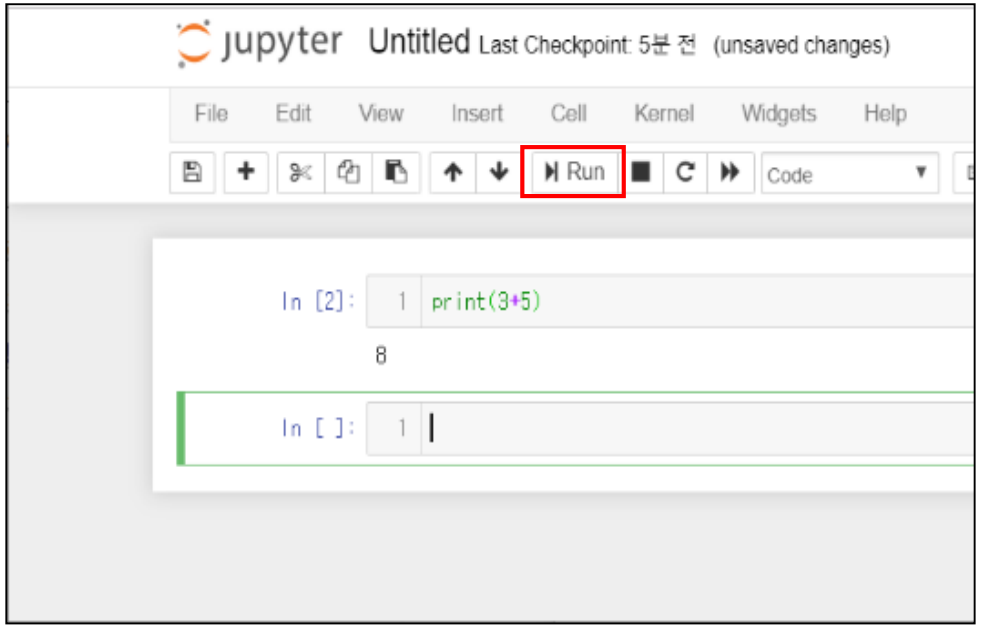
주피터 노트북 기본 사용법

3 실행하기

[코드 입력 후 Ctrl + Enter 를 실행한 화면]

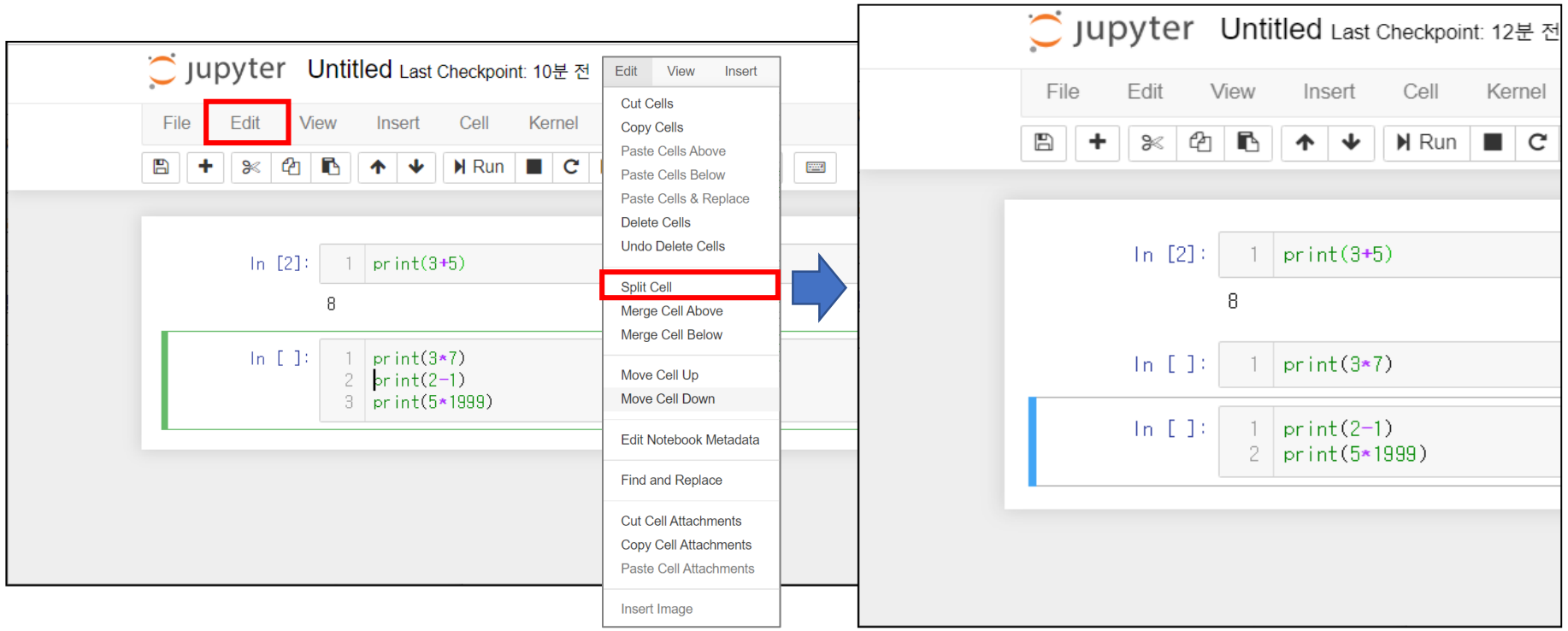


[코드 입력 후 Shift + Enter 를 실행한 화면]



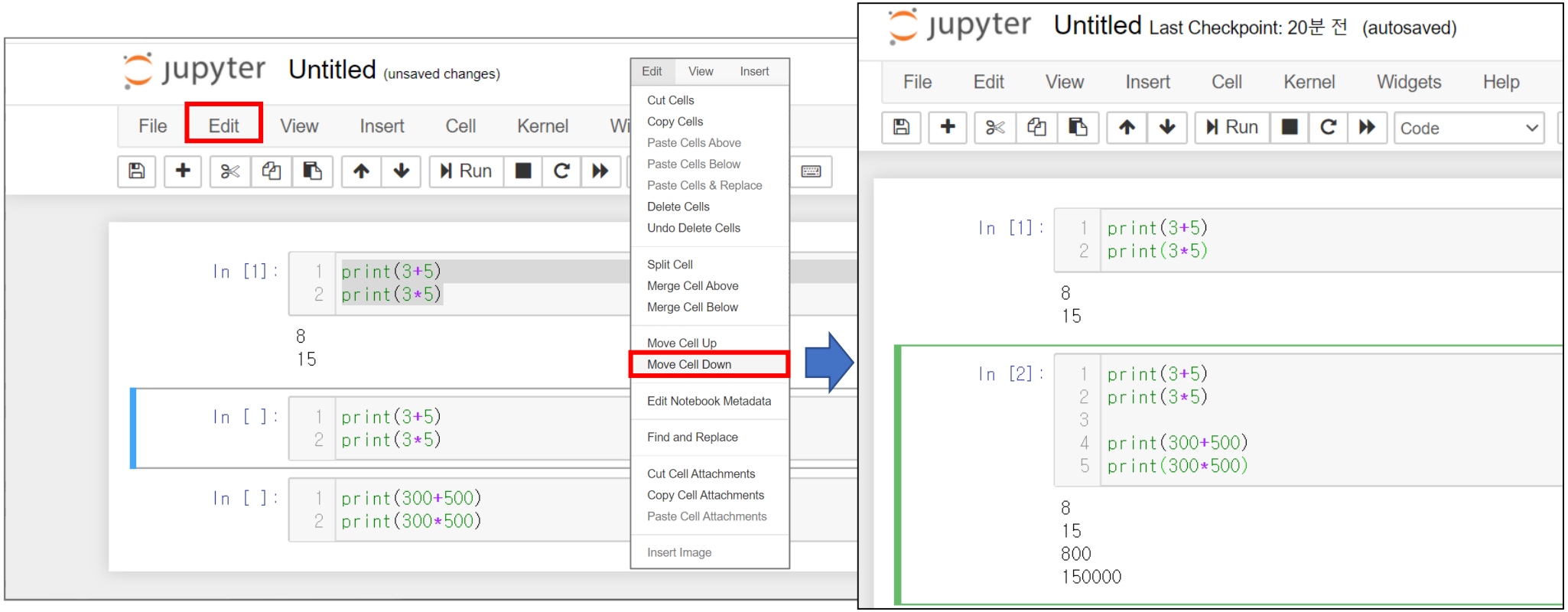
주피터 노트북 기본 사용법

4 cell 분리하고 합치기



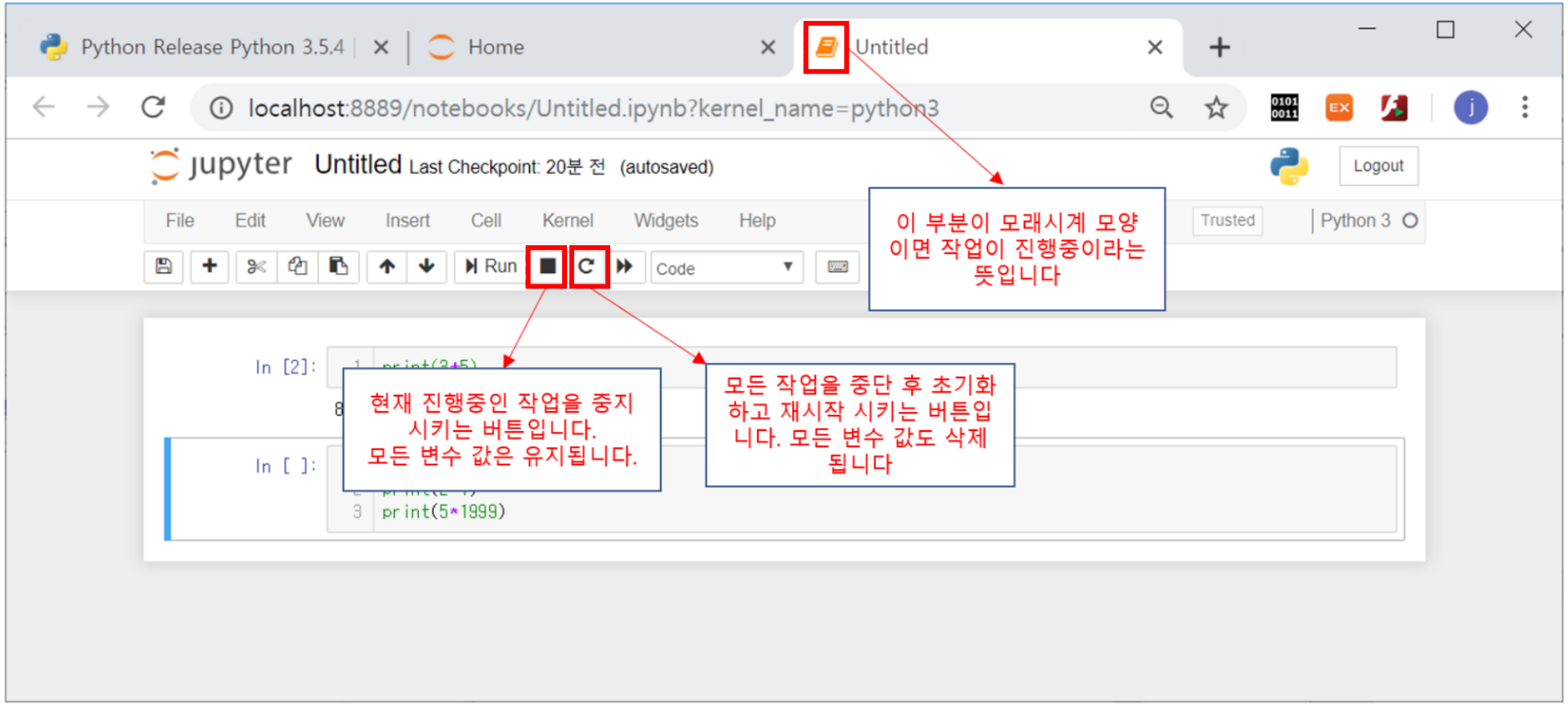
주피터 노트북 기본 사용법

5 cell 분리하고 합치기



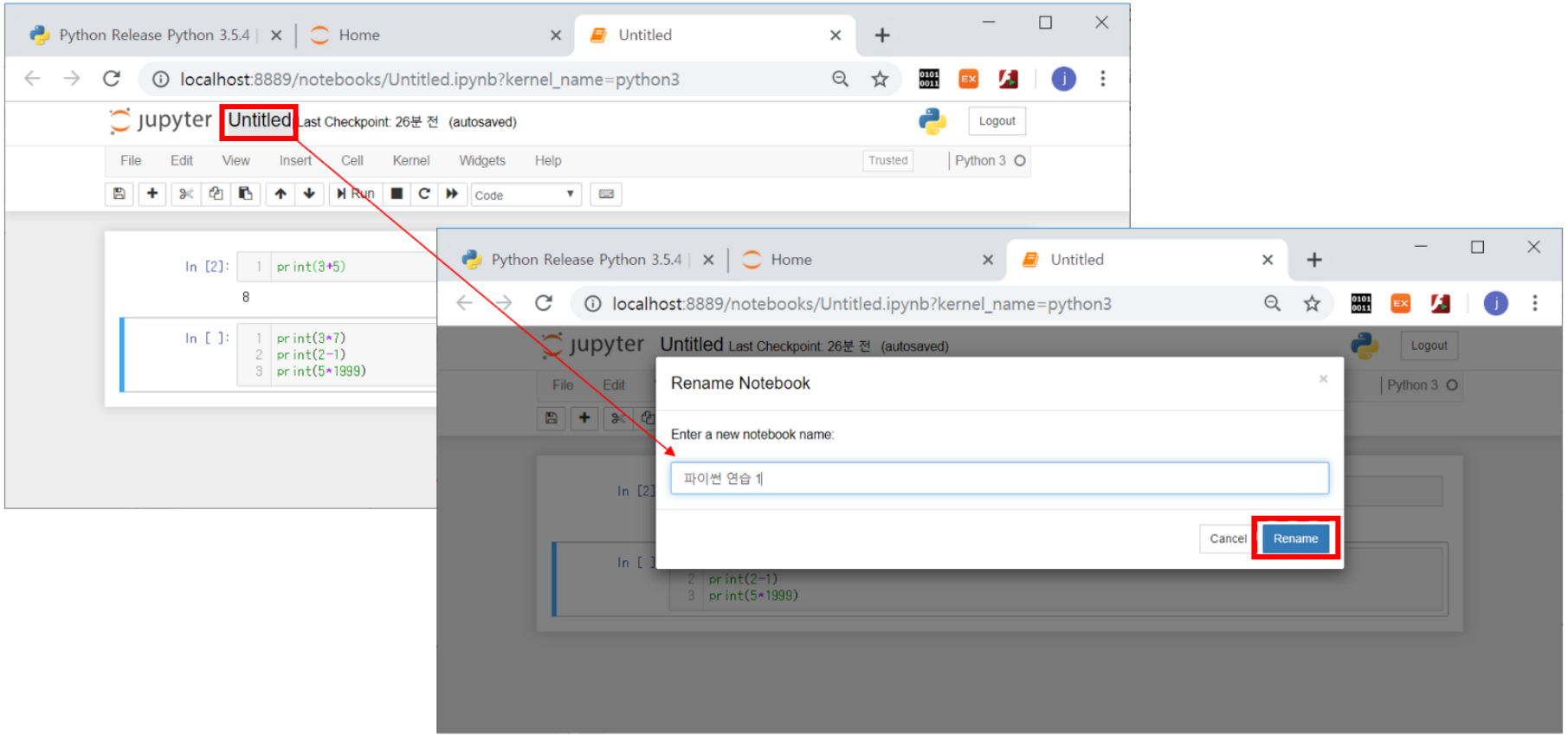
주피터 노트북 기본 사용법

6 진행중인 작업을 중단하거나 재시작 하기



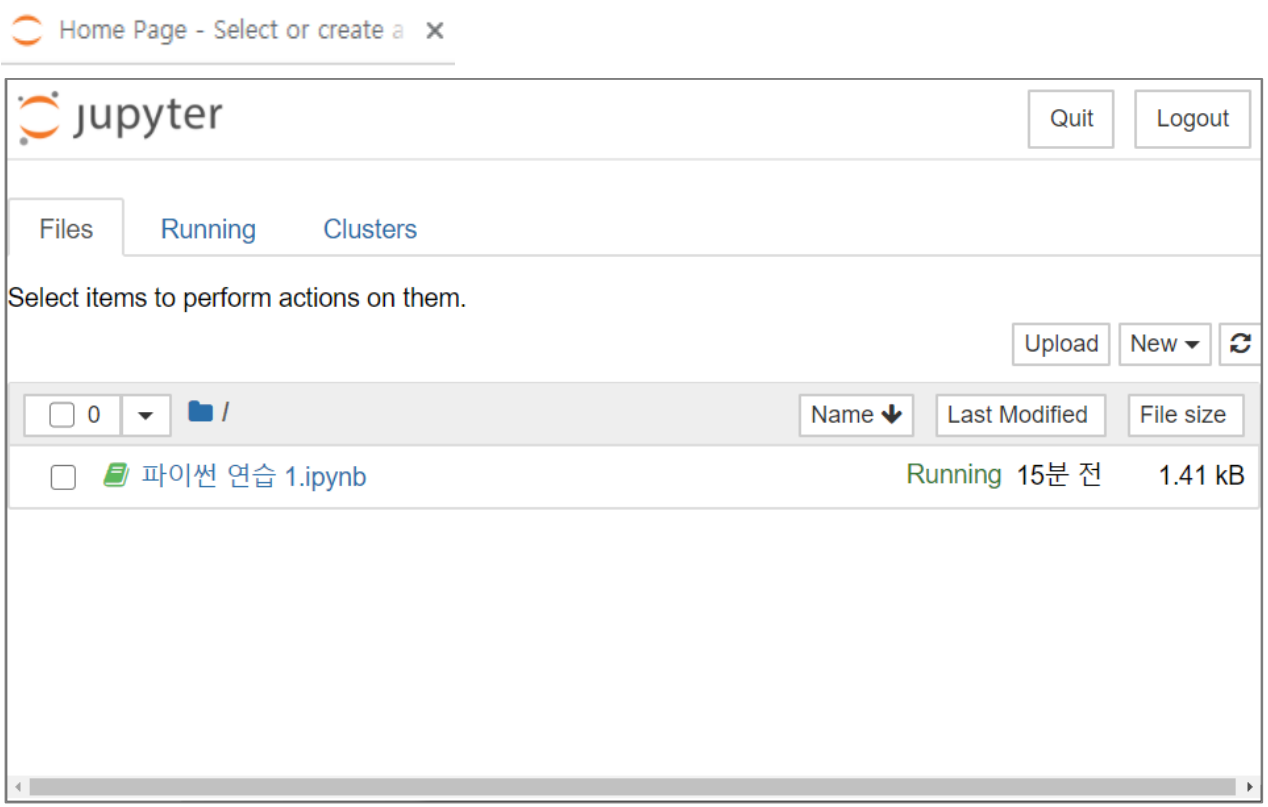
주피터 노트북 기본 사용법

7 작업 결과 저장하기



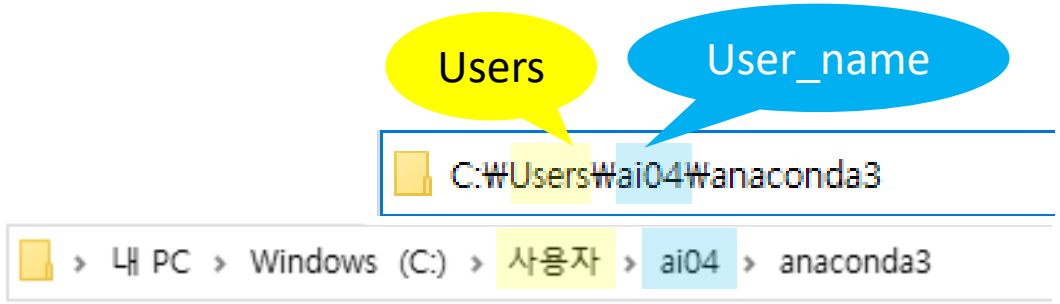
주피터 노트북 기본 사용법

8 저장된 작업 결과 확인하기 (홈 디렉토리 확인하기)



주피터 노트북 홈 디렉토리 변경

홈 디렉토리 확인하기



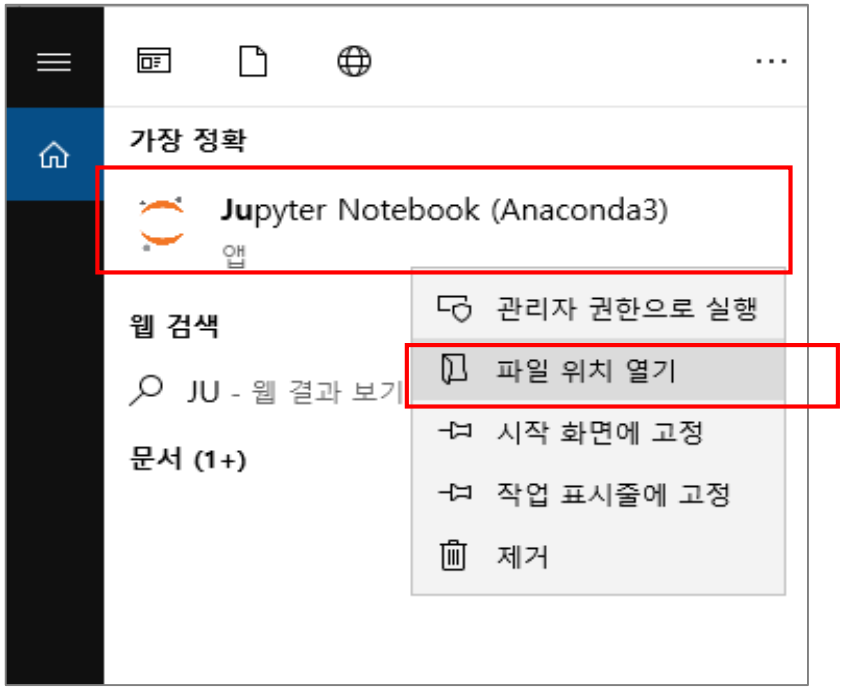
Anaconda가 Windows10(64)에서 Just me 기반으로 설치된 경우,

Jupyter notebook의 홈 디렉토리는

C:\Users\User_name (예: C:\Users\ai04)으로 설정된다.

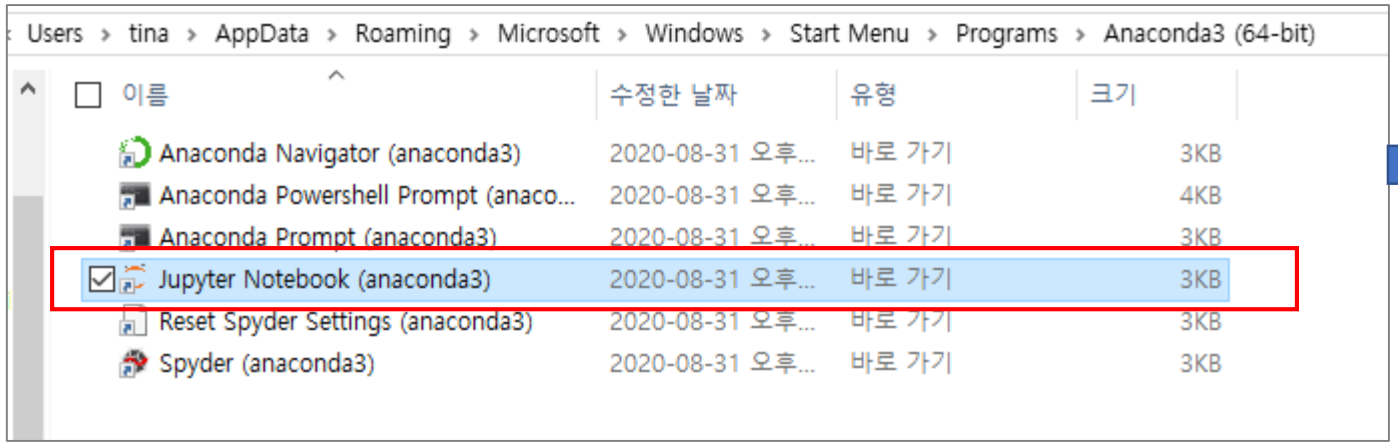
주피터 노트북 홈 디렉토리 변경

1 Jupyter notebook에서 마우스 오른쪽 버튼 클릭 → '파일 위치 열기'를 클릭한다.



주피터 노트북 홈 디렉토리 변경

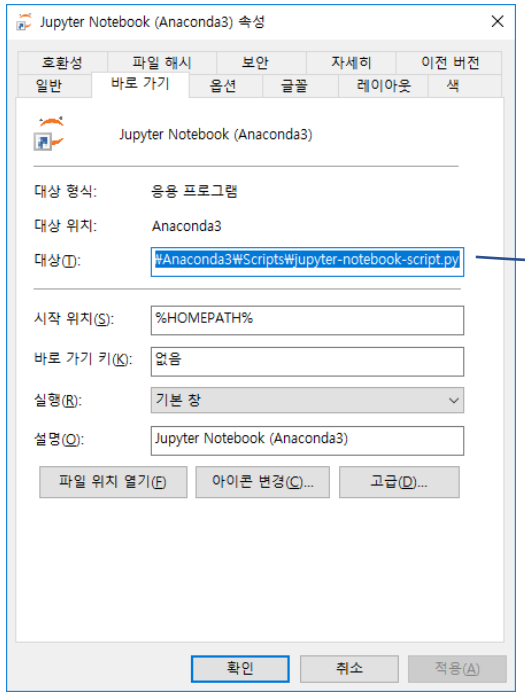
2 Jupyter Notebook ‘바로가기’ 아이콘에서
마우스 오른쪽 버튼을 클릭하여 맨 아래 ‘속성’을 클릭한다.



주피터 노트북 홈 디렉토리 변경

3 속성 창에서 대상을 선택하여

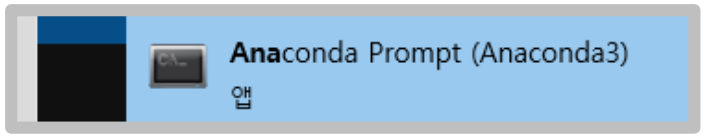
아래와 같은 경로를 확인한 후 "%USERPROFILE%/" 부분을 삭제한다.



C:\ProgramData\Anaconda3\python.exe
C:\ProgramData\Anaconda3\cwp.py
C:\ProgramData\Anaconda3
C:\ProgramData\Anaconda3\python.exe
C:\ProgramData\Anaconda3\Scripts\jupyter-notebook-script.py **"%USERPROFILE%/"**

주피터 노트북 홈 디렉토리 변경

4 Anaconda prompt를 실행한 후, 아래 명령을 입력한다.



(base) C:\Users\ai04>jupyter notebook --generate-config

(base) C:\Users\ai04>cd .jupyter

(base) C:\Users\ai04\.jupyter>dir

```
(base) C:\Users\ai04>jupyter notebook --generate-config
Overwrite C:\Users\ai04\.jupyter\jupyter_notebook_config.py with default config? [y/N]y
Writing default config to: C:\Users\ai04\.jupyter\jupyter_notebook_config.py

(base) C:\Users\ai04>cd .jupyter

(base) C:\Users\ai04\.jupyter>dir
 C 드라이브의 볼륨: Windows-SSD
 볼륨 일련 번호: 60D8-FCD4

 C:\Users\ai04\.jupyter 디렉터리

2021-05-31  오후 03:44    <DIR>          .
2021-05-31  오후 03:44    <DIR>          ..
2021-05-31  오후 03:54    <DIR>          custom
2021-03-03  오후 09:01            48,325 jupyter_notebook_config.py
2021-03-03  오후 09:02                26 migrated
2021-04-03  오후 07:43    <DIR>          nbconfig
               2개 파일              48,351 바이트
               4개 디렉터리  367,619,043,328 바이트 남음
(base) C:\Users\ai04\.jupyter>
```

C:\Users\User_name\.jupyter\jupyter_notebook_config.py 을 확인한다.

주피터 노트북 홈 디렉토리 변경

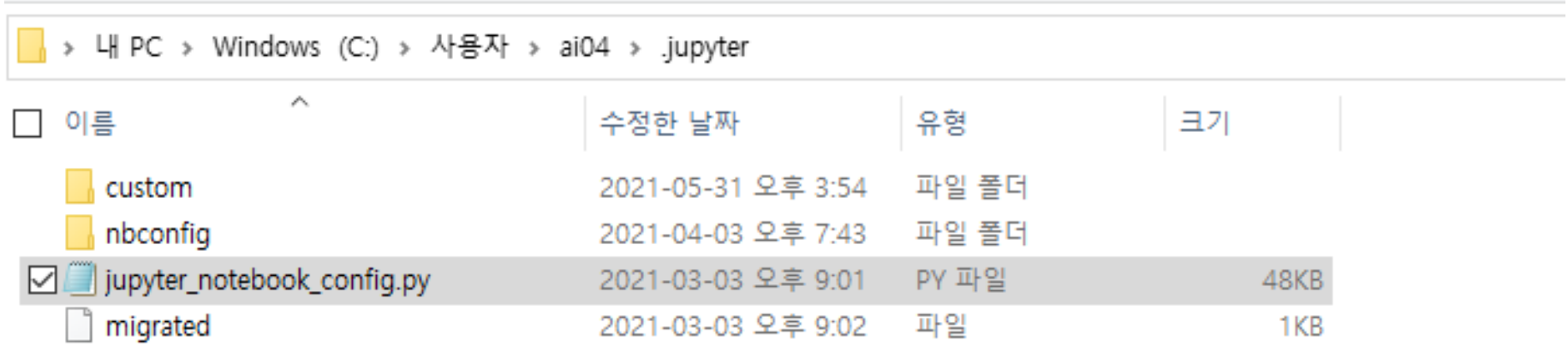
5

Users

User_name

C:\Users\ai04\.jupyter 디렉토리에서

jupyter_notebook_config.py 파일을 메모장 프로그램으로 읽는다.



주피터 노트북 홈 디렉토리 변경

6 메모장으로 읽은 jupyter_notebook_config.py 파일에서 문자 찾기 기능(Ctrl + f)을 이용하여 아래와 같은 코드를 찾는다.

```
#c.NotebookApp.notebook_dir = ''
```

찾아진 코드에서 맨 앞에 있는 #을 지우고, 맨 뒤에 있는 ' ' 사이에 교체할 홈 디렉토리명을 기록한다.

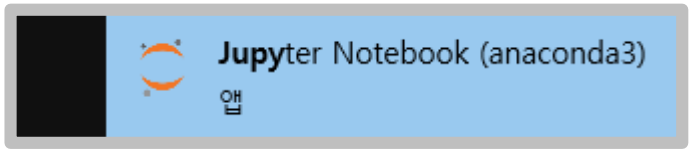
디렉토리 구분 기호는 백슬래쉬(\)가 아니라 슬래쉬(/) 를 사용한다.

예) c.NotebookApp.notebook_dir = 'D:/ai'

삭제

주피터 노트북 홈 디렉토리 변경

7 Jupyter Notebook을 재실행시켜서 지정한 디렉토리로 변경되어 있는지 확인한다.




```
[I **.**.**.*** NotebookApp] The port 8888 is already in use, trying another port.
[I **.**.**.*** NotebookApp] The port 8889 is already in use, trying another port.
[I **.**.**.*** NotebookApp] The port 8890 is already in use, trying another port.
[I **.**.**.*** NotebookApp] The port 8891 is already in use, trying another port.
[I **.**.**.*** NotebookApp] JupyterLab extension loaded from C:\Users\ai04\Anaconda3\lib\site-packages\jupyterlab
[I **.**.**.*** NotebookApp] JupyterLab application directory is C:\Users\ai04\Anaconda3\share\jupyter\lab
[I **.**.**.*** NotebookApp] Serving notebooks from local directory: D:/ai
[I **.**.**.*** NotebookApp] The Jupyter Notebook is running at:
[I **.**.**.*** NotebookApp] http://localhost:8892/?token=24907f4e1deee1a0a87870a25ecf65892934f50af1a0a6a9
[I **.**.**.*** NotebookApp]  or http://127.0.0.1:8892/?token=24907f4e1deee1a0a87870a25ecf65892934f50af1a0a6a9
[I **.**.**.*** NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C **.**.**.*** NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/ai04/AppData/Roaming/jupyter/runtime/nbserver-20596-open.html
    Or copy and paste one of these URLs:
        http://localhost:8892/?token=24907f4e1deee1a0a87870a25ecf65892934f50af1a0a6a9
     or http://127.0.0.1:8892/?token=24907f4e1deee1a0a87870a25ecf65892934f50af1a0a6a9
```

주피터 노트북 or 아나콘다 프롬프트에서 Python 버전 확인방법

참고

Jupyter Notebook에서 설치된 python 버전 확인하는 방법

```
1 import sys
2 print("--sys.version-")
3 print(sys.version)
```

```
--sys.version-
3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
```

아나콘다 프롬프트에서 설치된 python 버전 확인하는 방법

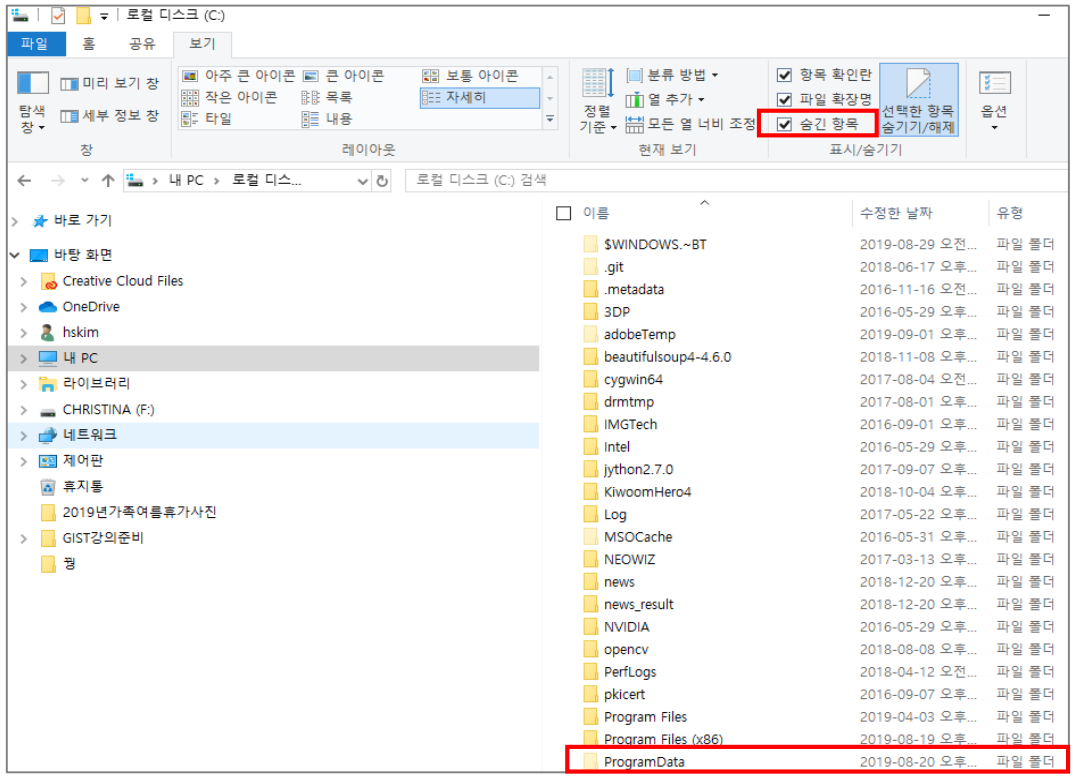
```
(base) C:\Users\ai04>python --version
Python 3.8.8
```

아나콘다 Uninstall (예시 : 2019. 07 버전)

1

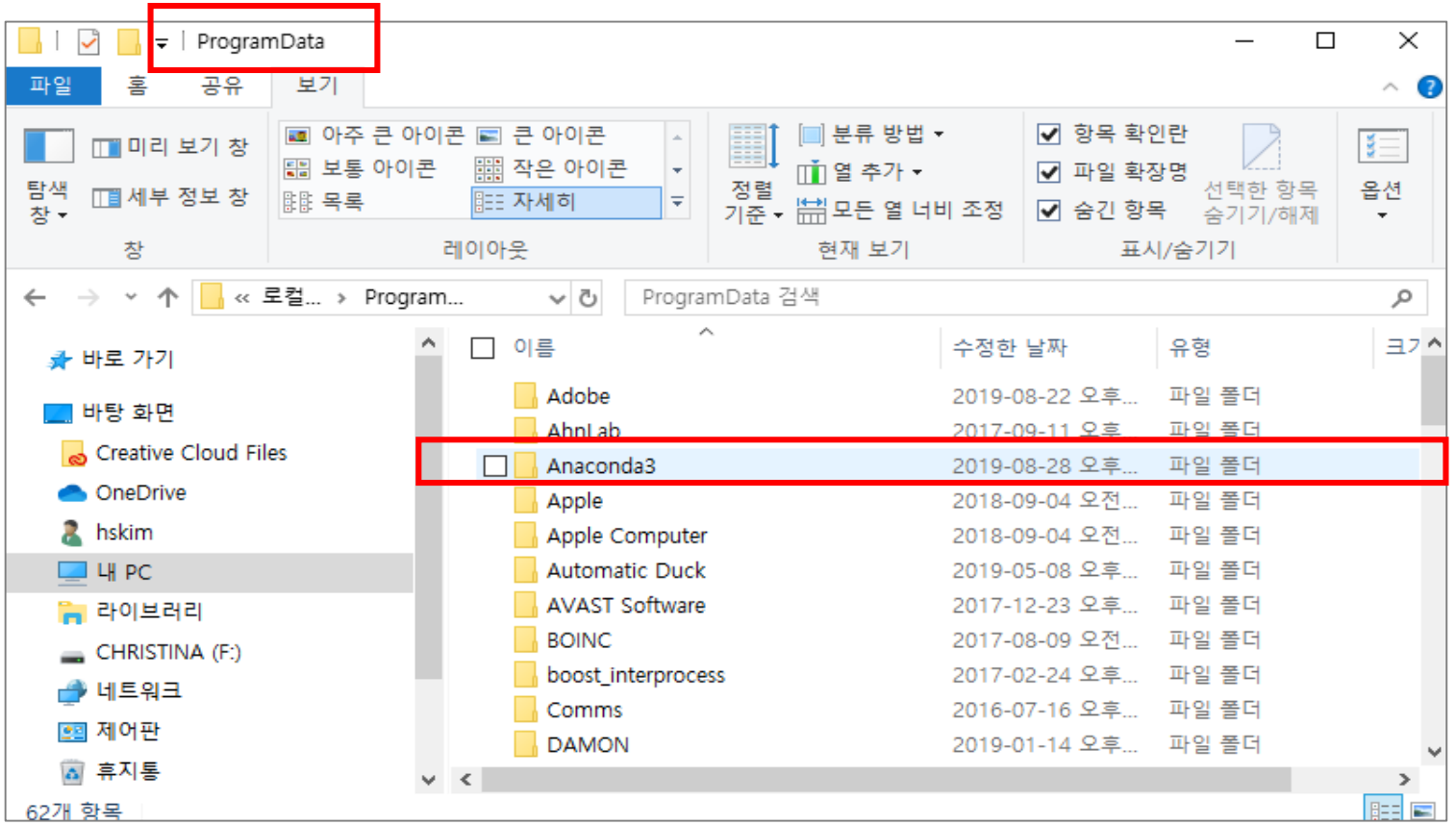
- **Just me type**
  Program Files 경로에 설치됨

- **All user type**
  Program Data 경로에 설치됨 (Program Data 경로는 숨김 폴더)



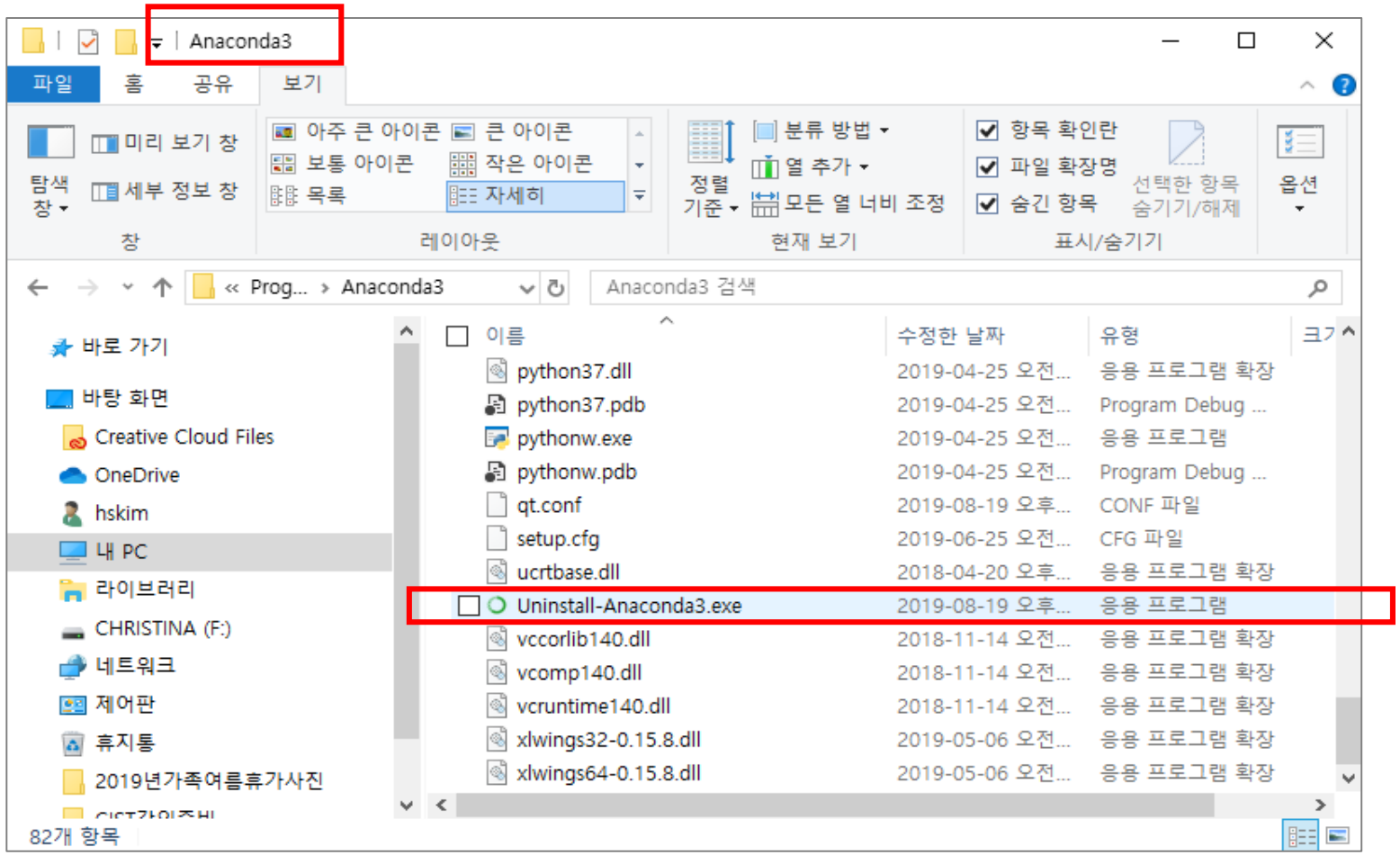
아나콘다 Uninstall (예시 : 2019. 07 버전)

2



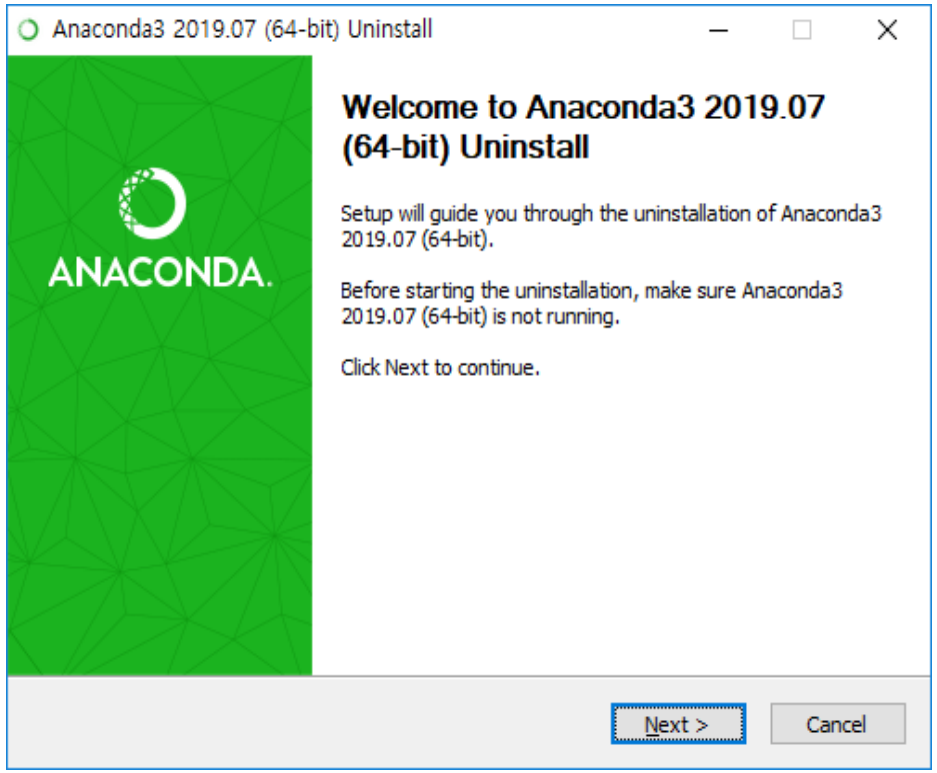
아나콘다 Uninstall (예시 : 2019. 07 버전)

3

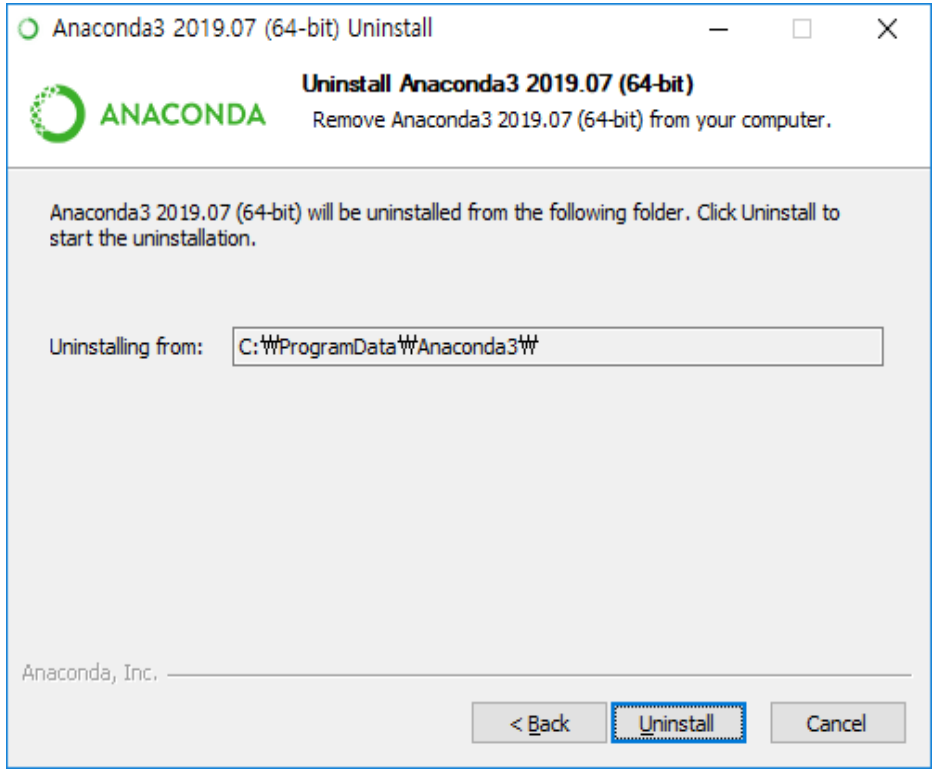


아나콘다 Uninstall (예시 : 2019. 07 버전)

4

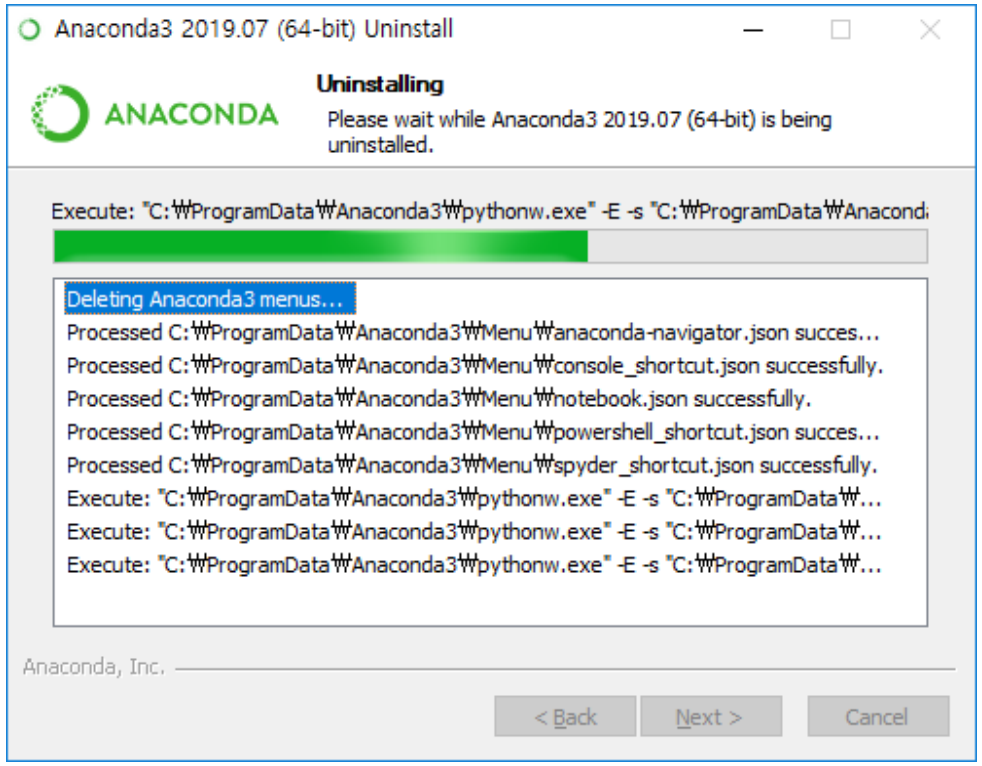


5

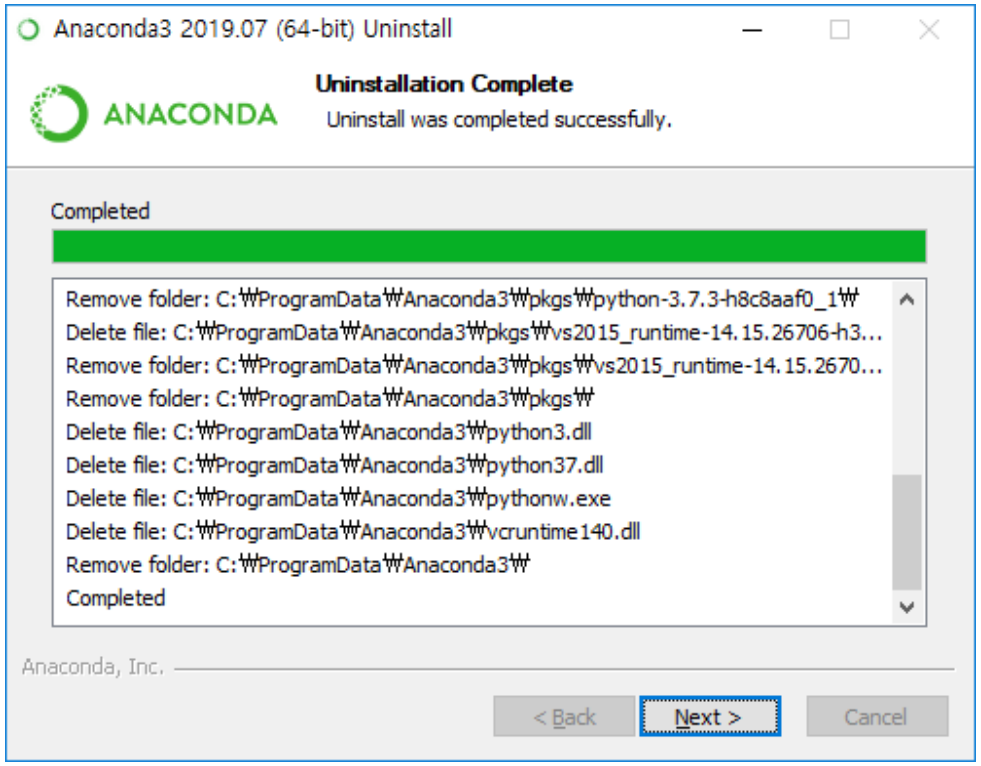


아나콘다 Uninstall (예시 : 2019. 07 버전)

6

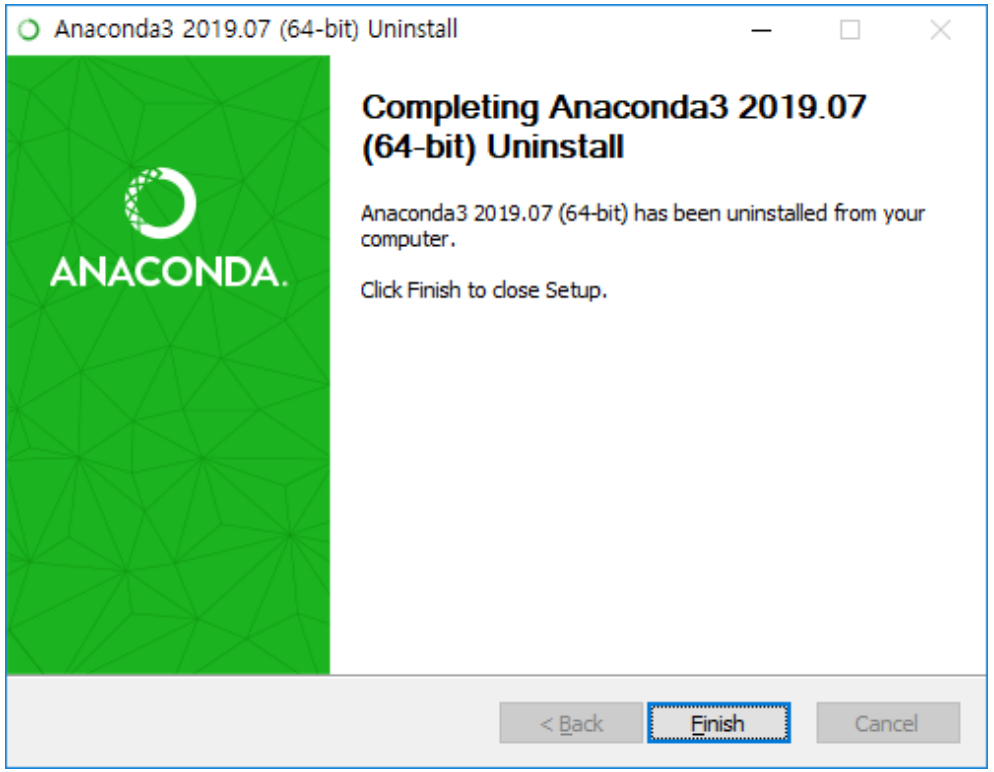


7



아나콘다 Uninstall (예시 : 2019. 07 버전)

8

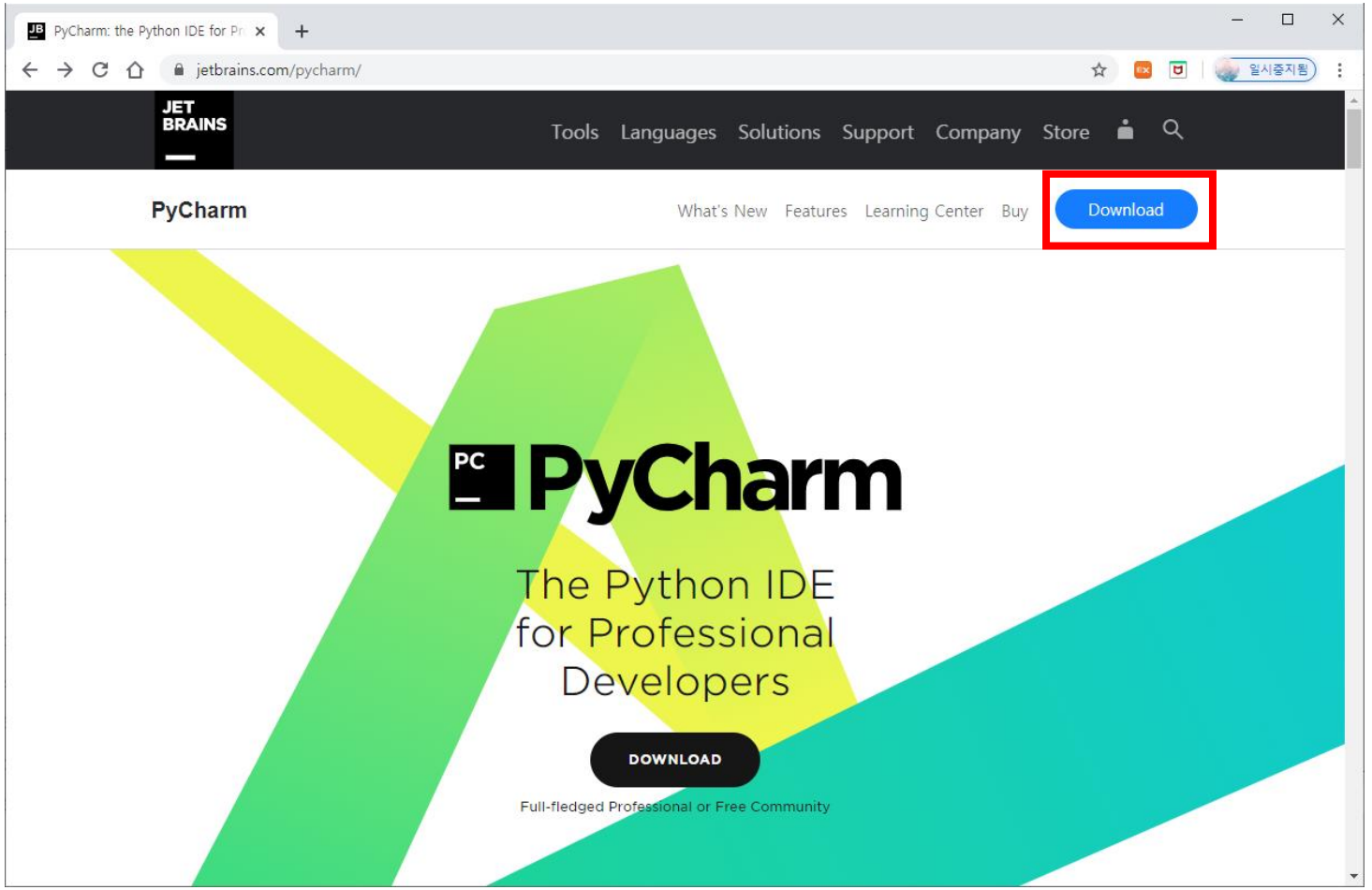


Anaconda를 재설치 하기위해서는 재부팅 후 설치를 권장함

Pycharm 설치 (예시 : 2020. 02. 01 버전)

- 파이참 공식 홈페이지 https://www.jetbrains.com/Pycharm/ 에서 설치 실행 파일 다운로드

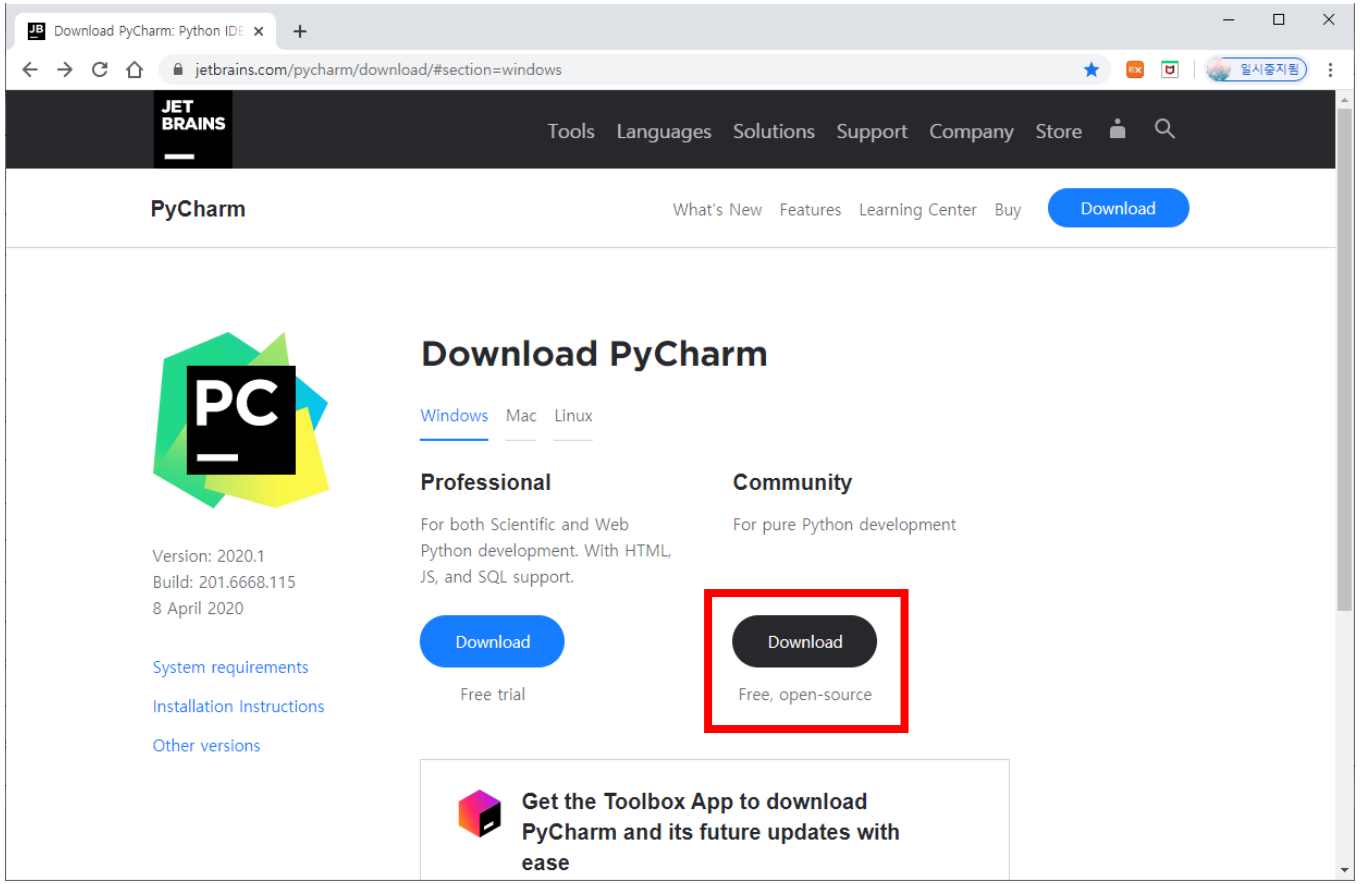
1



Pycharm 설치 (예시 : 2020. 02. 01 버전)

- 파이참 공식 홈페이지 https://www.jetbrains.com/Pycharm/ 에서 무료 community 버전 다운로드

2

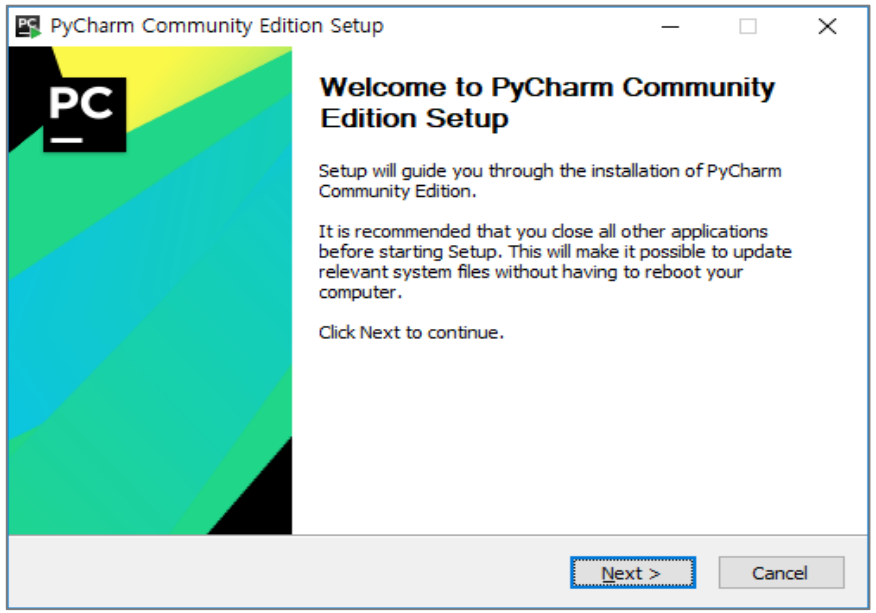


Pycharm 설치 (예시 : 2020. 02. 01 버전)

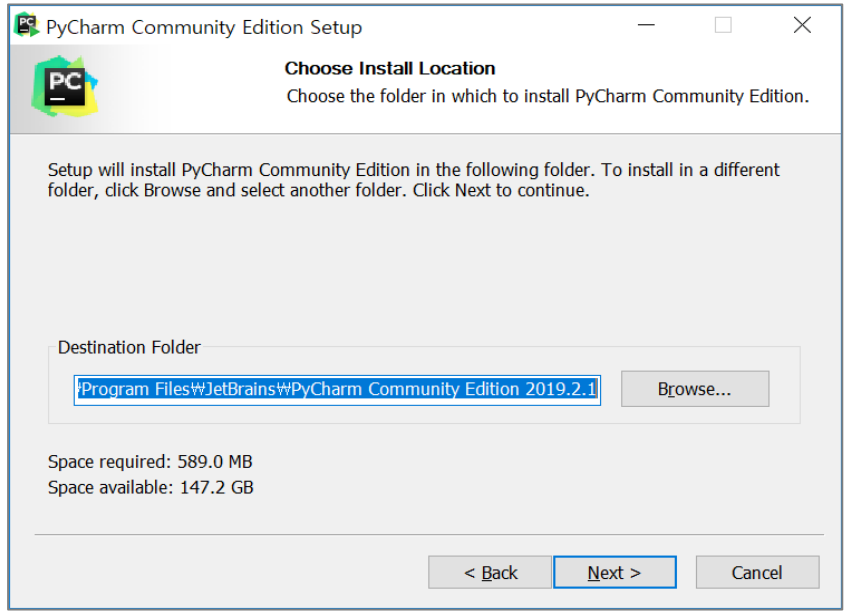
■ 다운로드된 파이참 community 버전 실행 파일을 더블클릭하여 설치하기

1

2

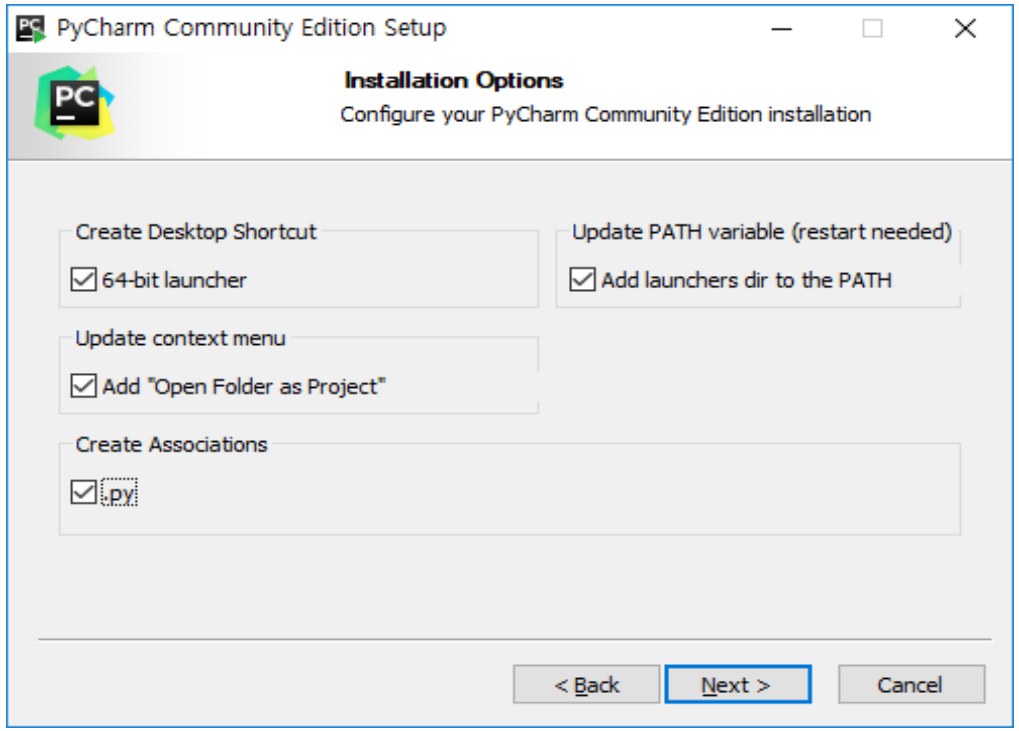


3

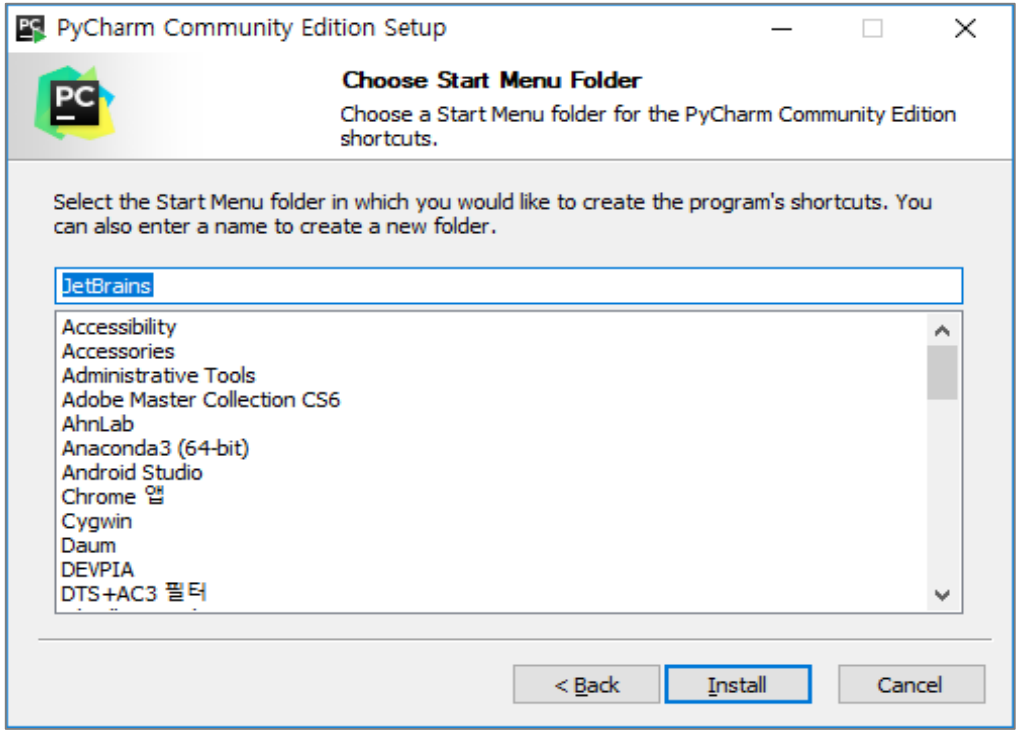


Pycharm 설치 (예시 : 2020. 02. 01 버전)

4

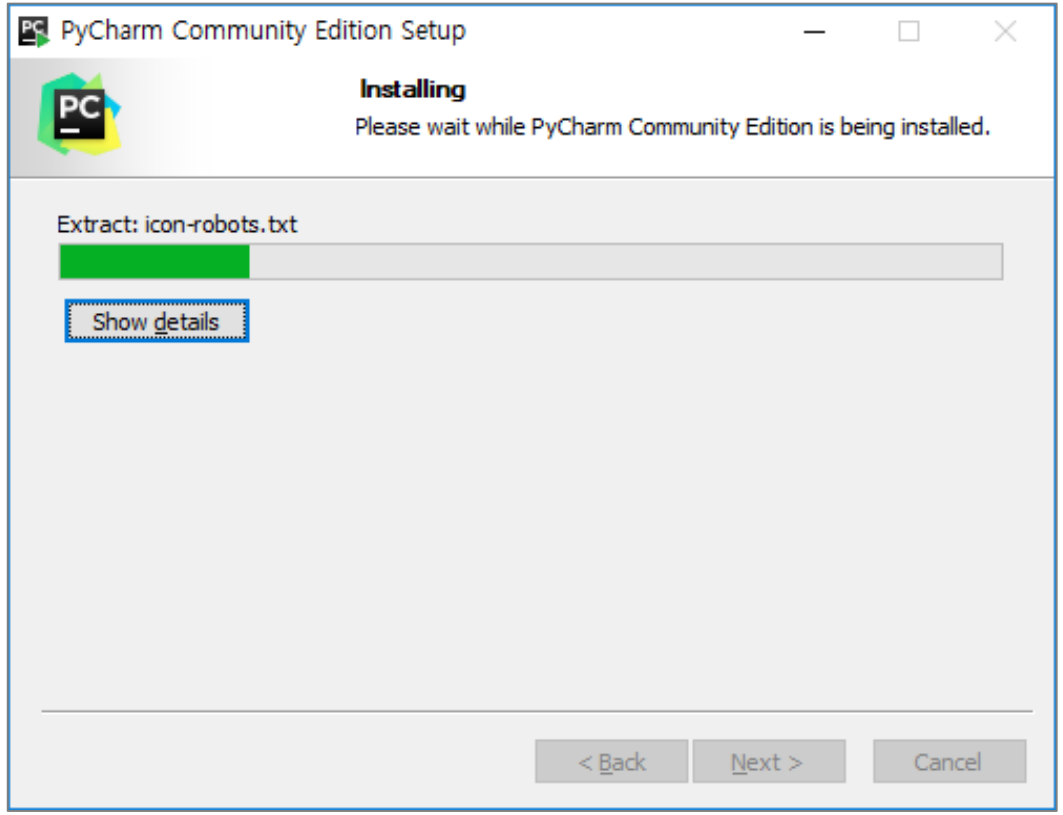


5

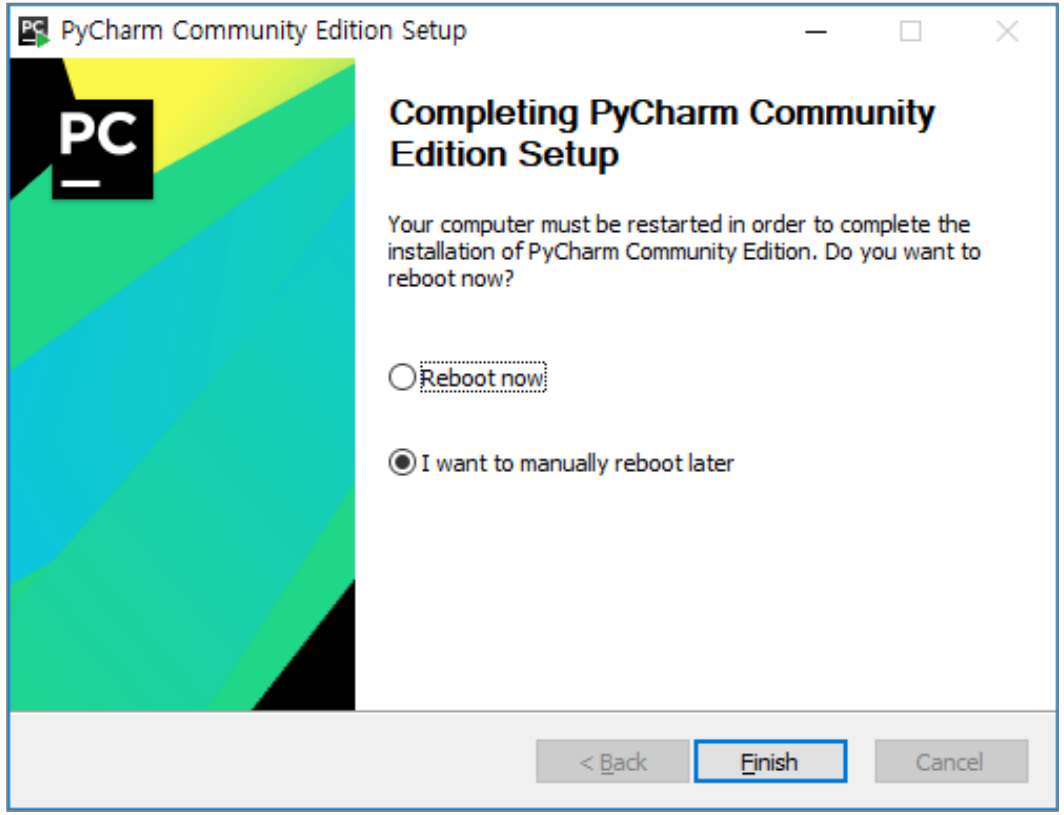


Pycharm 설치 (예시 : 2020. 02. 01 버전)

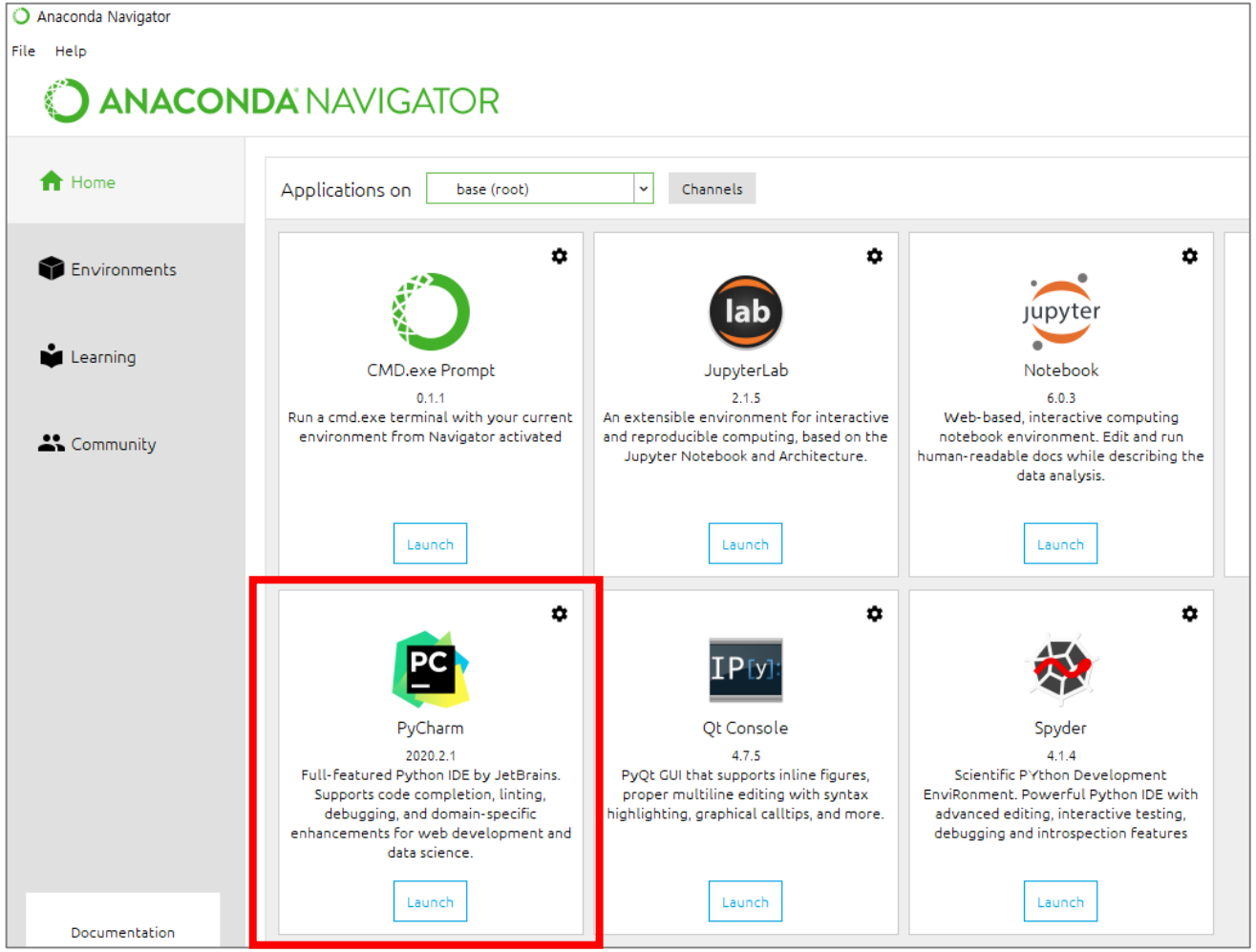
6



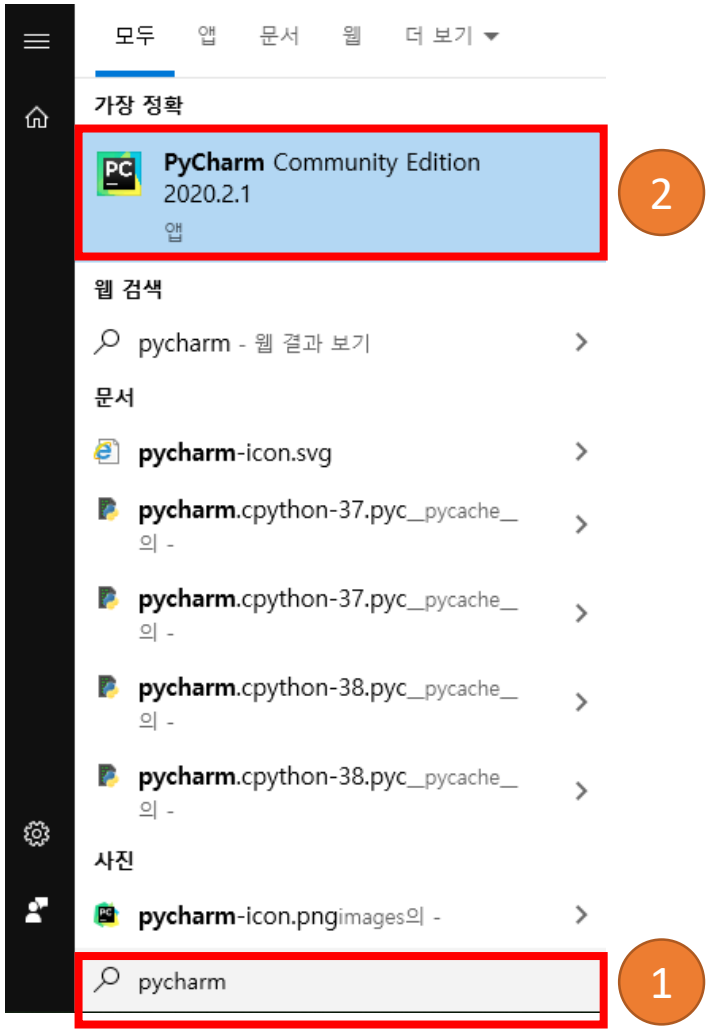
7



Pycharm 실행 (예시 : 2020. 02. 01 버전)

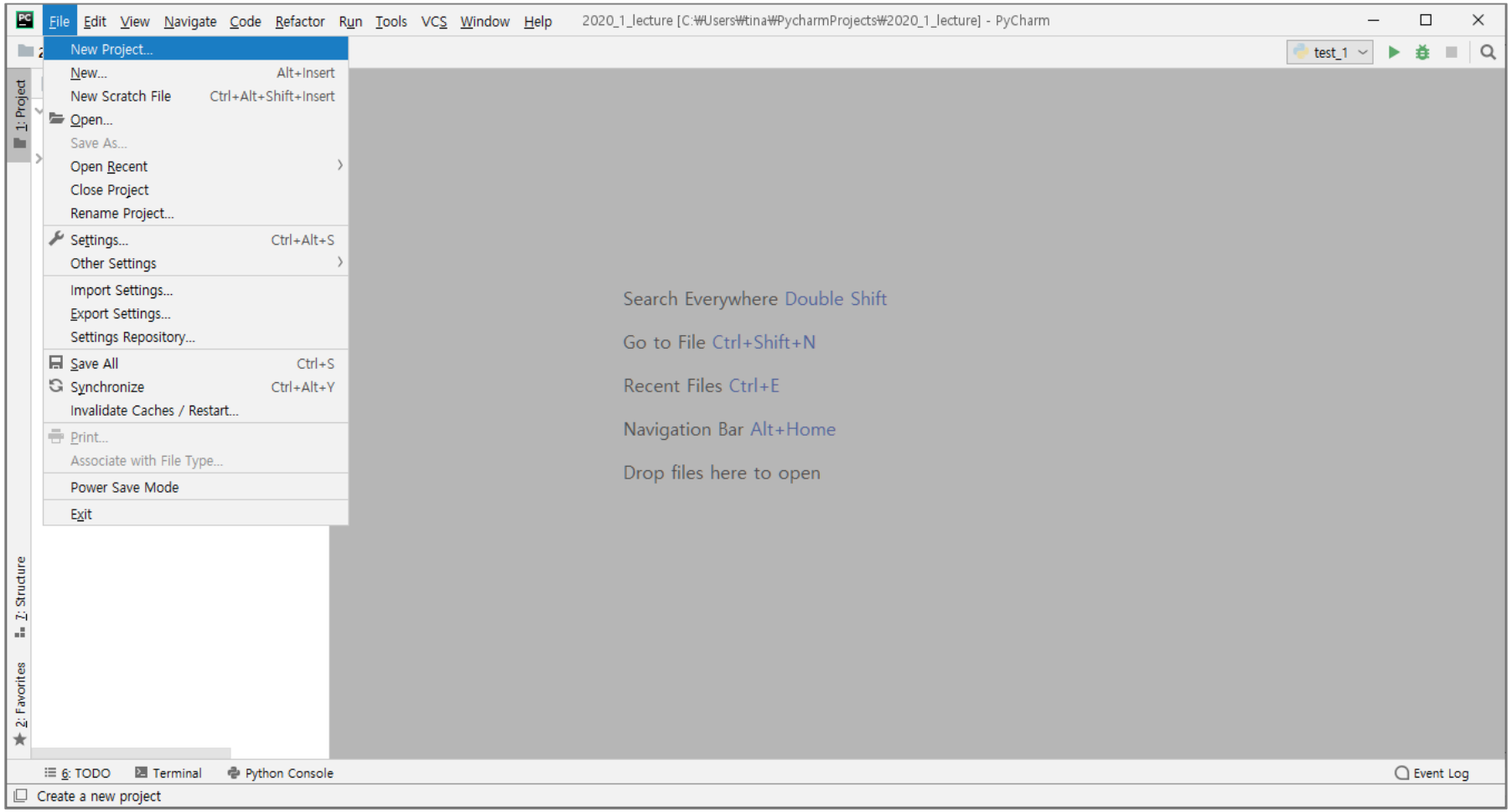


Pycharm 실행 (예시 : 2020. 02. 01 버전)



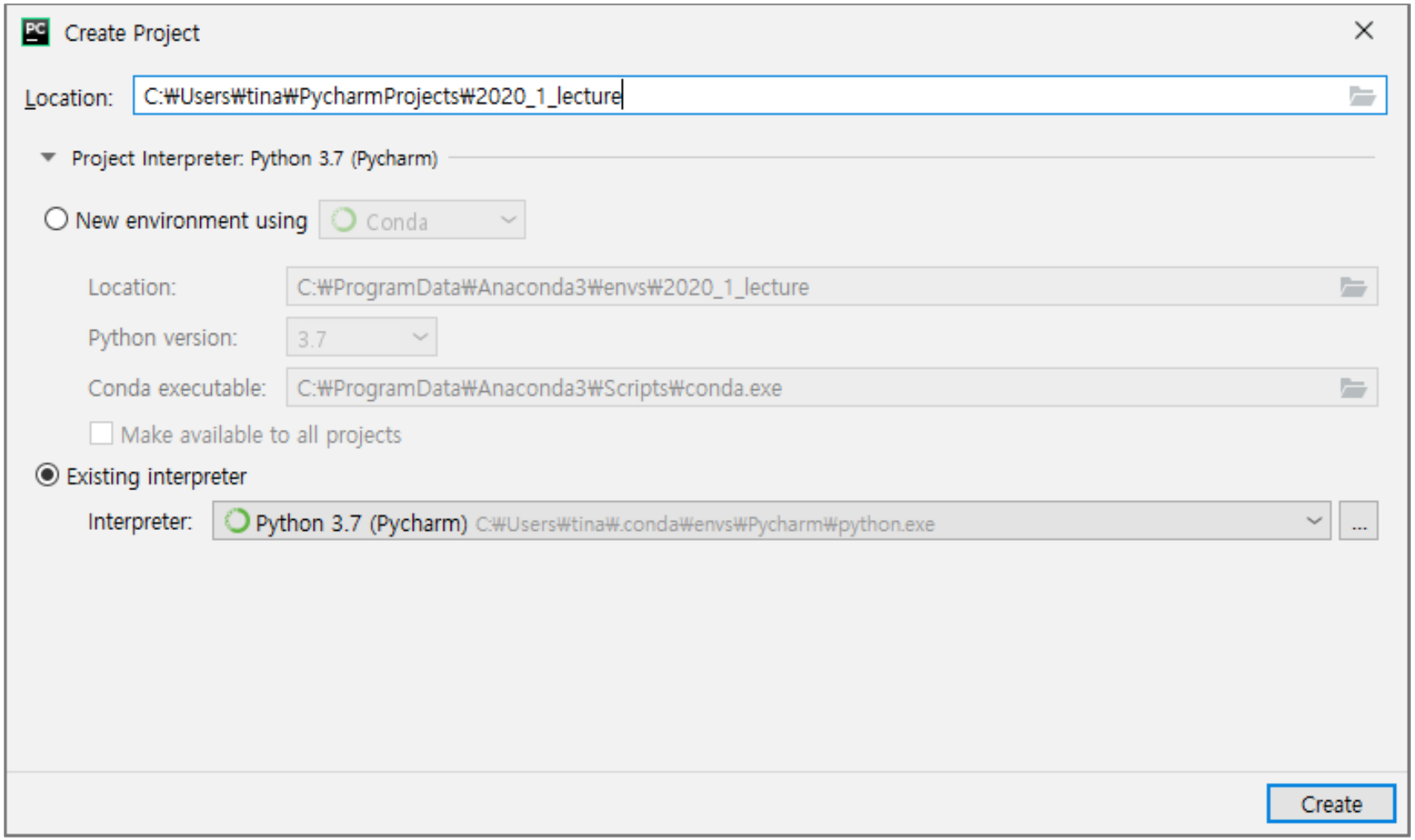
Pycharm 사용법 (예시 : 2020. 02. 01 버전)

1



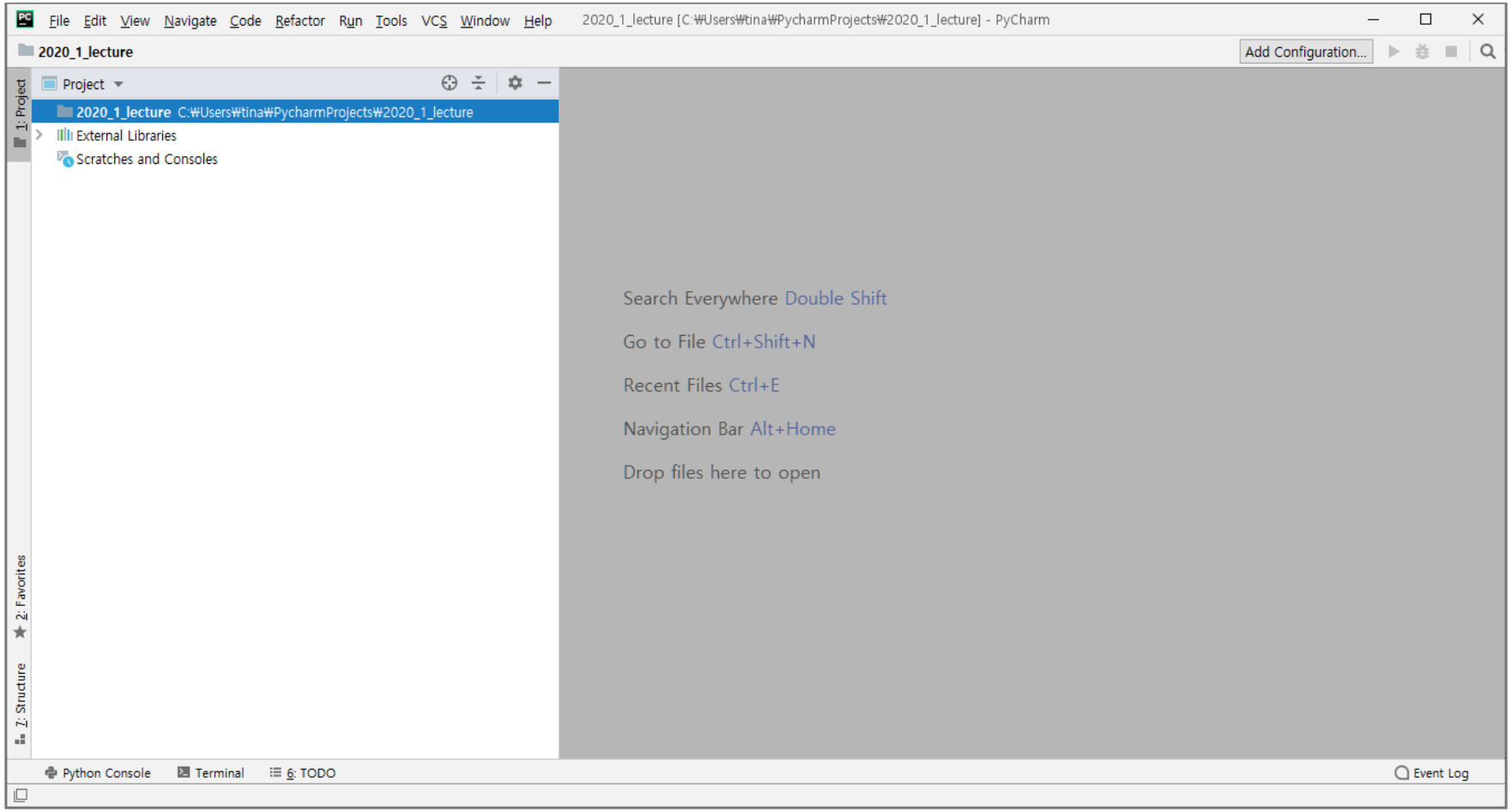
Pycharm 사용법 (예시 : 2020. 02. 01 버전)

2



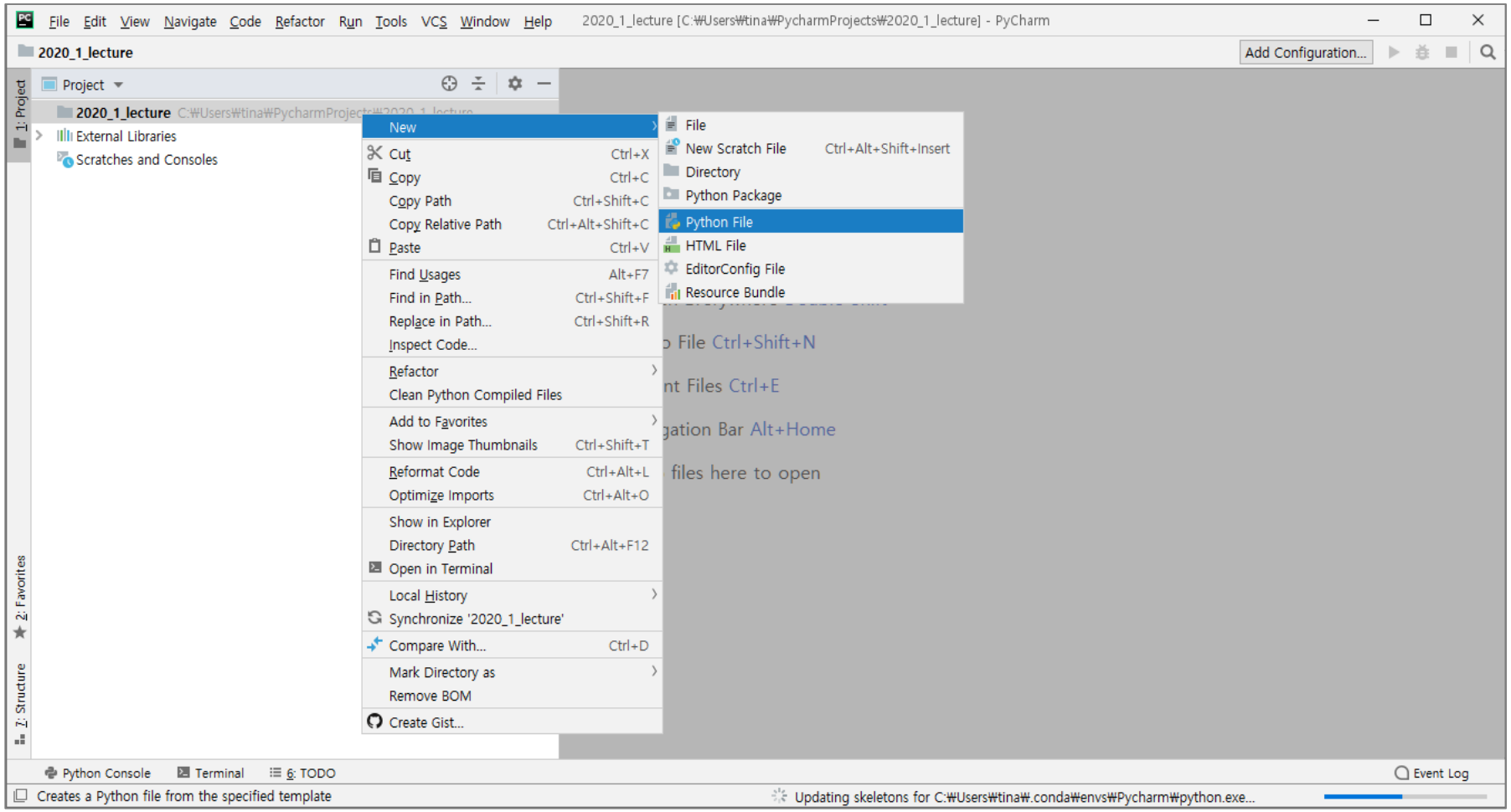
Pycharm 사용법 (예시 : 2020. 02. 01 버전)

3



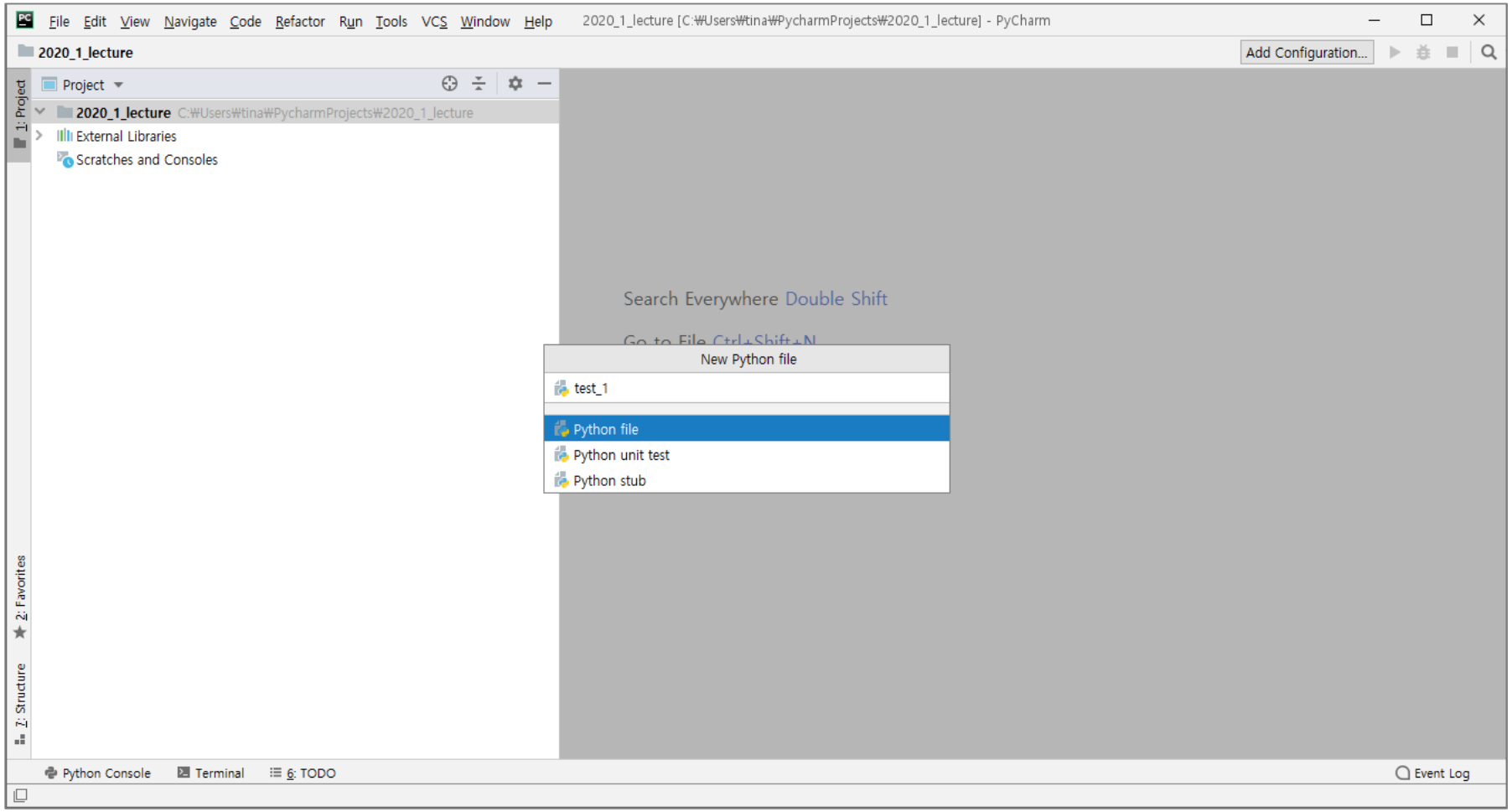
Pycharm 사용법 (예시 : 2020. 02. 01 버전)

4



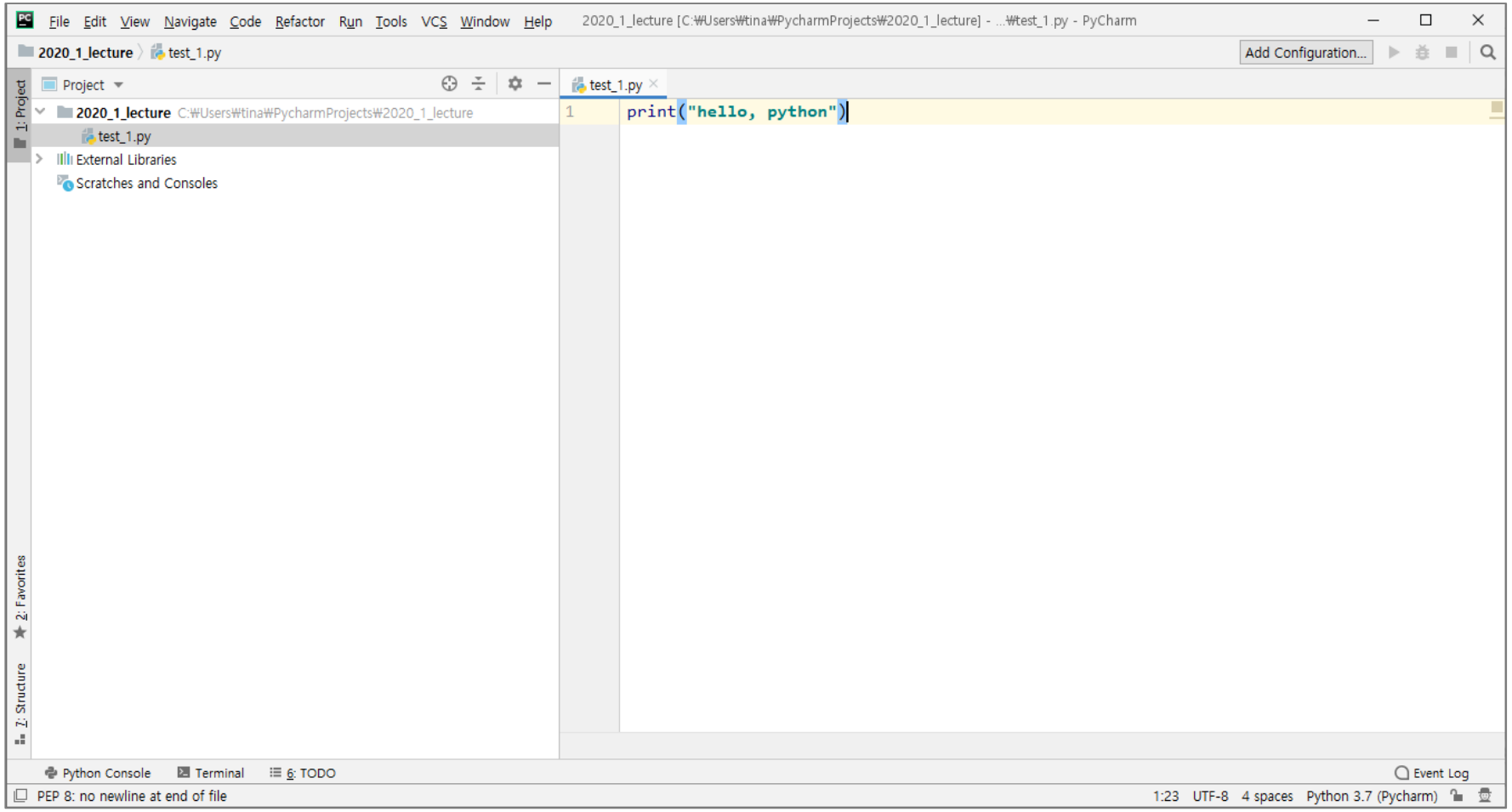
Pycharm 사용법 (예시 : 2020. 02. 01 버전)

5



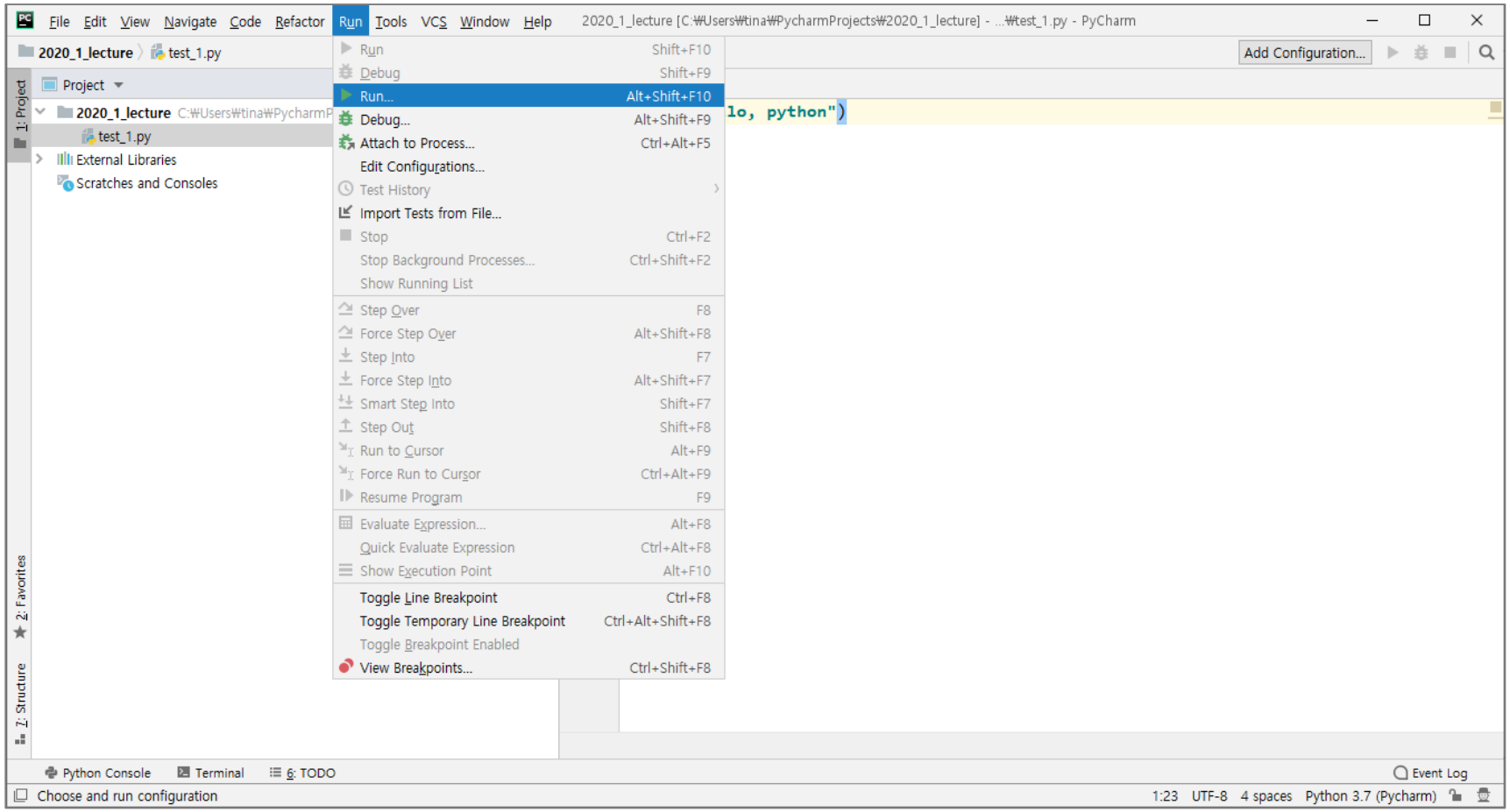
Pycharm 사용법 (예시 : 2020. 02. 01 버전)

6



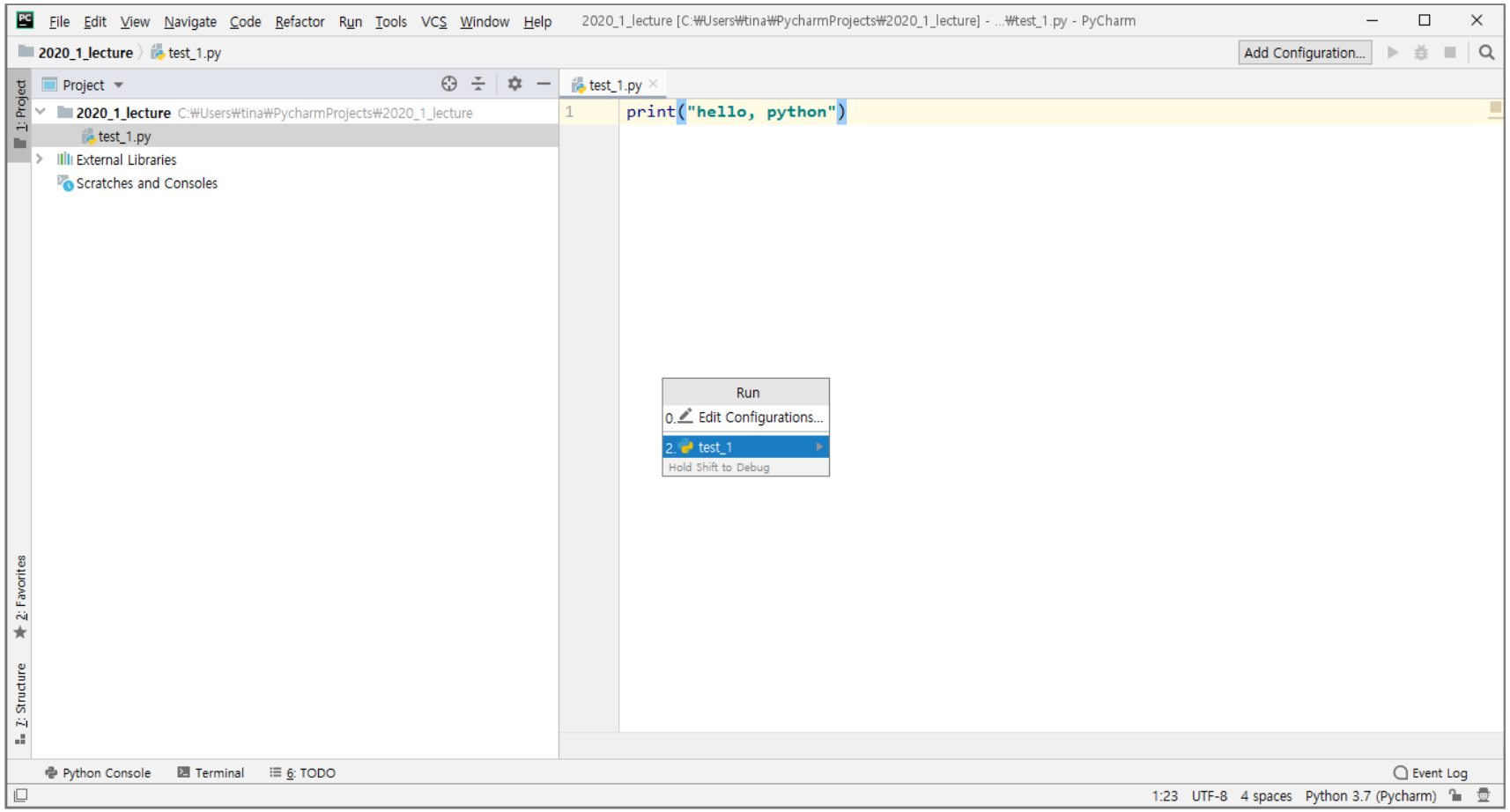
Pycharm 사용법 (예시 : 2020. 02. 01 버전)

7



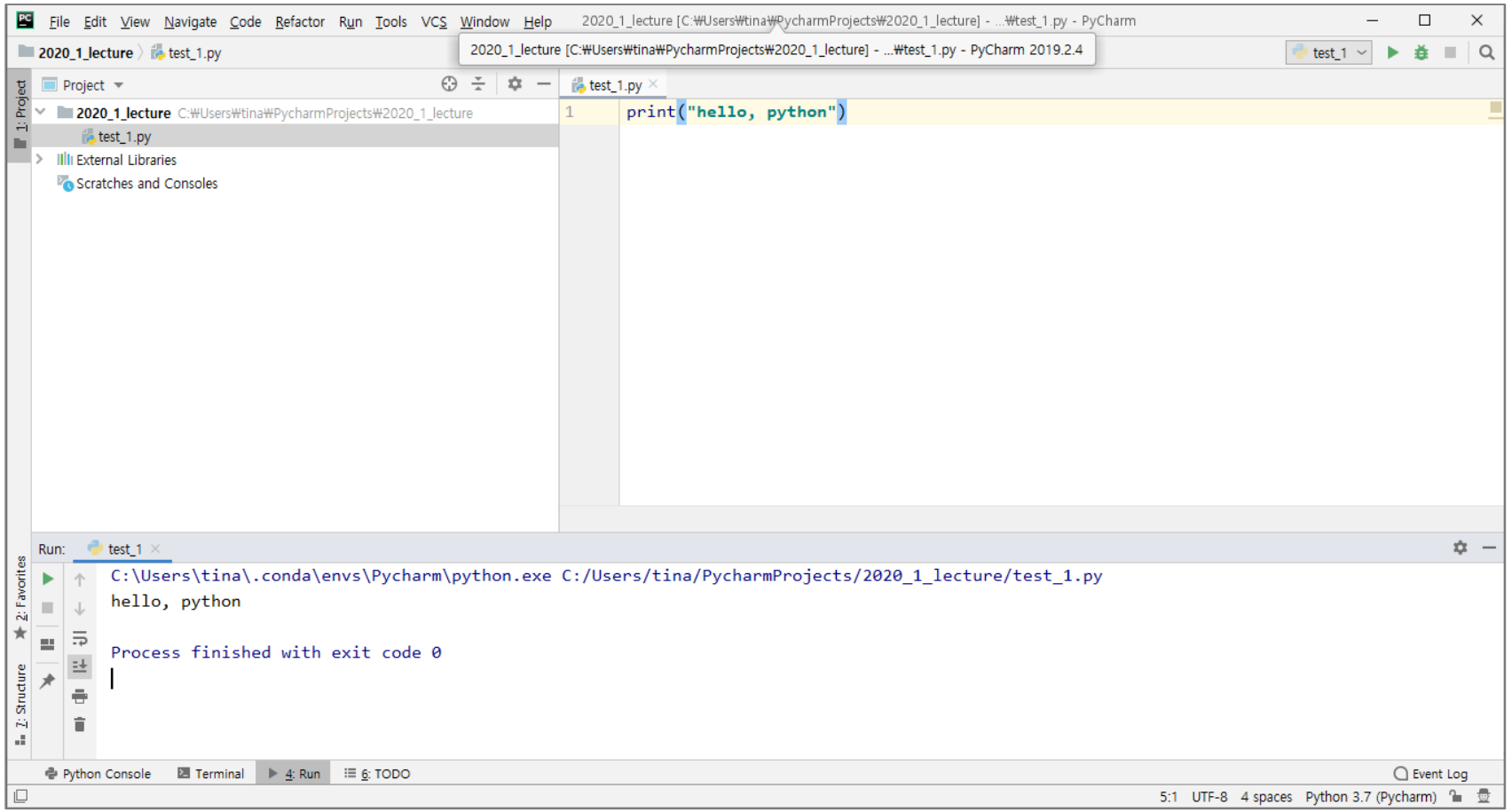
Pycharm 사용법 (예시 : 2020. 02. 01 버전)

8



Pycharm 사용법 (예시 : 2020. 02. 01 버전)

9



Python 2 / Python 3 비교

- Python은 1990년 처음 탄생한 이후에 많은 발전하였고, version 1에서 version 2로 발전 하면서 많은 새로운 개념과 기능이 추가되면서도 하위 버전과 호환성을 유지하였다.
- version 2의 경우 version 1의 잘못된 문제나 버리고 싶은 문제들도 호환성이라는 이름으로 유지하였지만 version 3은 기존 version에서 잘못되거나 비효율적인 것들을 정리하고 새롭게 시작하였다.
  즉 version 3부터는 하위 version과 호환성을 유지하지 않는다.
- Python version 2에서 작성된 프로그램을 version 3에서 완벽히 실행할 수 없다.
- Version 2도 2.7을 기준으로 더 이상 새로운 버전이 발표되지 않으며 2020년 1월 1일부로 파이썬 2의 지원이 종료되었다.
- 보안 및 버그로 인하여 2.7.x 버전으로 업데이트는 진행되고 있으나 기능상 업데이트는 없다.
- Python version 3으로 계속 업데이트 되고 있다고 해서 version 3으로만 권장할 수가 없다.
  가장 대표적인 이유가 현재 프로그램의 생태 환경이다.
  - ✓ Python은 수 많은 외부 모듈과 함께 동작하는데 아직 많은 모듈이 Python version 3을 완벽하게 지원하지 않았기 때문에 기존의 많은 모듈이 Python 2 버전을 유지하고 있다.
  - ✓ 점차적으로 많은 모듈들이 version 3로 이식되고 있으므로 향후 Python을 이용하여 프로그램을 개발할 경우 version 3으로 개발하는 것을 권장한다.