

ML&DL

WITH PYTHON

PROGRAMING

A thick, wavy yellow line runs vertically along the left side of the slide, starting from the top and extending to the bottom. It has a slightly irregular, hand-drawn appearance.

PART 1

ABOUT ML&DL

CH00. ML & DL 살펴보기

ML & DL 살펴보기

◆ 4차 산업혁명



- 인공지능, 사물인터넷, 빅데이터, 모바일 등 첨단 정보통신기술이 경제, 사회 전반에 융합되어 혁신적인 변화가 나타나는 차세대 산업혁명
- 핵심 기술 ➔ 인공지능, 사물인터넷, 클라우드, 빅데이터, 드론, 자율주행, 가상현실

ML & DL 살펴보기

◆ AI란

- Artificial Intelligence 약자
- 인간의 학습능력과 추론능력, 지각능력, 자연언어의 이해능력 등을 컴퓨터 프로그램으로 실현한 기술
- 컴퓨터를 사람처럼 생각하게 만드는 기술
- 인공지능을 공부하고 구현하기 위해서
 - 값비싼 고성능 컴퓨터와 무수히 많은 데이터가 필요
 - 많은 비용과 시간이 필요
 - 인공지능 분야의 전문가가 되기 위한 저렴한 플랫폼이 부족한 현실

ML & DL 살펴보기

◆ AI 발전 역사



* 출처 : I-Korea 4.0 실현을 위한 인공지능 R&D 전략 (과기정통부, 2018.5)

* 본 그림은 2018.5 현재까지의 인공지능분야 주요 결과물과 향후 2030년까지의 주요 기술동향을 함께 표시함

무단 전재 및 복제를 금합니다. Copyright© by SO.A

ML & DL 살펴보기

◆ AI 발전 역사

- 1955년 존 매카시 논문 <지능이 있는 기계를 만들기 위한 과학과 공학> 처음 등장
- 1956년 다스머스 학회에서 공식적으로 인공지능이라는 용어 사용
- 뇌를 모사한 인공지능을 뜻하는 퍼셉트론(Perceptron) 용어도 탄생
- 1960년에 접어들면서 인공지능 연구가 본격화
- 1970년대 초까지 인간처럼 생각하고 문제를 푸는 인공지능 연구 계속
- 1970~1980년대 한계
- 1980~1990년대 신경망, 다층 인식론 등 연구
- 2000년대 머신러닝, 딥러닝 인공지능 연구 확산

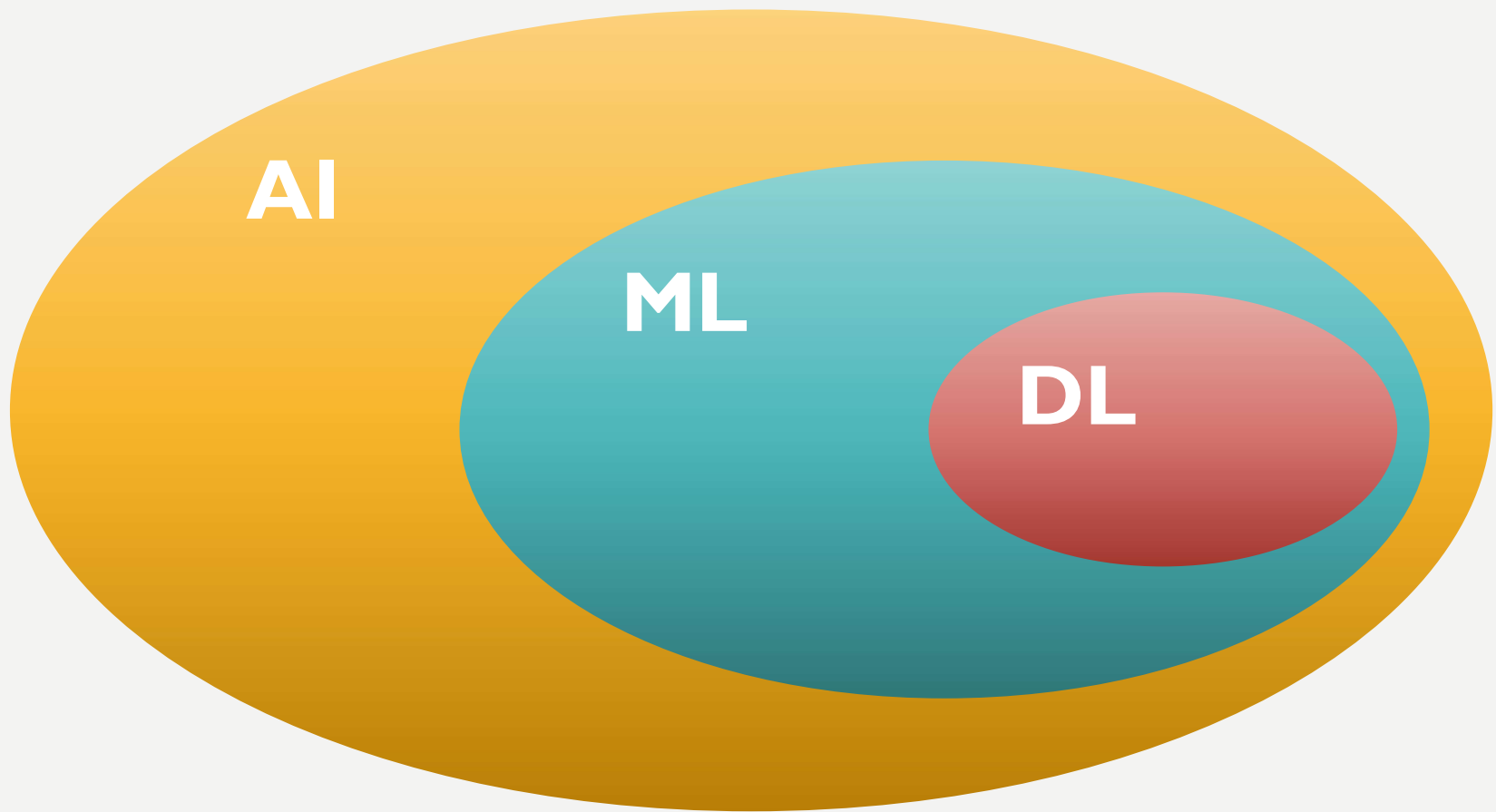
ML & DL 살펴보기

◆ AI 구현 기술

구분	1세대	2세대	3세대	4세대
시기	1950~1980년대	1990년대	2000년 대	2010년 대 ~ 현재
특징	제어 프로그램	경로 및 DB 탐색	머신러닝	딥러닝
내용	- 기계 및 가전제품에 탐재된 단순 제어 프로그램	-대량 정보와 규칙기반 경로 탐색 -DB검색 후 정답 파악 -전문가 시스템	-컴퓨터 스스로 규칙 및 지식 학습 -예측 방법 파악 -인공시경막	-추상화된 특징 표현 등 고급 지식 학습 -데이터 변형 및 인사이트 파악 -깊은 인공신경막
활용	자동 세탁기	검색 DB	문자 및 패턴인식	영상 및 음성인식 자연어 처리

ML & DL 살펴보기

◆머신러닝 & 딥러닝 관계



ML & DL 살펴보기

◆ 머신러닝(Machine Learning)

- 인공지능의 한 분야
 - 패턴인식과 컴퓨터 학습 이론의 연구로부터 진화한 분야
-
- 1949년 Hubbian Learning Theroy를 발표하면서 시작
 - 그 이후 1952년 Arthur Samuel이 경험으로부터 배우는 방법을 사용한 최초의 머신러닝 프로그램인 체커 프로그램 개발
-
- 경험적 데이터 기반 학습하고 예측 수행으로 스스로의 성능을 향상시키는 시스템과 이를 위한 알고리즘을 연구하고 구축하는 기술
 - 입력 데이터 기반으로 예측이나 결정을 이끌어내기 위해 특정한 모델 구축 방식

ML & DL 살펴보기

◆ 머신러닝(Machine Learning)

알고리즘	특징	
지도(supervised)학습	-입력 -출력 -기법	문제 + 결과 데이터 예측 결과 제공 회귀, 분류, 랭킹
강화(reinforcement)학습	-입력 -출력 -기법	문제 데이터만 제공 결과, 결과 평가 후 보상 알고리즘 트레이닝 -상 최대화 / 벌 최소화 방향
비지도(unsupervised)학습	-입력 -출력 -기법	문제 데이터만 제공 분석 결과 제공 군집화, 토픽 모델링, 밀도 추정, 차원 축소

ML & DL 살펴보기

◆ 머신러닝(Machine Learning)

기법 및 모델	특징
분류(Classification)	데이터의 특정 그룹 부여하여 구분
회귀(Regression)	수치 데이터에 사용되는 기법, 연속형 결과 예측
의사결정 나무(Decision Tree)	트리 구조 형태의 예측 모델을 사용하는 기법
인공 신경망 (Neural Network)	생물의 신경 네트워크 구조와 기능 모방 기법
유전자 프로그래밍(Genetic Programming)	생물의 진화 알고리즘에 기반한 기법
군집화(Clustering)	관측된 예를 군집하여 부분집합으로 배분 기법
몬테카를로 방법 (Monter Carlo method)	무작위 추출된 난수 통해 확률로 계산하는 기법

ML & DL 살펴보기

◆ 딥 러닝(Deep Learning)

- 머신러닝 방식 중 학습 데이터를 구분하는 층을 많이 만들어 그 정확도를 올리는 방법을 딥러닝(Deep Learning)
- 처음에는 단순하게 선이나 색만 구별
- 나중에는 모양 인식
- 다음엔 추상적인 레벨까지 구분
- 2016년 알파고에 적용
- 영상 인식, 음성 인식, 자연 언어 처리 등 분야에서 우수한 성능 발휘

ML & DL 살펴보기

◆ 딥 러닝(Deep Learning)

알고리즘	특징
퍼셉트론(Perceptron)	<ul style="list-style-type: none">-1957년 고안된 알고리즘-딥러닝 신경망의 기원이되는 알고리즘-다수의 신호를 입력 받아서 하나의 신호 출력(0 / 1)-뉴런 또는 노드로부터 신호 입력-뉴런 또는 노드로 신호 출력
인공신경망 (ANN: Artificial Neural Network)	<ul style="list-style-type: none">-1980년대부터 활발히 연구-뇌신경망의 패턴 인식 방식의 통계학적 학습 알고리즘-선형 맞춤(linear fitting)과 비선형 변환(nonlinear-transformation or activation) 반복 수행하여 최적화
심층신경망 (DNN: Deep Neural Network)	<ul style="list-style-type: none">-신경망이 다수의 층의 깊이로써 구성된 개념-입력층과 출력층 사이 다수의 은닉층으로 구성된 ANN-분류 및 수치 예측을 위한 것-이미지 트레이닝, 문자인식 매우 유용

ML & DL 살펴보기

◆ ML & DL용 필수 라이브러리

라이브러리	언어	특징
NumPy	python	<ul style="list-style-type: none">- www.numpy.org- 파이썬에서 수치해석, 통계, 과학 계산을 위한 모듈- Numeric라이브러리를 이어 SciPy Core로 다시 2005년에 NumPy로 개명- array자료 생성, 색인, 처리, 연산 수행- Scipy, Pandas, matplotlib 등 다른 Python 패키지와 함께 사용
SciPy	python	<ul style="list-style-type: none">- www.scipy.org/- 과학 계산을 위한 함수를 모아 놓은 오픈 소스 파이썬 패키지- 선형 대수, 함수 최적화, 신호 처리, 특수 수학 함수, 통계 분포 등
Matplotlib	python	<ul style="list-style-type: none">- www.matplotlib.org- Matlab을 기반으로 파이썬으로 확장한 패키지로 그래프 패키지- 데이터와 분석 결과를 다양한 관점으로 시각화 가능
Pandas	python	<ul style="list-style-type: none">- www.pandas.pydata.org/- Series와 DataFrame 데이터 구조 제공 및 데이터 처리 패키지
Mglearn	python	<ul style="list-style-type: none">- 데이터 시각화 유틸리티 함수 제공

ML & DL 살펴보기

◆ ML 라이브러리 프레임워크

라이브러리	언어	특징
Scikit-learn	python	- 다양한 지도학습, 비지도학습 알고리즘이 구현된 라이브러리 - python기반으로 NumPy, SciPy이용한 고속화 지원
Statsmodels	python	- 검정 및 추정, 회귀분석, 시계열분석등 다양한 통계분석 기능 - 회귀분석의 경우 patsy 패키지 포함
NLTK	python	- http://www.nltk.org/ - Natural Language Toolkit약자로 자연어 처리 라이브러리
MLib	python,Java, Scala	- 아파치 스마크용 머신러닝 라이브러리
Weka	Java	- 자바로 쓰인 데이터마이닝 라이브러리 - GUI 제공
OpenCV	C/C++, Java, Python	- 2000년에 공개되었으며 이미지 데이터 처리 함수 풍부
Pytorch	python	- 페이스북 인공지능 연구팀에서 개발

ML & DL 살펴보기

◆ DL 라이브러리 프레임워크

라이브러리	언어	특징
Tensorflow	Python/ C++	<ul style="list-style-type: none">▪ 구글 AI 연구팀에서 개발한 오픈소스 라이브러리▪ 머신 러닝 & 딥 러닝 알고리즘 충족▪ 풍부한 자연어용 딥러닝 함수, 최신 데이터 모델 라이브러리 제공
Caffe	Python, Matlab	<ul style="list-style-type: none">• caffe.berkeleyvision.org)• 캘리포니아 버클리대 오픈소스로 개발된 딥러닝 프레임워크• 유연한 구조, 코드 확장성, 고속으로 처리가능, 커뮤니티 활성화• 미리 학습된 다양한 분류 모델 사용 가능
Torch	C, Lua	<ul style="list-style-type: none">▪ 뉴욕 대학에서 C, LuaJIT로 개발한 라이브러리▪ 스크립트 언어인 루아로 구현▪ 빠른 계산 속도 및 직관적인 API 제공

ML & DL 살펴보기

◆ DL 라이브러리 프레임워크

라이브러리	GPU	특징
CUDA	NVIDIA	<ul style="list-style-type: none">▪ NVIDIA에서 만든 GPU 플랫폼이자 API▪ 많은 딥러닝 라이브러리에서 CUDA 지원▪ NVIDIA의 GPU에서 사용
OpenCL	Intel, AMD	<ul style="list-style-type: none">▪ Apple에서 개발하고 Khronos Group에서 관리하는 연산 플랫폼▪ CPU/GPU 동시동작용 프로그램 개발 라이브러리▪ Intel/AMD 외 이기종 컴퓨터 표준 제공을 위해 개발▪ 많은 딥러닝 프레임워크에서 OpenCL 미지원

A decorative wavy line in yellow and white on the left side of the slide.

PART 2

개발환경구축

PYTHON ML&DL 환경구축

◆ python 명령어

- 커맨드 라인에서 사용
- 파이썬 파일 실행 및 패키지, 라이브러리 설치 등에 사용
- <https://docs.python.org/ko/3/using/cmdline.html>

PYTHON ML&DL 환경구축

◆ python 명령어

```
python [-기타옵션] [ -c command | -m module-name | script | - ] [args]
```

인터페이스 옵션	기능
-c command	command 문장 실행, 개행 문자로 여러 개의 command 가능
-m module_name	Module_name의 모듈을 sys.path에서 검색, __main__ 모듈로 실행 *확장자(py)제외
<script>	파이썬 파일이나 __main__.py 파일이 있는 디렉터리 실행

PYTHON ML&DL 환경구축

◆ python 명령어

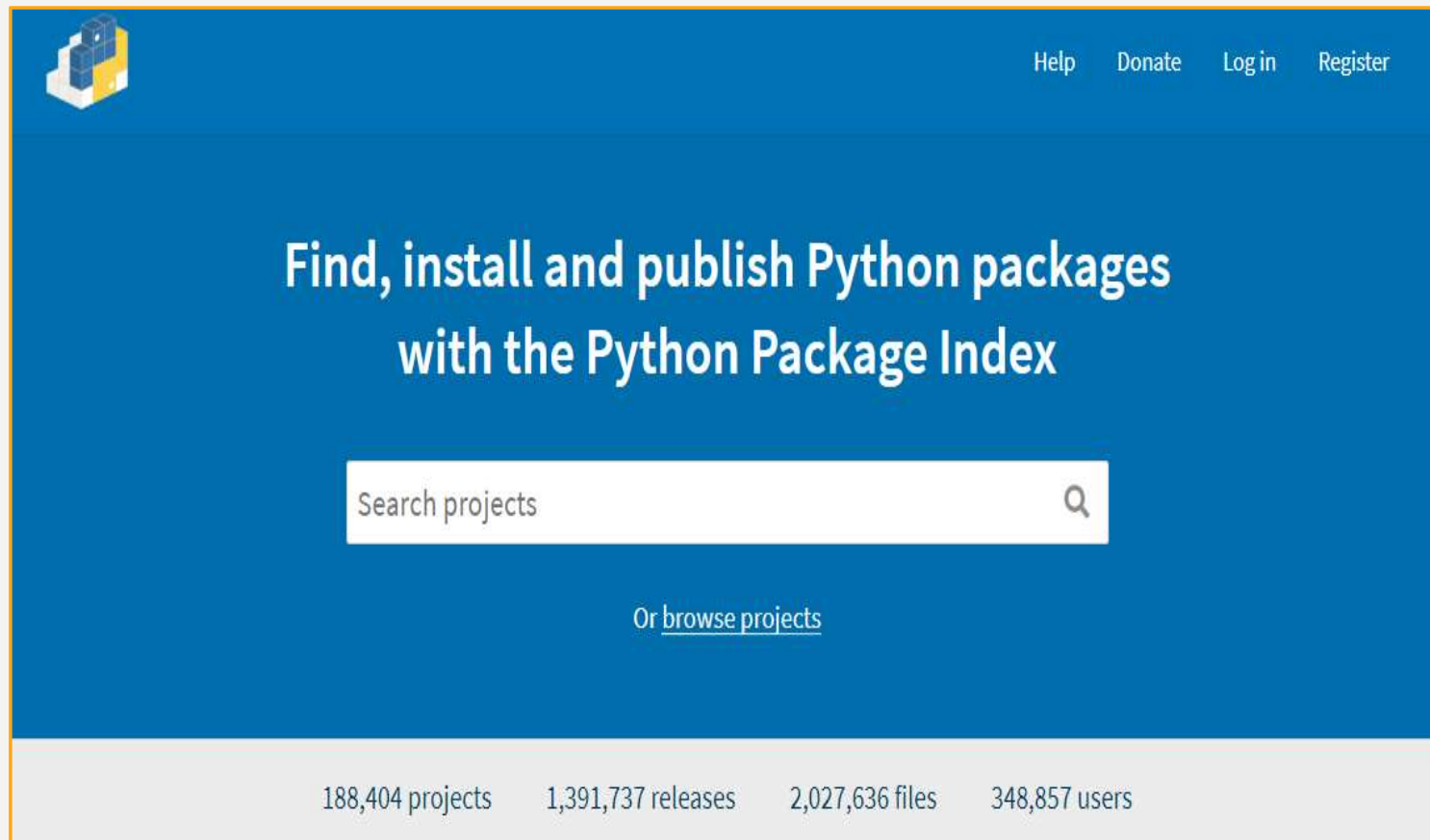
python [-기타옵션] [-c command | -m module-name | script | -] [args]

일반 옵션	기능
-? / -h / --help	python 명령어에 대한 설명 매뉴얼 출력
-v / --version	python 버전 정보 출력

PYTHON ML&DL 환경구축

◆ PyPI(Python Package Index)

- 파이썬 패키지들이 모여 있는 저장소
- <https://pypi.org/>



PYTHON ML&DL 환경구축

◆ PyPI(Python Package Index)

➤ PIP 명령어

- 파이썬으로 작성된 패키지 관리 시스템
- 파이썬 2.7.9 이후 버전과 3.4 이후 버전은 pip를 기본 포함

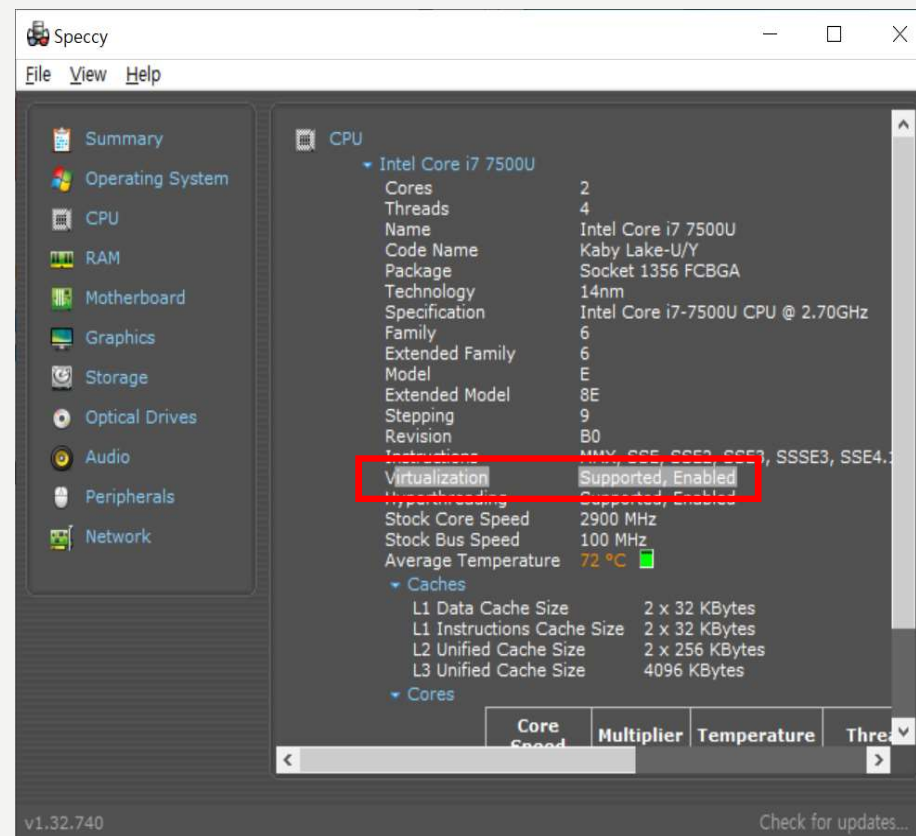
명령어	기능
pip help	pip 명령어 설명서 출력
pip list	현재 설치된 패키지 목록 확인 및 업데이트 필요 패키지 정보 제공
pip show package-name	패키지 정보 출력
pip install package-name	패키지 설치
pip install --upgrade package-name	패키지 설치 및 업그레이드
pip uninstall package-name	패키지 삭제

PYTHON ML&DL 환경구축

◆ PC 상태 확인

➤ Speccy 프로그램 실행

- CPU의 Virtualization 가능 여부 체크



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

- 수 백 개의 데이터 과학 파이썬 패키지 포함하고 있는 파이썬 배포판
- 패키지를 별도로 시간 들여 추가 설치하는 수고 감소
- 기능
 - 패키지 설치 및 관리
 - 가상환경 관리
- 관리 프로그램
 - conda 제공



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ conda 프로그램

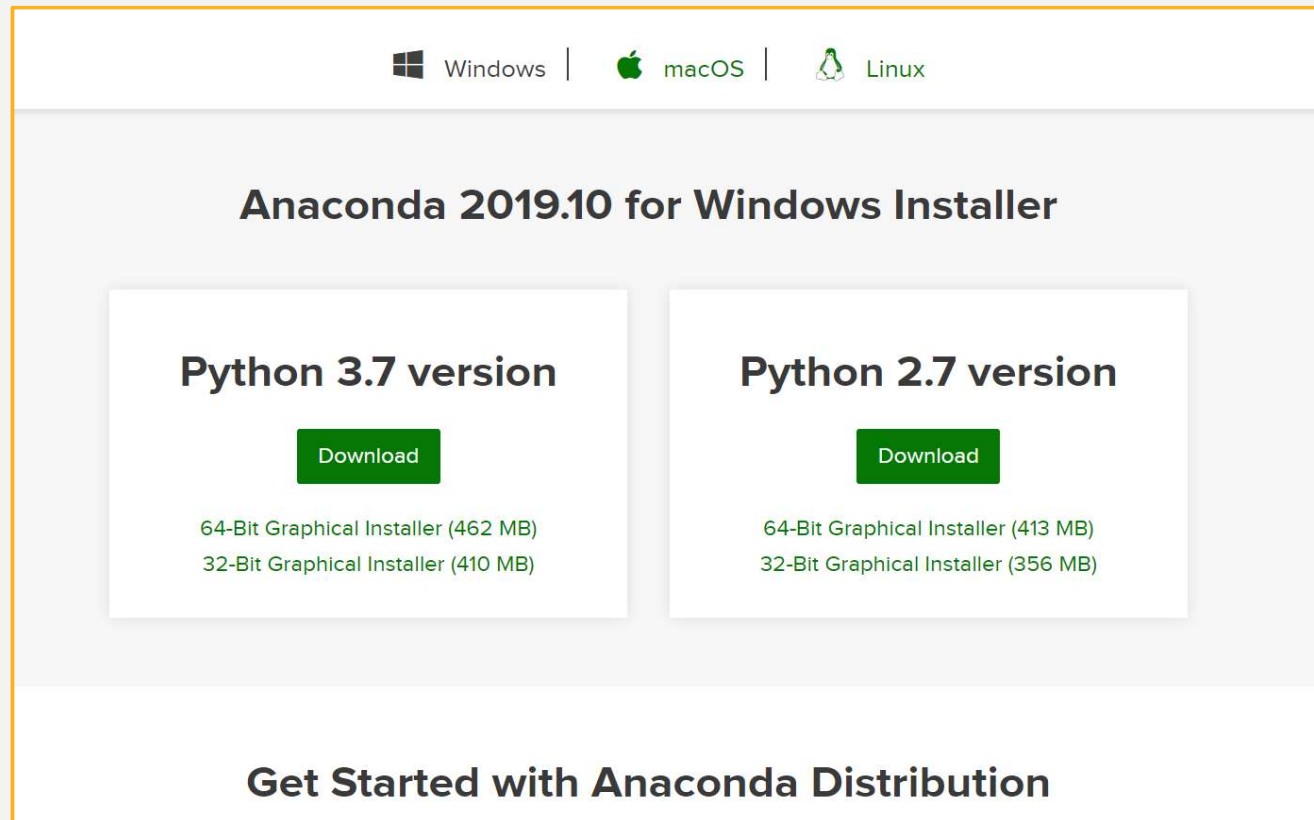
- 자동 의존성 체크 후 설치
- 모든 라이브러리 없으므로 pip와 병행 사용

기능	명령어
패키지 설치	\$ conda install numpy # numpy 설치
동시에 여러 패키지 설치	\$ conda install numpy scipy pandas # numpy, scipy, pandas 동시 설치
특정 버전 설치	\$ conda install numpy=1.10 # 특정 버전 설치
패키지 제거	\$ conda remove package_name
패키지 업데이트	\$ conda update package_name
모든 패키지 업데이트	\$ conda update --all
설치된 목록 출력	\$ conda list # 정보 확인
설치하려는 패키지 검색	\$ conda search search_term

PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

- 공식사이트 : <https://www.anaconda.com/distribution/>
 - Anaconda3-2019.10-Windows-x86_64.exe 다운로드



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ 저장소 : <https://repo.continuum.io/archive/>

- Anaconda3-2019.10-Windows-x86_64.exe 다운로드

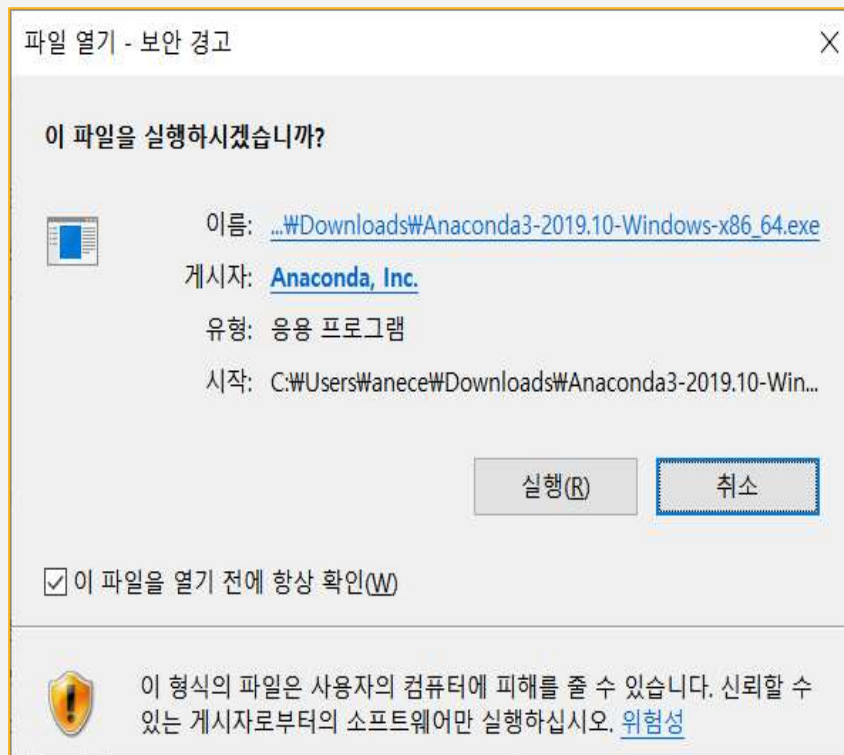
Anaconda installer archive

Filename	Size	Last Modified
Anaconda2-2019.10-Linux-ppc64le.sh	295.3M	2019-10-15 09:26:13
Anaconda2-2019.10-Linux-x86_64.sh	477.4M	2019-10-15 09:26:03
Anaconda2-2019.10-MacOSX-x86_64.pkg	635.7M	2019-10-15 09:27:30
Anaconda2-2019.10-MacOSX-x86_64.sh	408.8M	2019-10-15 09:27:31
Anaconda2-2019.10-Windows-x86.exe	355.6M	2019-10-15 09:26:15
Anaconda2-2019.10-Windows-x86_64.exe	412.8M	2019-10-15 09:26:08
Anaconda3-2019.10-Linux-ppc64le.sh	320.3M	2019-10-15 09:26:11
Anaconda3-2019.10-Linux-x86_64.sh	505.7M	2019-10-15 09:26:05
Anaconda3-2019.10-MacOSX-x86_64.pkg	653.5M	2019-10-15 09:27:33
Anaconda3-2019.10-MacOSX-x86_64.sh	424.2M	2019-10-15 09:27:31
Anaconda3-2019.10-Windows-x86.exe	409.6M	2019-10-15 09:26:10
Anaconda3-2019.10-Windows-x86_64.exe	461.5M	2019-10-15 09:27:17
Anaconda2-2019.07-Linux-ppc64le.sh	298.2M	2019-07-25 09:36:29

PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

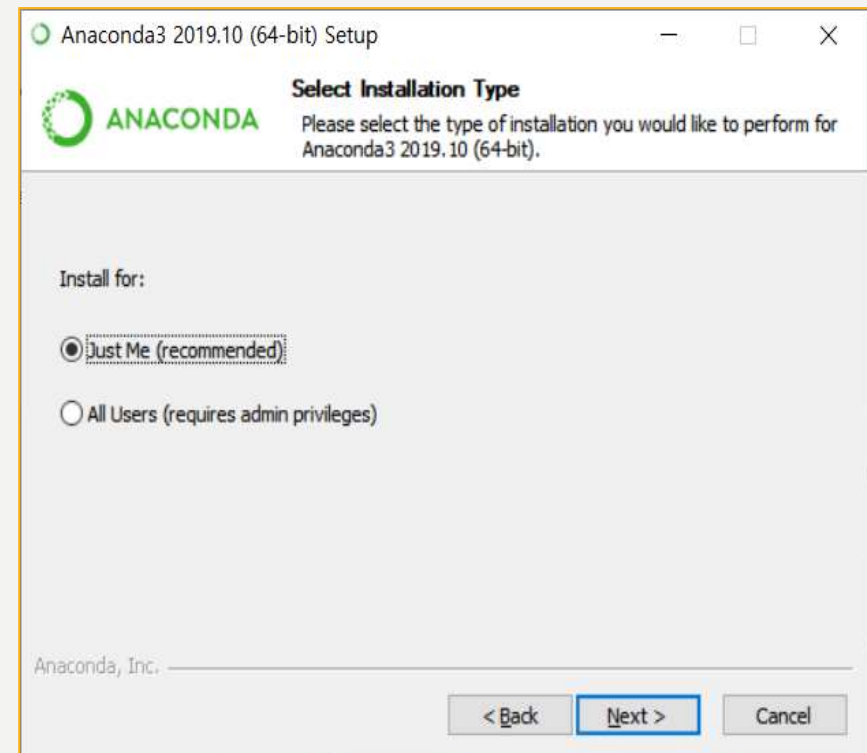
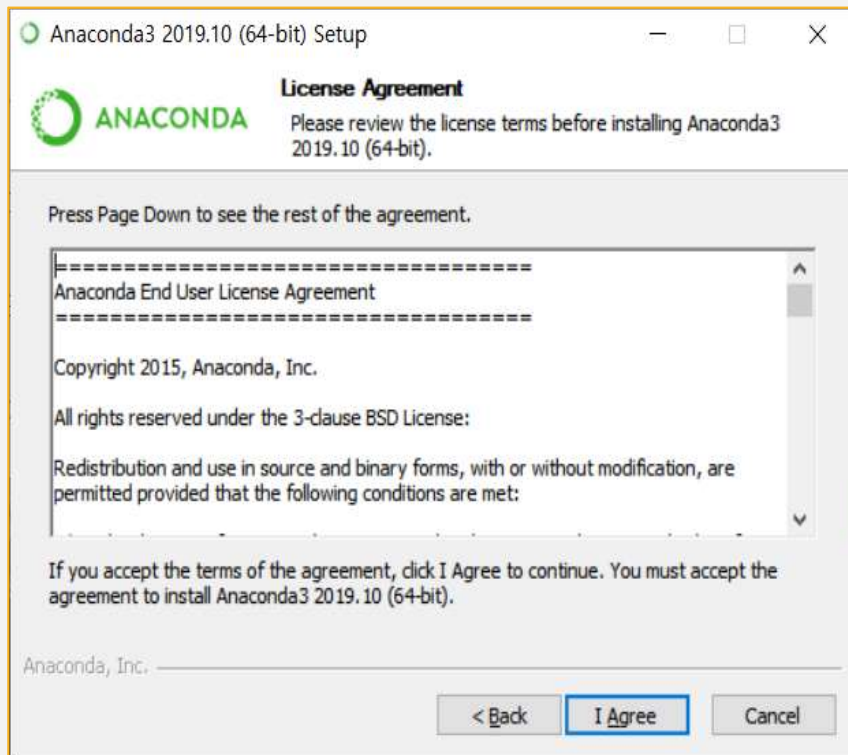
➤ Anaconda3-2019.10-Windows-x86_64.exe 실행



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

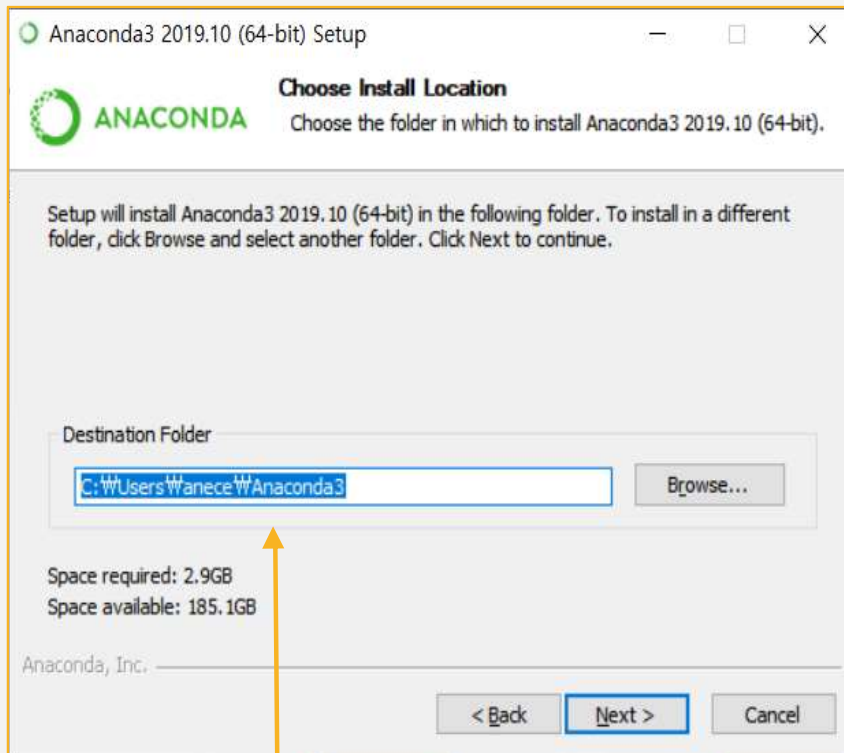
➤ Anaconda3-2019.10-Windows-x86_64.exe 실행



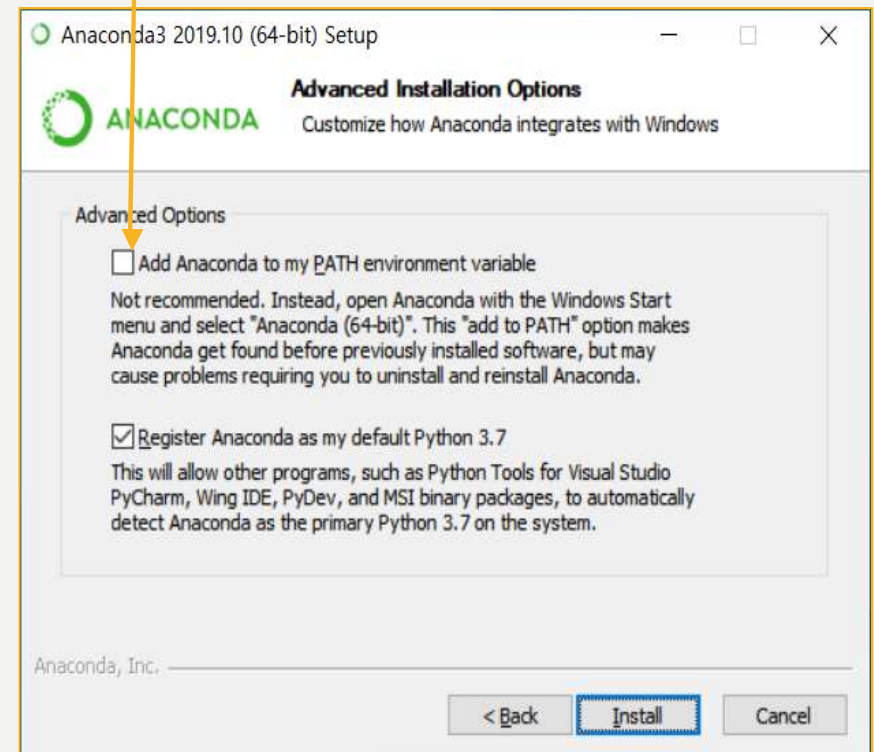
PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ Anaconda3-2019.10-Windows-x86_64.exe 실행



원하는 경로 변경 가능

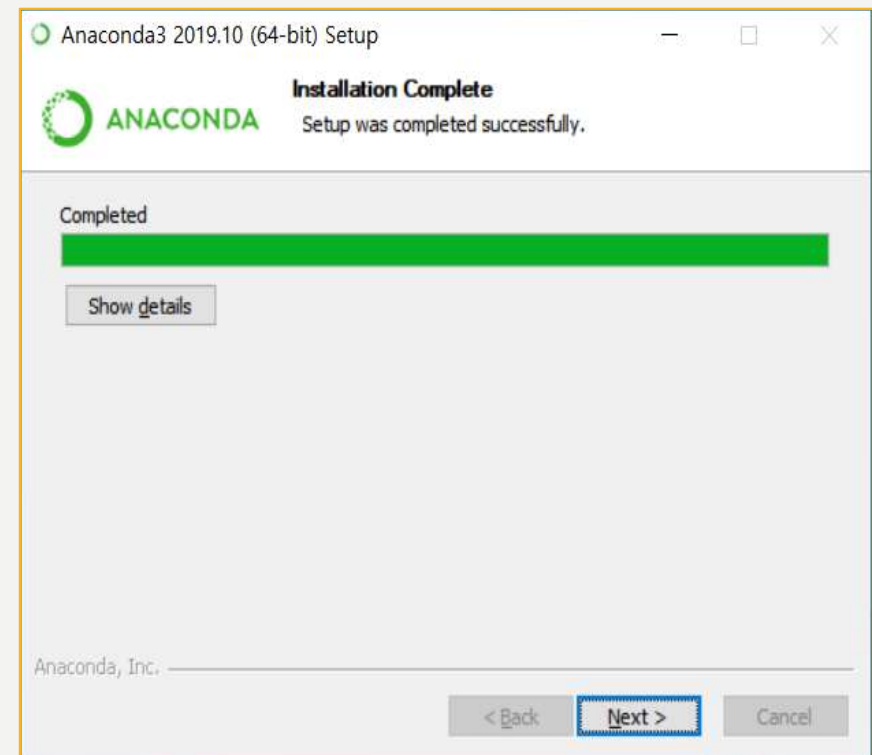
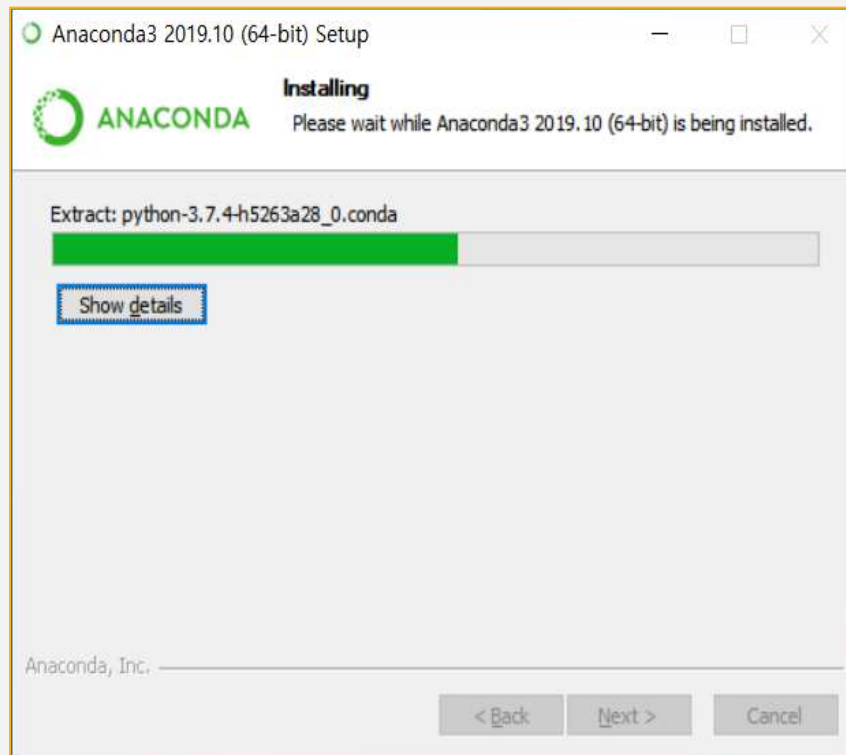


권장: 다른 SW 방해 할 수 있으므로 추가하지 마세요.

PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

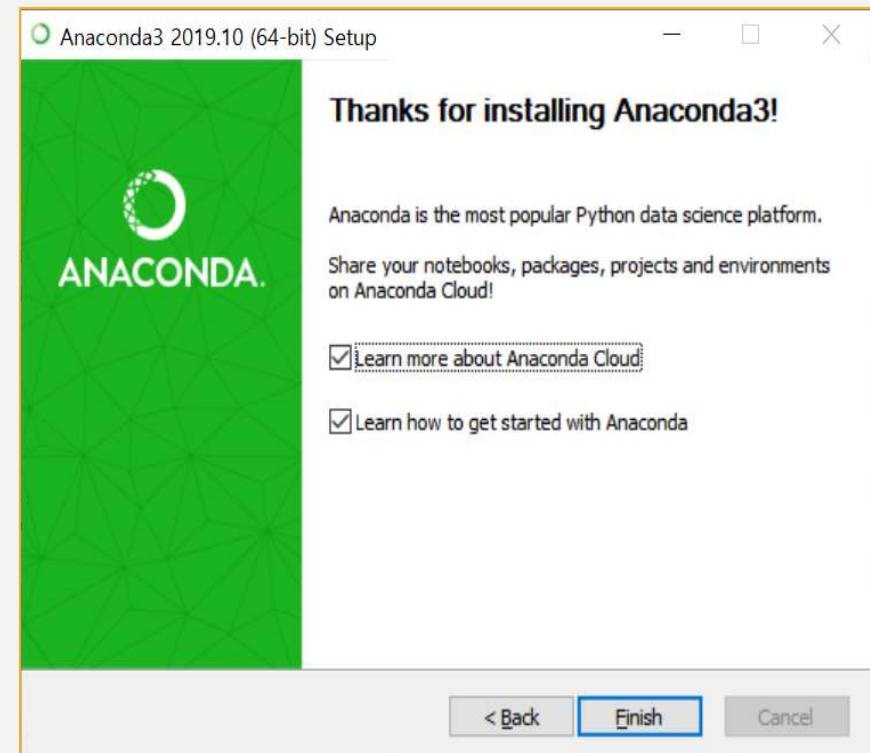
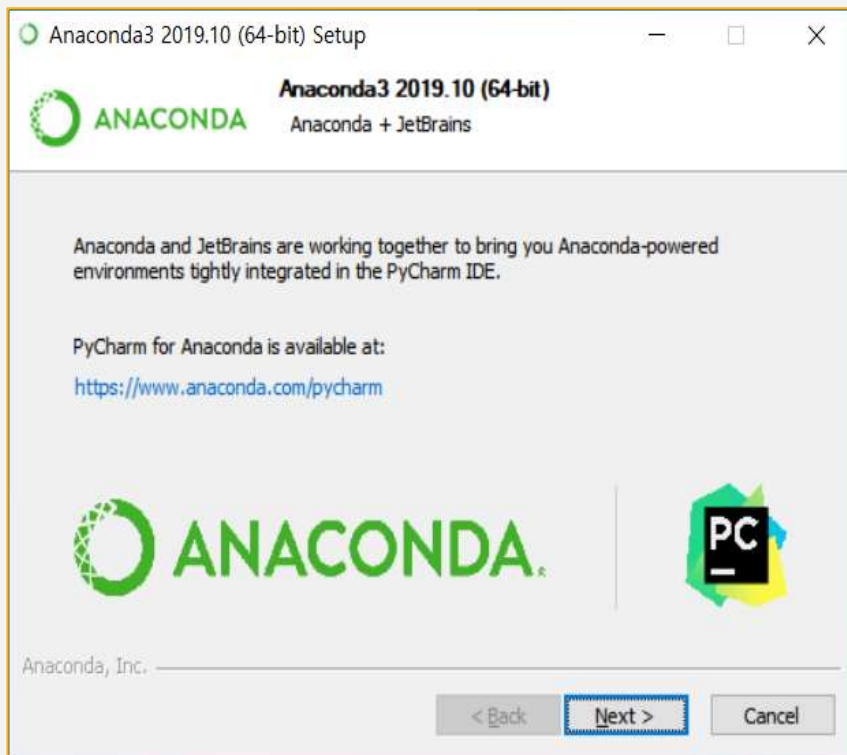
➤ Anaconda3-2019.10-Windows-x86_64.exe 실행



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ Anaconda3-2019.10-Windows-x86_64.exe 실행

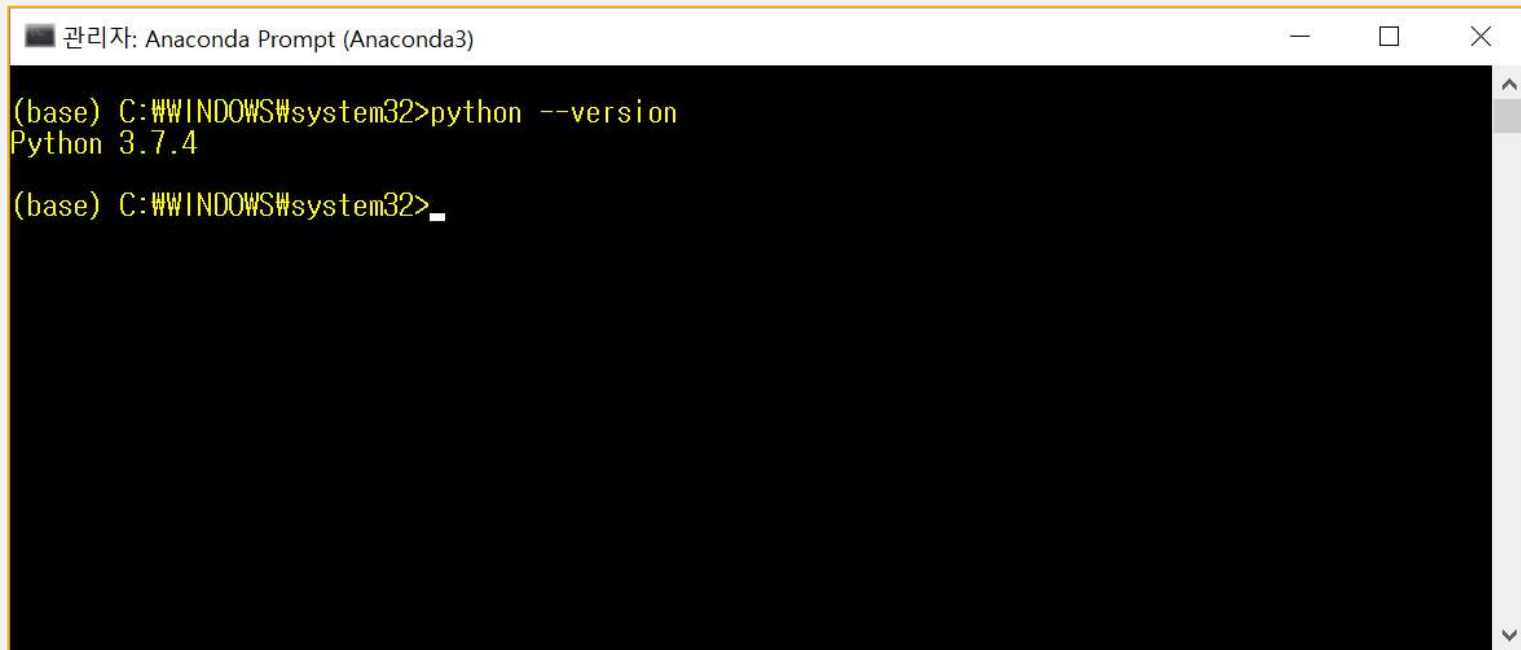


PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ Python 설치 버전 확인

- Anaconda Prompt 관리자 권한으로 실행
- `python --version` 입력 => 버전 확인



```
관리자: Anaconda Prompt (Anaconda3)

(base) C:\WINDOWS\system32>python --version
Python 3.7.4

(base) C:\WINDOWS\system32>_
```

PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ 업데이트

- conda 업데이트 : `conda update -n base conda`

```
관리자: Anaconda Prompt (Anaconda3) - conda update -n base conda

(base) C:\WINDOWS\system32>conda update -n base conda
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\Wanece\Anaconda3

added / updated specs:
- conda

The following packages will be downloaded:

package | build | size
-----|-----|-----
backports.functools_lru_cache-1.6.1 | py_0 | 11 KB
conda-4.8.0 | py37_1 | 2.8 MB
future-0.18.2 | py37_0 | 656 KB
-----|-----|-----
Total: | | 3.5 MB

The following packages will be UPDATED:

backports.functoo~ | 1.5-py_2 --> 1.6.1-py_0
conda | 4.7.12-py37_0 --> 4.8.0-py37_1
future | 0.17.1-py37_0 --> 0.18.2-py37_0

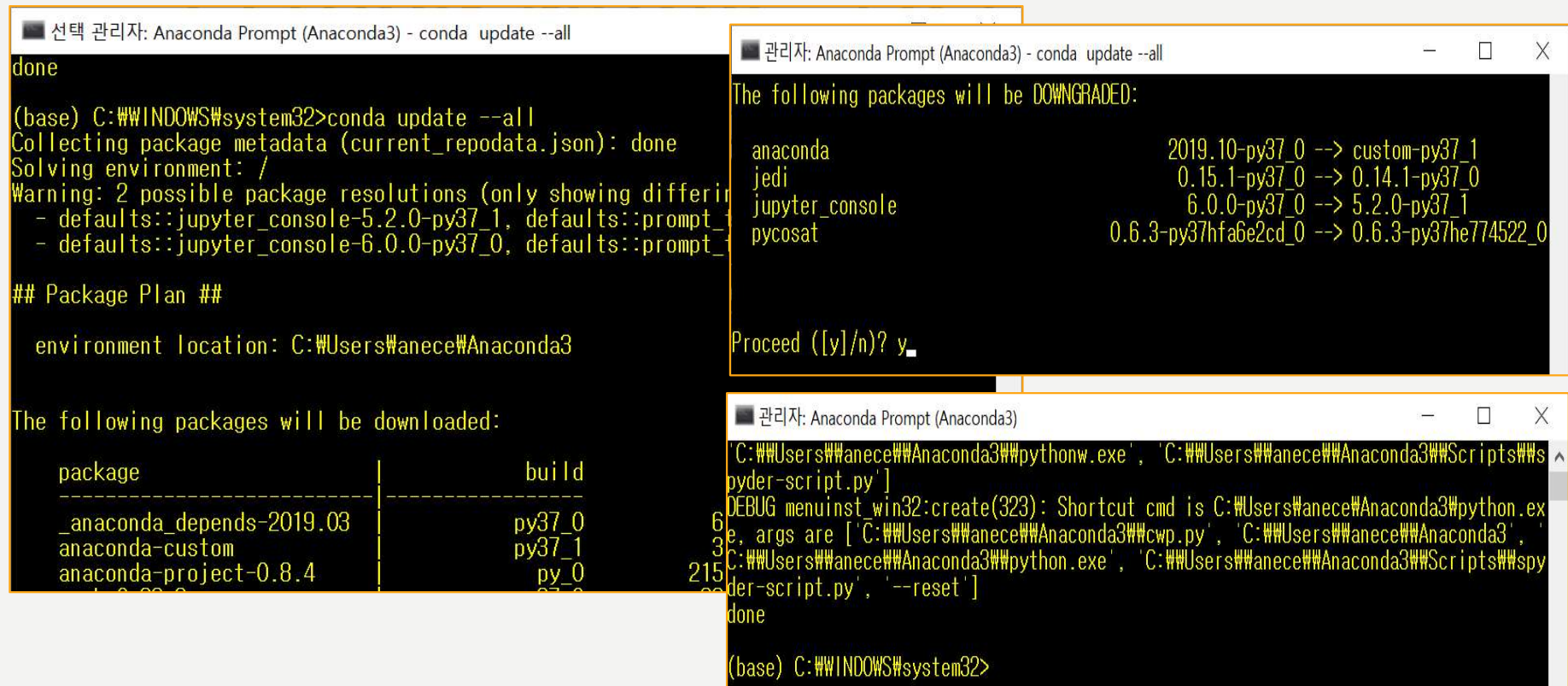
Proceed ([y]/n)? y_
```

PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ 업데이트

- 패키지 업데이트 : `conda update --all`



The first screenshot shows the command `conda update --all` being executed in the Anaconda Prompt. The output indicates that the environment is being solved and that two possible package resolutions are shown. The second screenshot shows the package plan, listing the packages to be downloaded: `_anaconda_depends-2019.03`, `anaconda-custom`, and `anaconda-project-0.8.4`. The third screenshot shows the packages to be downgraded, including `anaconda`, `jedi`, `jupyter_console`, and `pycosat`. The user is prompted to proceed with the update, and the final screenshot shows the update process completing successfully.

```
선택 관리자: Anaconda Prompt (Anaconda3) - conda update --all
done
(base) C:\WINDOWS\system32>conda update --all
Collecting package metadata (current_repodata.json): done
Solving environment: /
Warning: 2 possible package resolutions (only showing differing ones)
- defaults::jupyter_console-5.2.0-py37_1, defaults::prompt-toolkit-0.10.0-py37_0
- defaults::jupyter_console-6.0.0-py37_0, defaults::prompt-toolkit-0.10.0-py37_0

## Package Plan ##

environment location: C:\Users\anece\Anaconda3

The following packages will be downloaded:

package                                     build
-----
_anaconda_depends-2019.03                  py37_0
anaconda-custom                            py37_1
anaconda-project-0.8.4                     py_0

The following packages will be DOWNGRADED:

anaconda                                2019.10-py37_0 --> custom-py37_1
jedi                                    0.15.1-py37_0 --> 0.14.1-py37_0
jupyter_console                         6.0.0-py37_0 --> 5.2.0-py37_1
pycosat                                0.6.3-py37hfaf6e2cd_0 --> 0.6.3-py37he774522_0

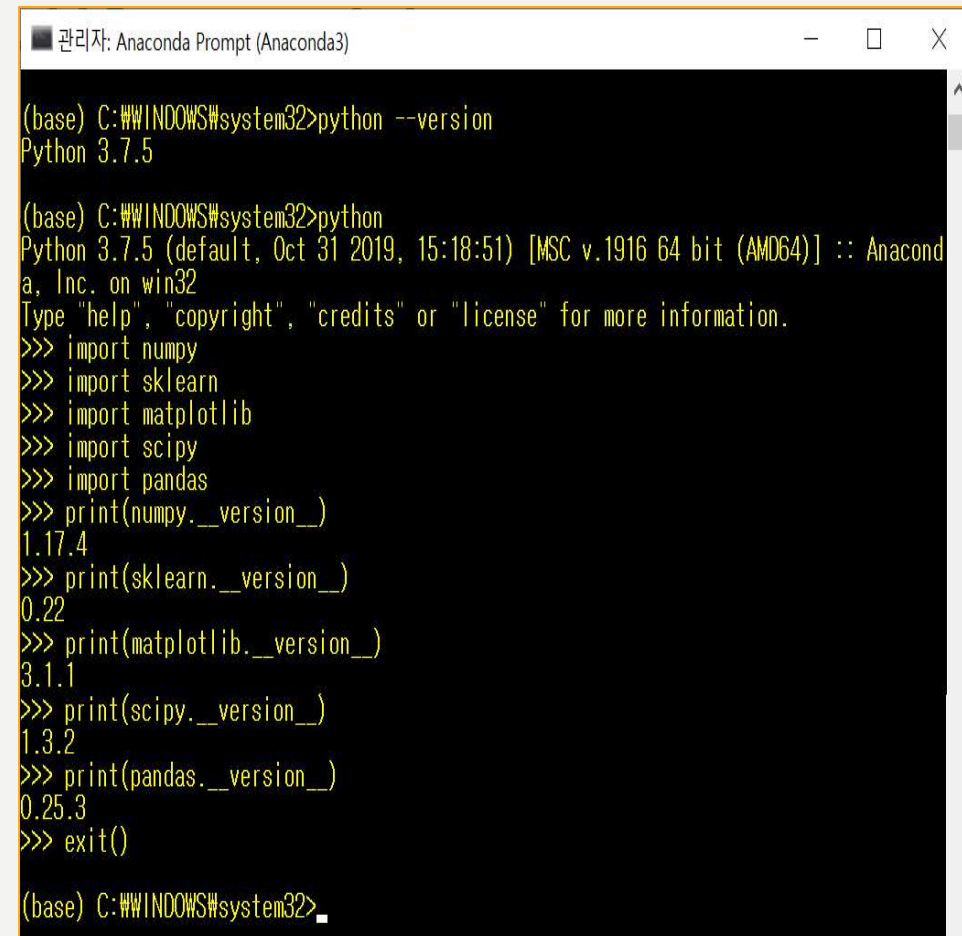
Proceed ([y]/n)? y

'C:\Users\anece\Anaconda3\pythonw.exe', 'C:\Users\anece\Anaconda3\Scripts\pythonw-script.py'
DEBUG menuinst.win32.create(323): Shortcut cmd is C:\Users\anece\Anaconda3\pythonw.exe, args are ['C:\Users\anece\Anaconda3\pythonw.exe', 'C:\Users\anece\Anaconda3\Scripts\pythonw-script.py', '--reset']
done
(base) C:\WINDOWS\system32>
```

PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ 업데이트 후 설치 라이브러리 확인

A screenshot of the Anaconda Prompt window. The title bar reads '관리자: Anaconda Prompt (Anaconda3)'. The command prompt shows the user running 'python --version' which returns 'Python 3.7.5'. Then, the user runs 'python' to enter the interactive shell. Inside the shell, several libraries are imported: numpy, sklearn, matplotlib, scipy, and pandas. The version of each library is printed using the 'print(library.__version__)' command. The versions shown are: numpy 1.17.4, sklearn 0.22, matplotlib 3.1.1, scipy 1.3.2, and pandas 0.25.3. Finally, the user runs 'exit()' to leave the shell, returning to the command prompt.

```
(base) C:\WINDOWS\system32>python --version
Python 3.7.5

(base) C:\WINDOWS\system32>python
Python 3.7.5 (default, Oct 31 2019, 15:18:51) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> import sklearn
>>> import matplotlib
>>> import scipy
>>> import pandas
>>> print(numpy.__version__)
1.17.4
>>> print(sklearn.__version__)
0.22
>>> print(matplotlib.__version__)
3.1.1
>>> print(scipy.__version__)
1.3.2
>>> print(pandas.__version__)
0.25.3
>>> exit()

(base) C:\WINDOWS\system32>
```

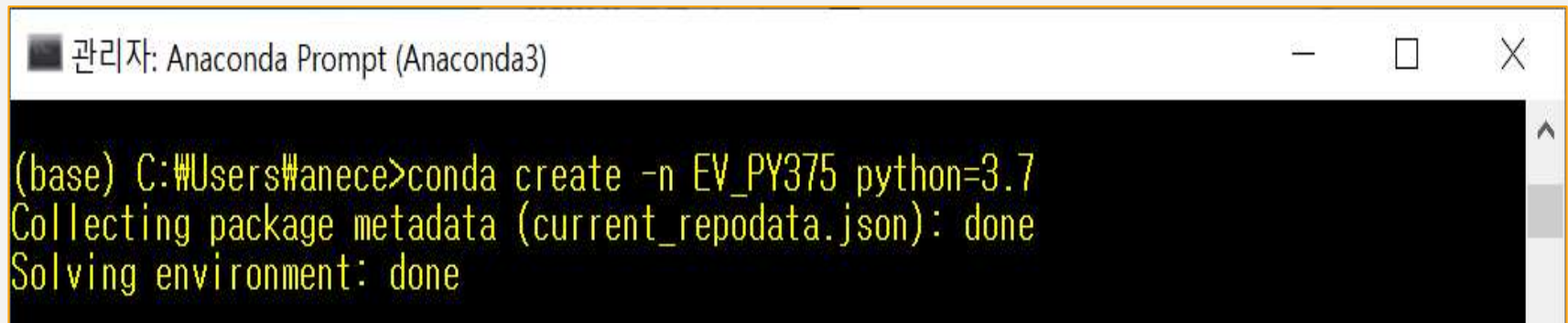
PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ 가상환경 만들기

`conda create -n 가상환경이름 python=파이썬버전`

(base) C:\Users\사용자계정> `conda create -n EV_PY375 python=3.7`



```
관리자: Anaconda Prompt (Anaconda3)

(base) C:\Users\anece>conda create -n EV_PY375 python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done
```


PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ 가상환경 만들기

```
## Package Plan ##

environment location: C:\Users\anece\Anaconda3\envs\EV_PY375

added / updated specs:
- python=3.7

The following packages will be downloaded:

package                        | build                | 58 KB
-----|-----
wheel-0.33.6                   | py37_0               | 58 KB
Total:                          |                      | 58 KB

The following NEW packages will be INSTALLED:

ca-certificates  pkgs/main/win-64::ca-certificates-2019.11.27-0
certifi          pkgs/main/win-64::certifi-2019.11.28-py37_0
openssl          pkgs/main/win-64::openssl-1.1.1d-he774522_3
pip              pkgs/main/win-64::pip-19.3.1-py37_0
python           pkgs/main/win-64::python-3.7.5-h8c8aaf0_0
setuptools       pkgs/main/win-64::setuptools-42.0.2-py37_0
sqlite           pkgs/main/win-64::sqlite-3.30.1-he774522_0
vc               pkgs/main/win-64::vc-14.1-h0510ff6_4
vs2015_runtime   pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_1
wheel            pkgs/main/win-64::wheel-0.33.6-py37_0
wincertstore     pkgs/main/win-64::wincertstore-0.2-py37_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
wheel-0.33.6 | 58 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate EV_PY375
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```


PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ 가상환경 확인

`conda env list`

```
(base) C:\Users\anece>conda env list
# conda environments:
#
base          * C:\Users\anece\Anaconda3
EV_PY375      C:\Users\anece\Anaconda3\envs\EV_PY375

(base) C:\Users\anece>
```

기본 생성 root에 해당

새로 생성된 가상환경

PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ 가상환경 실행 & 종료

실행 : `conda activate` 가상환경이름

종료: `conda deactivate`

```
(base) C:\Users\Wanece>conda activate EV_PY375
```

```
(EV_PY375) C:\Users\Wanece>conda deactivate
```

```
(base) C:\Users\Wanece>
```

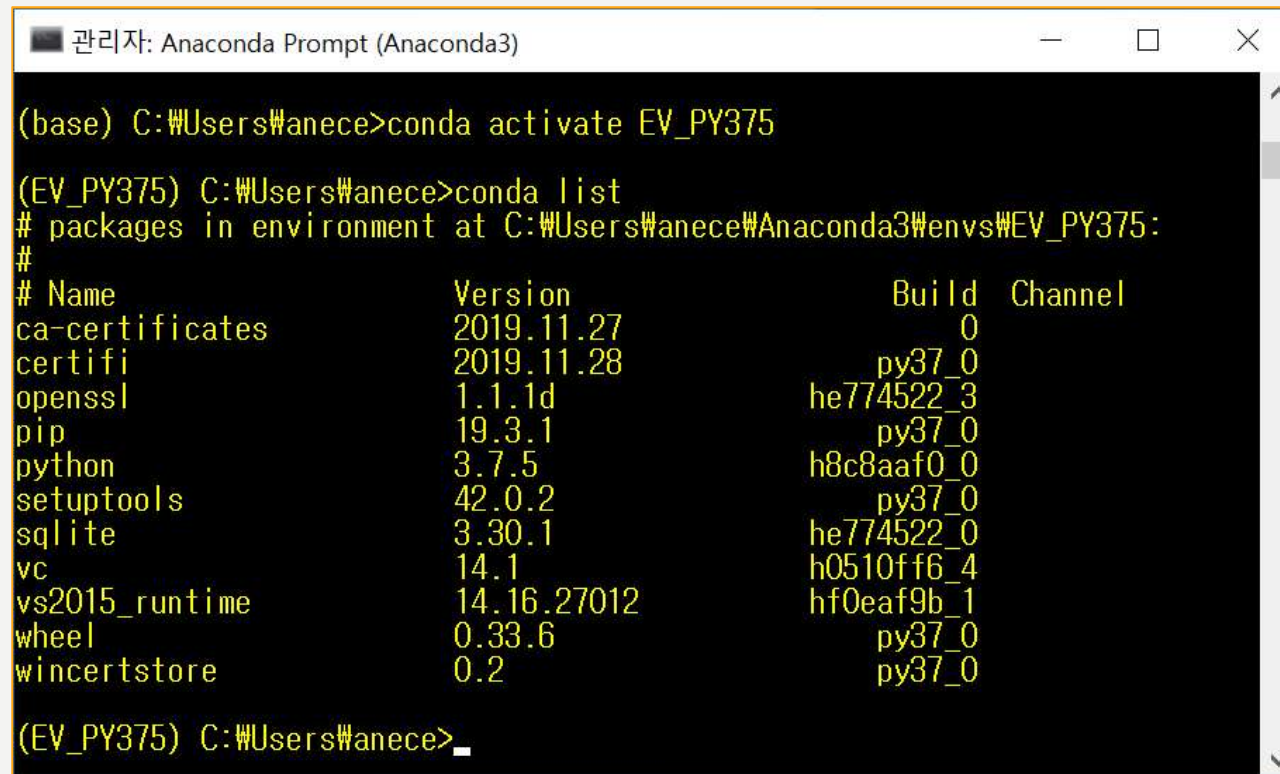
PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ 가상환경 설치 모듈(라이브러리) 체크

설치 모듈 정보 출력 : `conda list`

모듈 설치 명령어 : `conda install 라이브러리명`



```
관리자: Anaconda Prompt (Anaconda3)

(base) C:\Users\Wanece>conda activate EV_PY375

(EV_PY375) C:\Users\Wanece>conda list
# packages in environment at C:\Users\Wanece\Anaconda3\envs\EV_PY375:
#
# Name                        Version      Build    Channel
ca-certificates              2019.11.27   0
certifi                      2019.11.28   py37_0
openssl                      1.1.1d       he774522_3
pip                          19.3.1       py37_0
python                      3.7.5        h8c8aaf0_0
setuptools                   42.0.2       py37_0
sqlite                       3.30.1       he774522_0
vc                           14.1         h0510ff6_4
vs2015_runtime               14.16.27012  hf0eaf9b_1
wheel                        0.33.6       py37_0
wincertstore                  0.2          py37_0

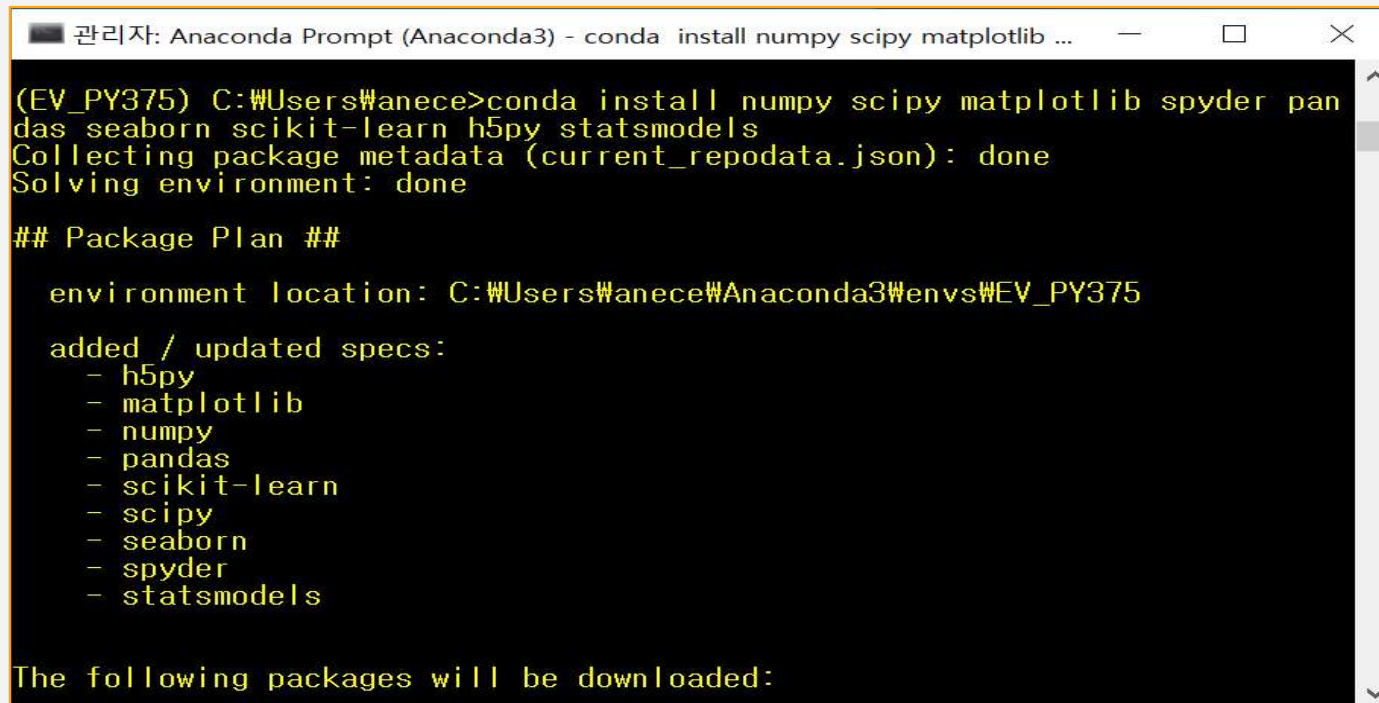
(EV_PY375) C:\Users\Wanece>_
```

PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ 가상환경 설치 모듈(라이브러리) 설치

```
conda install numpy scipy matplotlib  
spyder pandas seaborn  
scikit-learn h5py statsmodels
```



```
관리자: Anaconda Prompt (Anaconda3) - conda install numpy scipy matplotlib ...  
(EV_PY375) C:\Users\Wanece>conda install numpy scipy matplotlib spyder pandas  
seaborn scikit-learn h5py statsmodels  
Collecting package metadata (current_repodata.json): done  
Solving environment: done  
  
## Package Plan ##  
  
environment location: C:\Users\Wanece\Anaconda3\envs\EV_PY375  
  
added / updated specs:  
- h5py  
- matplotlib  
- numpy  
- pandas  
- scikit-learn  
- scipy  
- seaborn  
- spyder  
- statsmodels  
  
The following packages will be downloaded:
```

PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda)

➤ 가상환경 설치 모듈(라이브러리) 설치

```
관리자: Anaconda Prompt (Anaconda3) - conda install numpy scipy matplotlib ...
win_inet_pton      pkgs/main/win-64::win_inet_pton-1.1.0-py37_0
wrap               pkgs/main/win-64::wrap-1.11.2-py37he774522_0
yaml               pkgs/main/win-64::yaml-0.1.7-hc54c509_2
yapf               pkgs/main/noarch::yapf-0.28.0-py_0
zeromq             pkgs/main/win-64::zeromq-4.3.1-h33f27b4_3
zipp               pkgs/main/noarch::zipp-0.6.0-py_0
zlib               pkgs/main/win-64::zlib-1.2.11-h62dcd97_3

Proceed ([y]/n)? y_
```

```
관리자: Anaconda Prompt (Anaconda3)
.exe', 'C:\\Users\\anece\\Anaconda3\\envs\\EV_PY375\\Scripts\\spyder-script.py'
, '--reset']
done

(EV_PY375) C:\\Users\\anece>conda list
# packages in environment at C:\\Users\\anece\\Anaconda3\\envs\\EV_PY375:
#
# Name                Version           Build    Channel
alabaster              0.7.12           py37_0
argh                   0.26.2           py37_0
asn1crypto             1.2.0            py37_0
```

PYTHON ML&DL 환경구축

◆ PyCharm IDE 설치

➤ JetBrains사에서 개발한 Python용 통합개발환경 SW

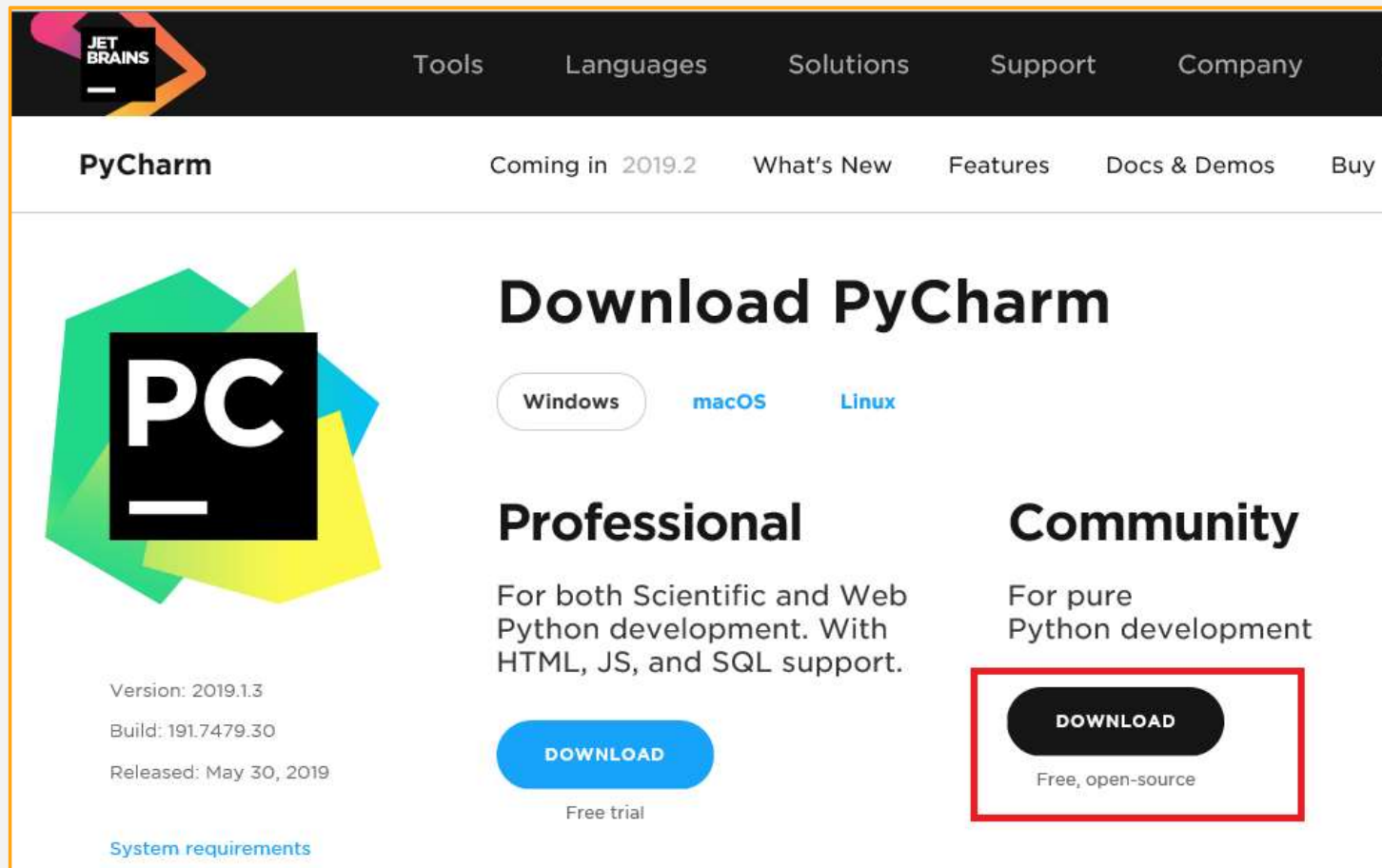
<https://www.jetbrains.com/pycharm/>



PYTHON ML&DL 환경구축

◆ PyCharm IDE 설치

- Professional → 유료 Community → 무료



The screenshot shows the PyCharm download page. At the top, there's a navigation bar with links: Tools, Languages, Solutions, Support, and Company. Below this, the main header includes 'PyCharm', 'Coming in 2019.2', 'What's New', 'Features', 'Docs & Demos', and 'Buy'. The main content area is titled 'Download PyCharm'. On the left, there's a large 'PC' logo. To the right, there are three tabs: 'Windows', 'macOS', and 'Linux'. Below these, there are two columns: 'Professional' and 'Community'. The 'Professional' column describes it as 'For both Scientific and Web Python development. With HTML, JS, and SQL support.' and has a blue 'DOWNLOAD' button with 'Free trial' text below it. The 'Community' column describes it as 'For pure Python development' and has a red-bordered box containing a black 'DOWNLOAD' button and the text 'Free, open-source' below it. On the far left, there's a sidebar with version information: 'Version: 2019.1.3', 'Build: 191.7479.30', 'Released: May 30, 2019', and a link for 'System requirements'.

PyCharm

Coming in 2019.2 What's New Features Docs & Demos Buy

Download PyCharm

Windows macOS Linux

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

DOWNLOAD

Free trial

Community

For pure Python development

DOWNLOAD

Free, open-source

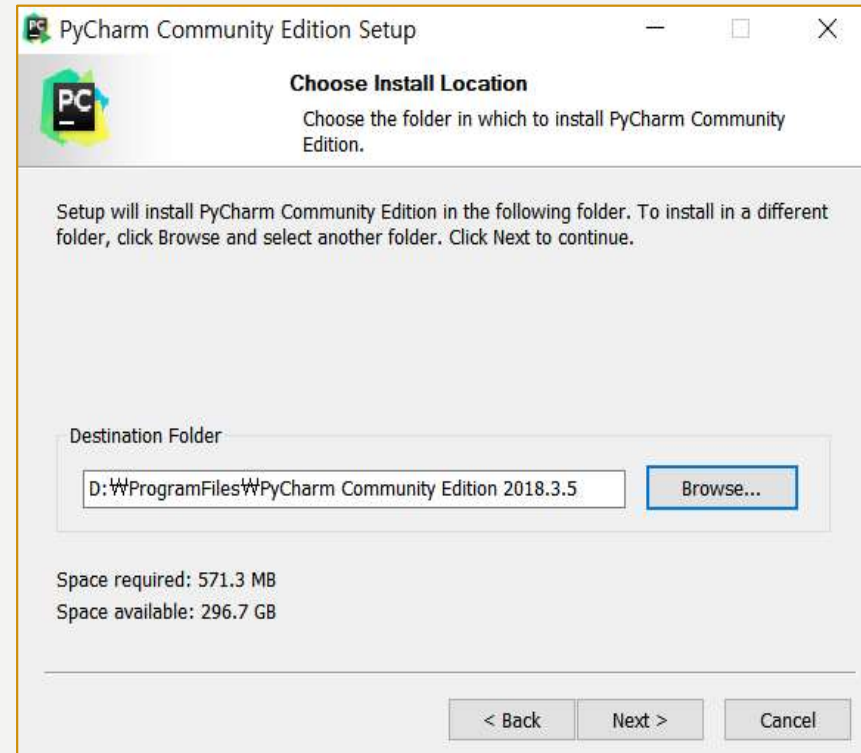
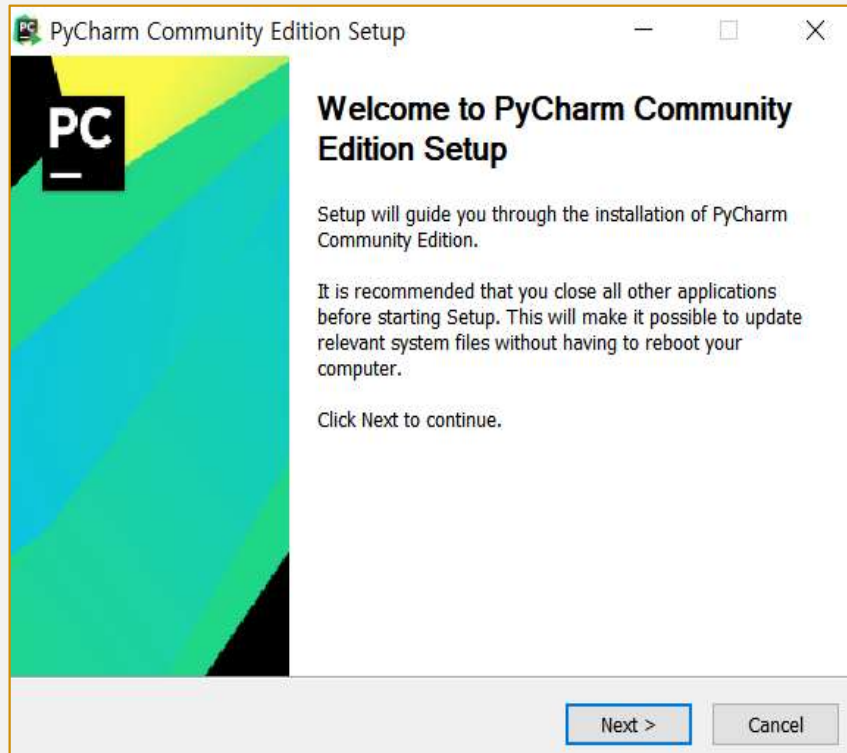
Version: 2019.1.3
Build: 191.7479.30
Released: May 30, 2019

[System requirements](#)

PYTHON ML&DL 환경구축

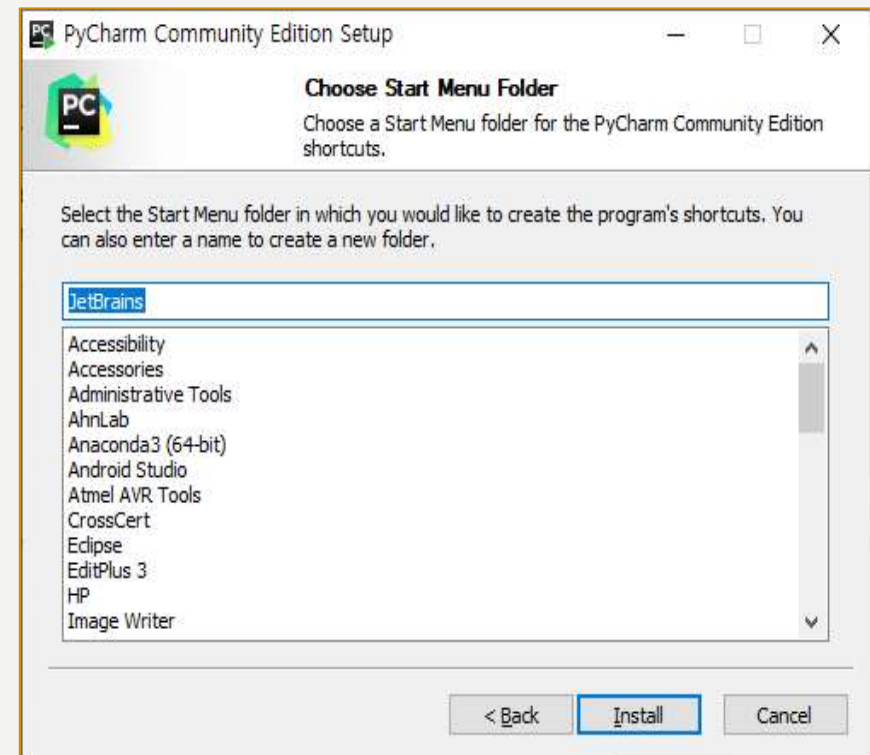
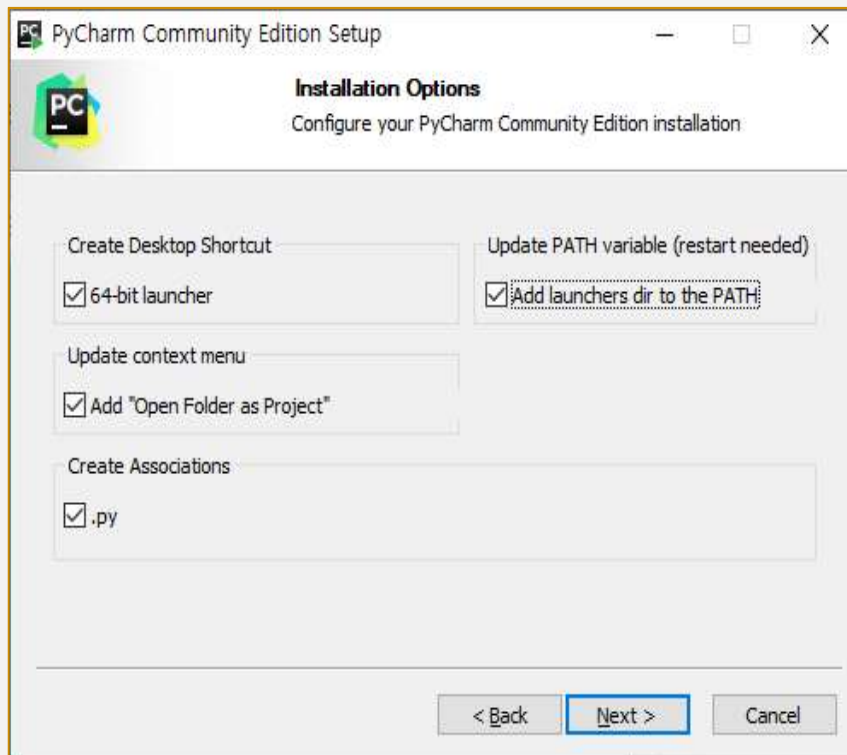
◆ PyCharm IDE 설치

➤ pycharm-community-2019.1.3.exe 클릭해서 실행



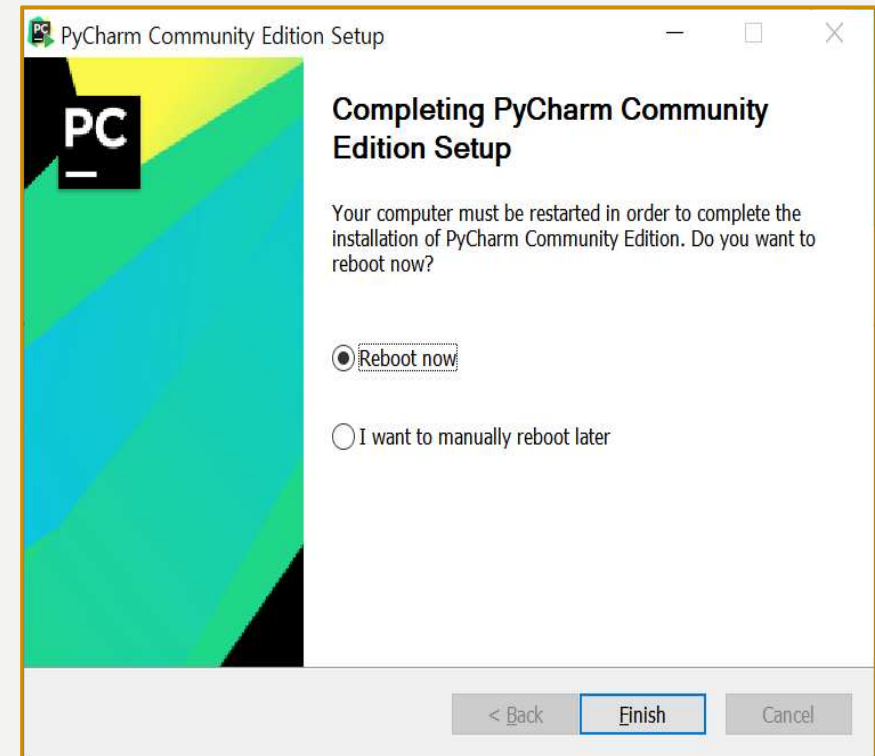
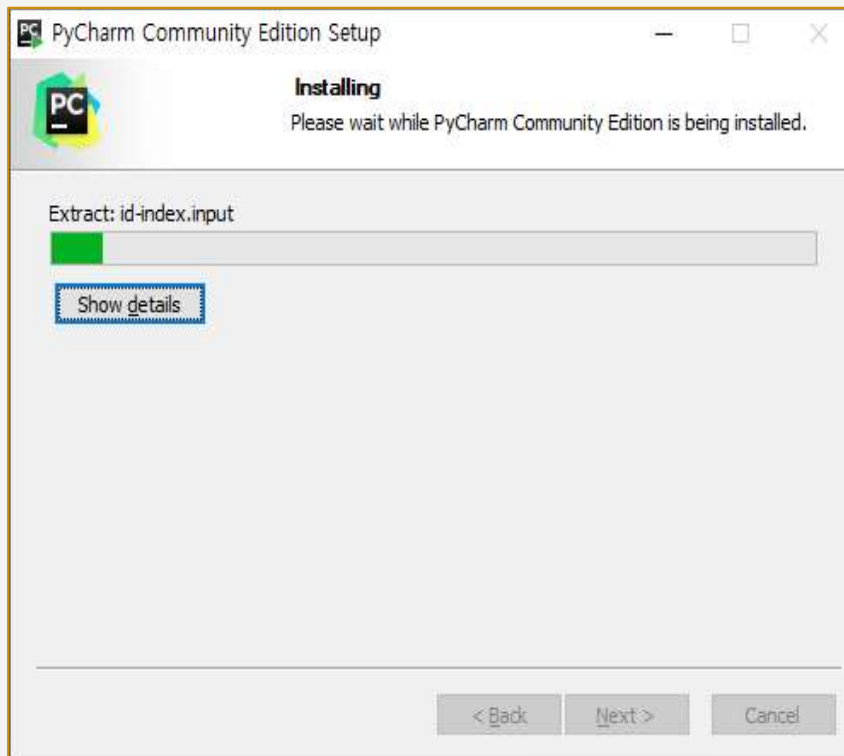
PYTHON ML&DL 환경구축

◆ PyCharm IDE 설치



PYTHON ML&DL 환경구축

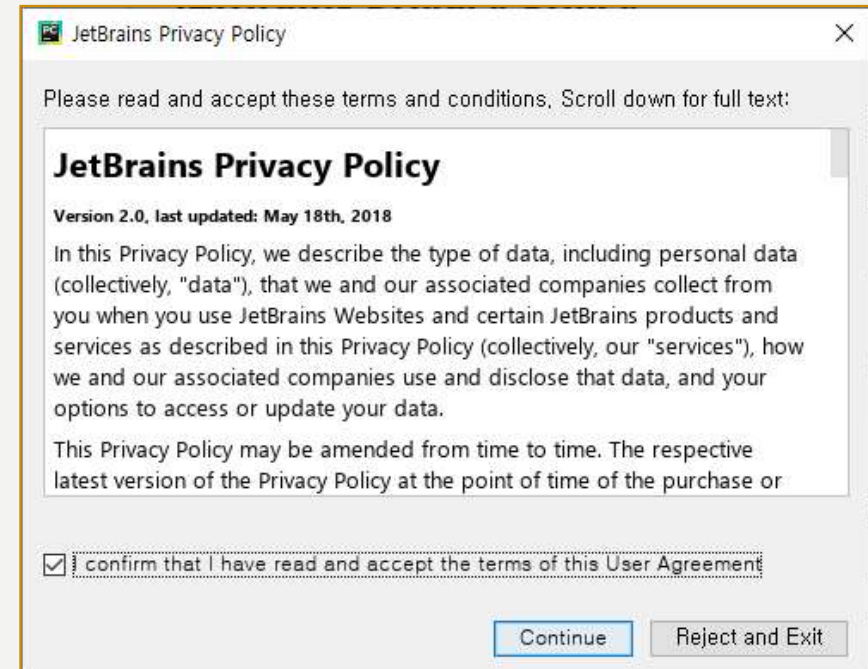
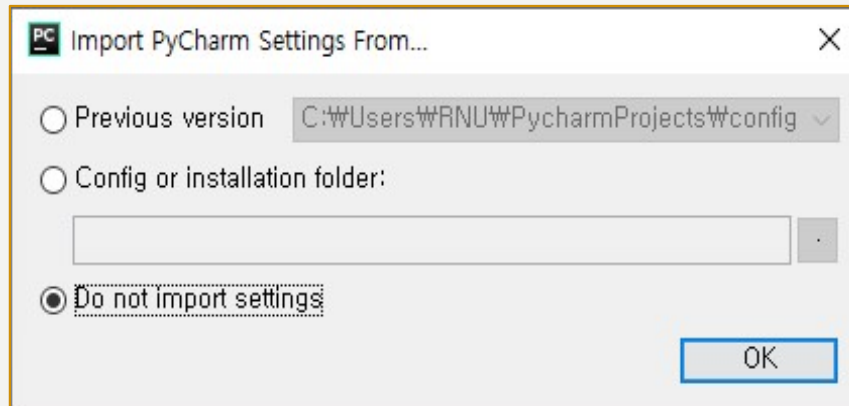
◆ PyCharm IDE 설치



PYTHON ML&DL 환경구축

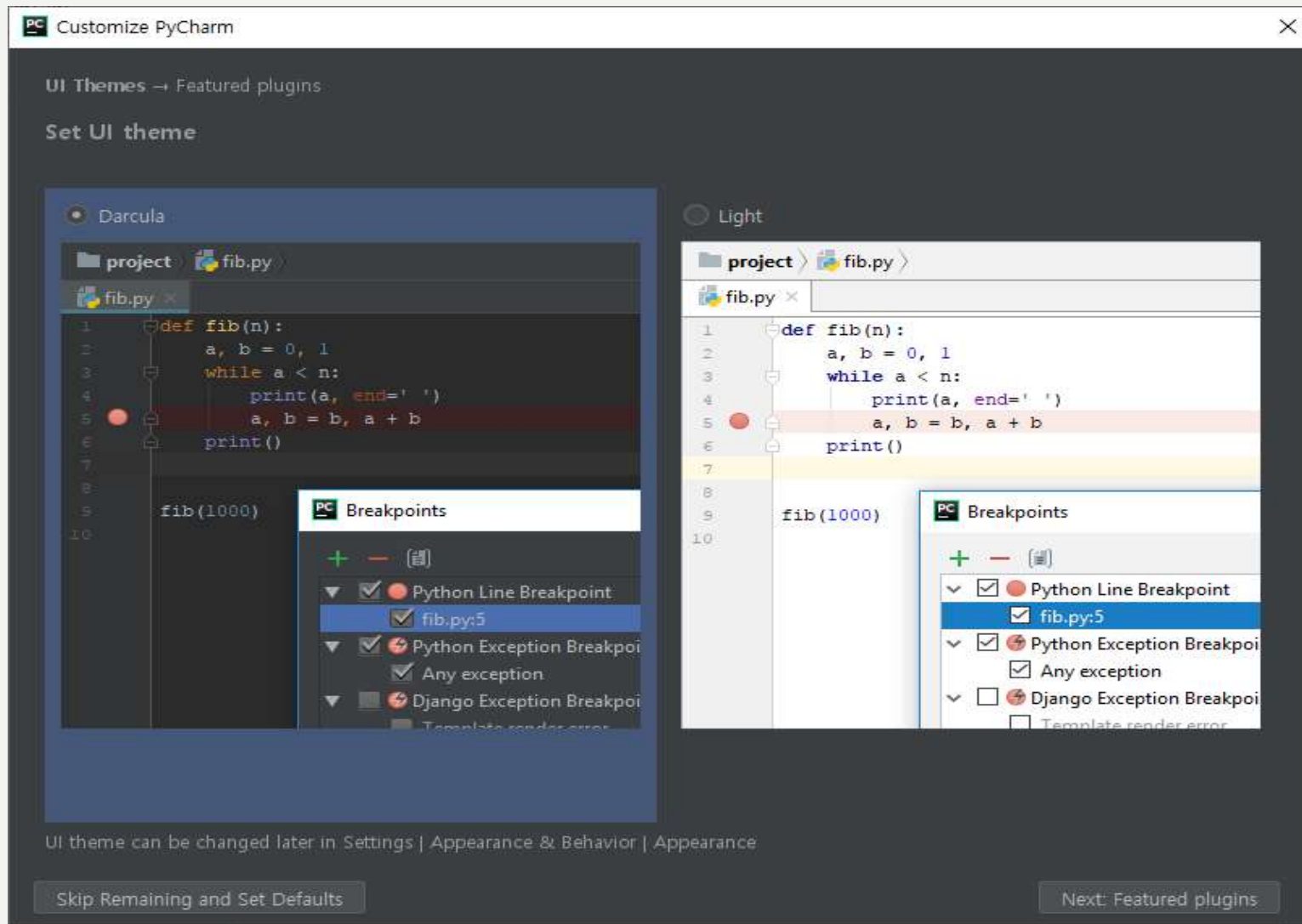
◆ PyCharm IDE 설치

➤ 설정 및 라이선서



PYTHON ML&DL 환경구축

◆ PyCharm IDE - UI 테마 설정

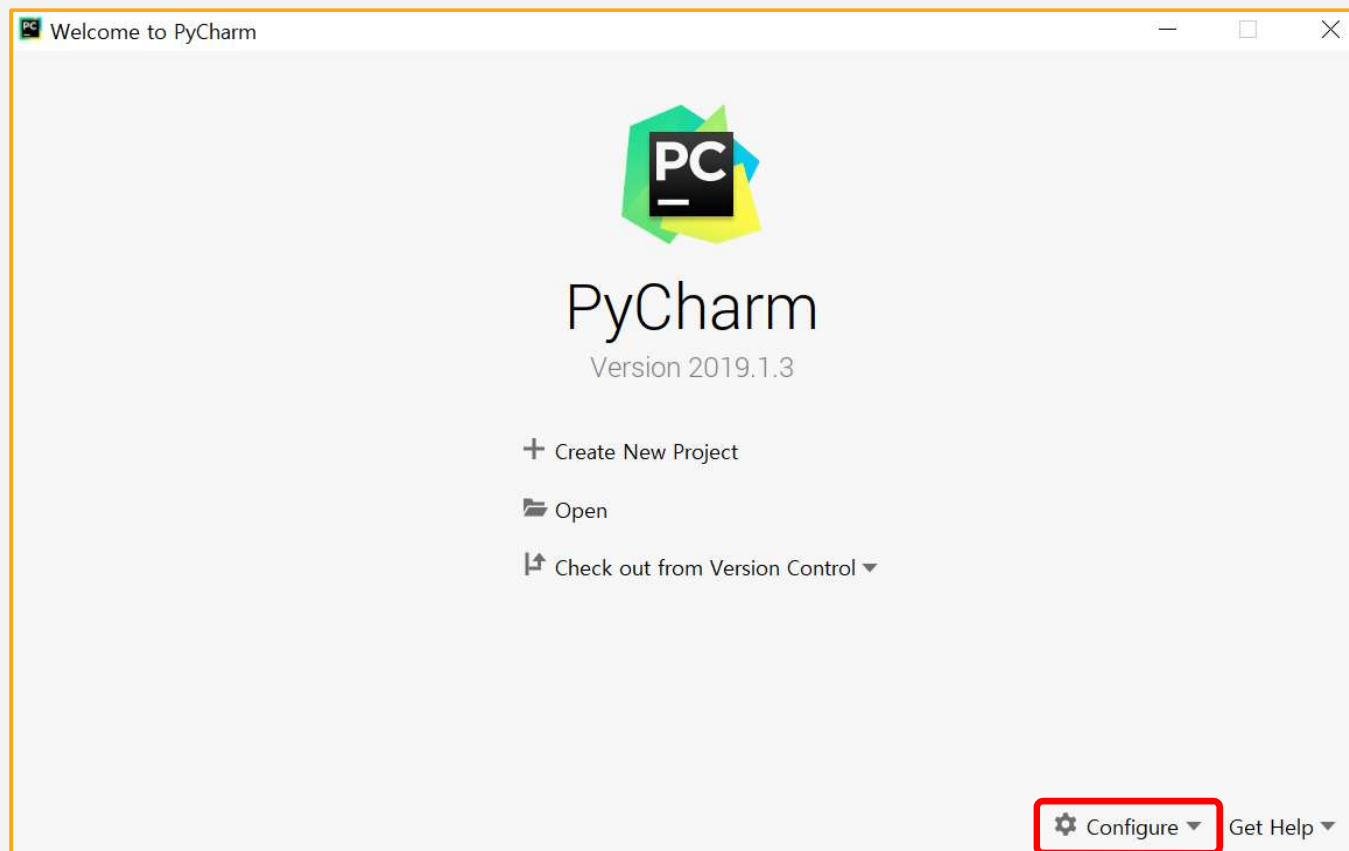


PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCharm 연동

➤ Python 인터프리터 설정

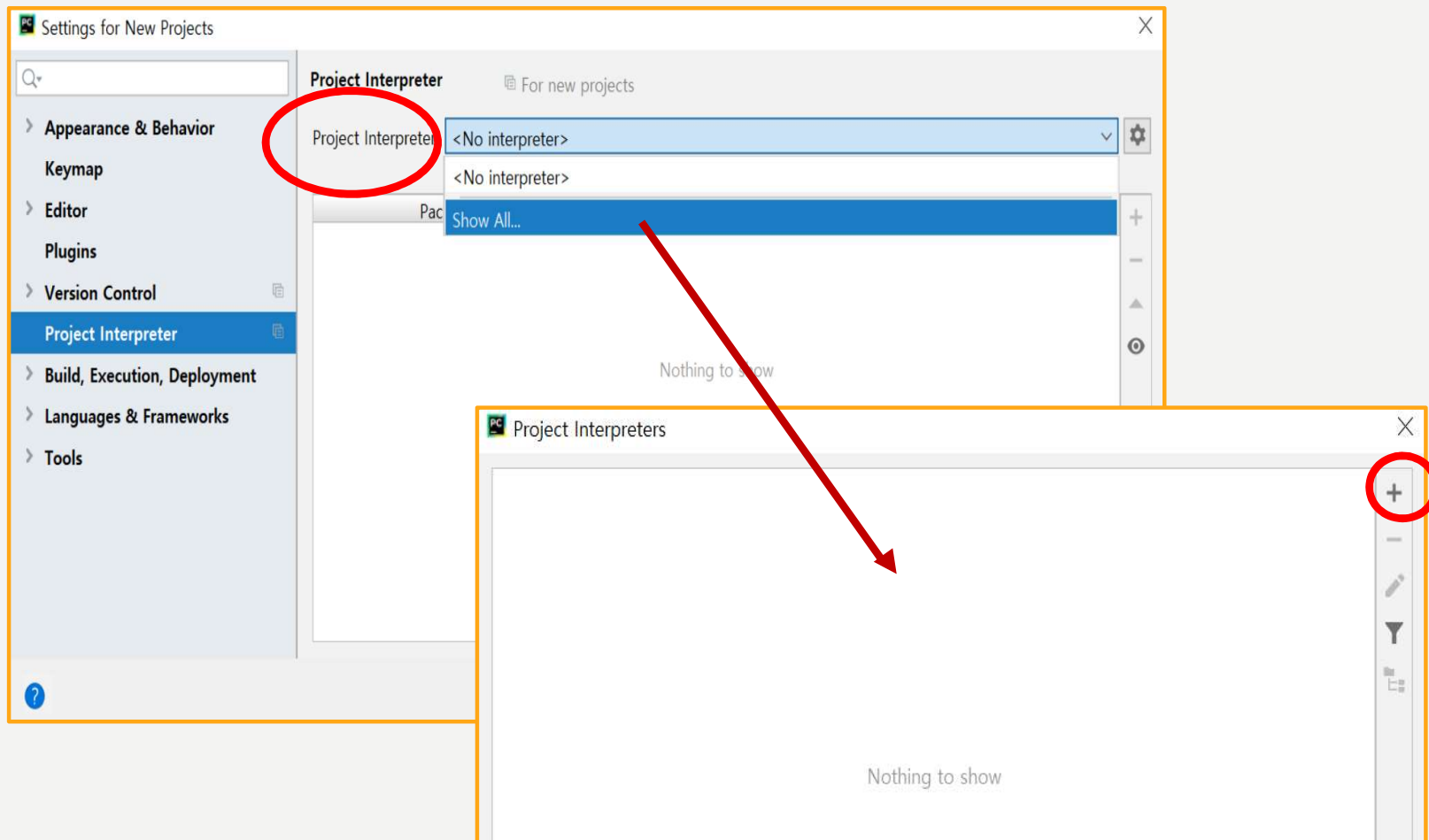
- Configure > Settings



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCharm 연동

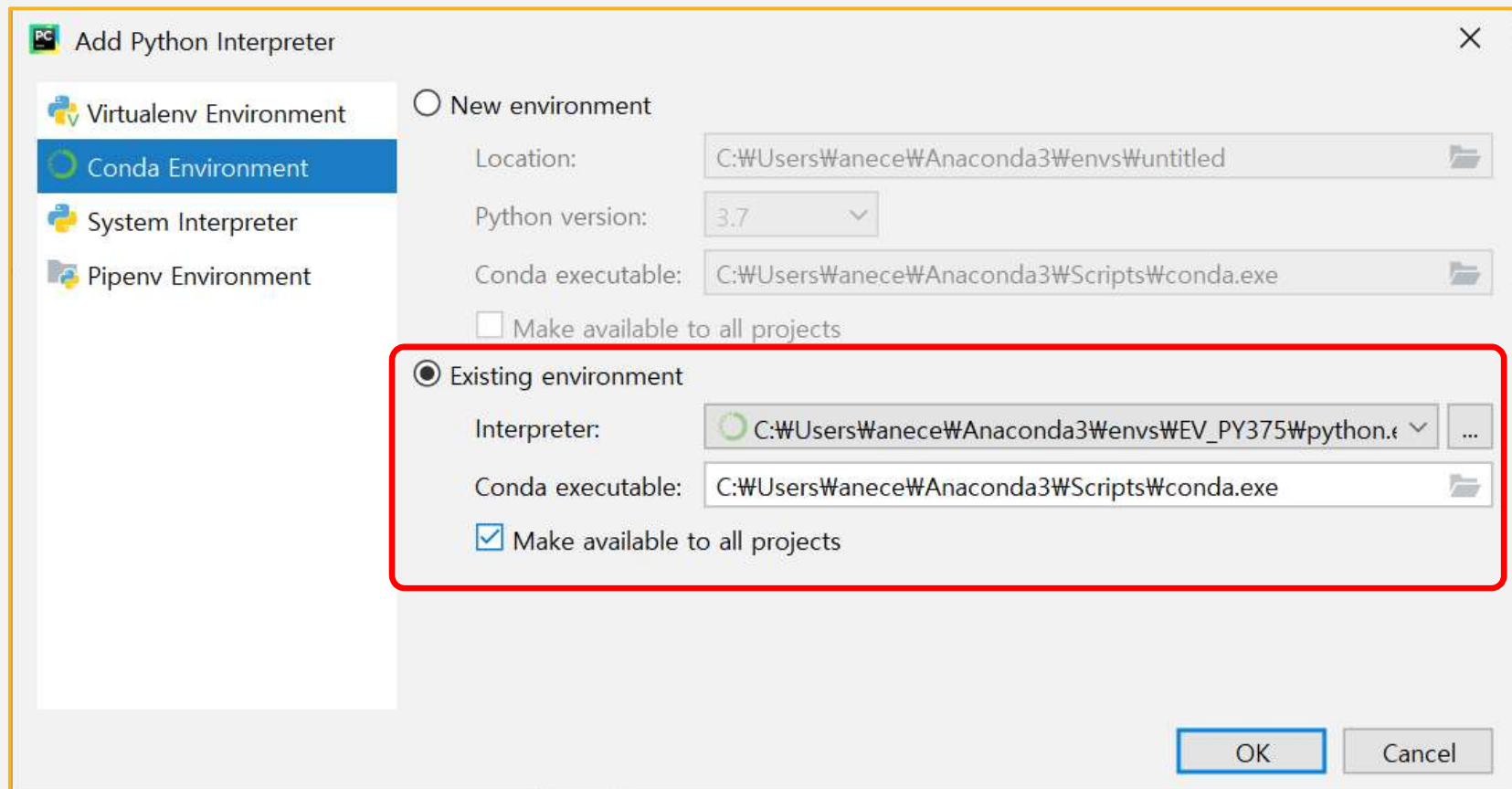
➤ Python 인터프리터 설정



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCharm 연동

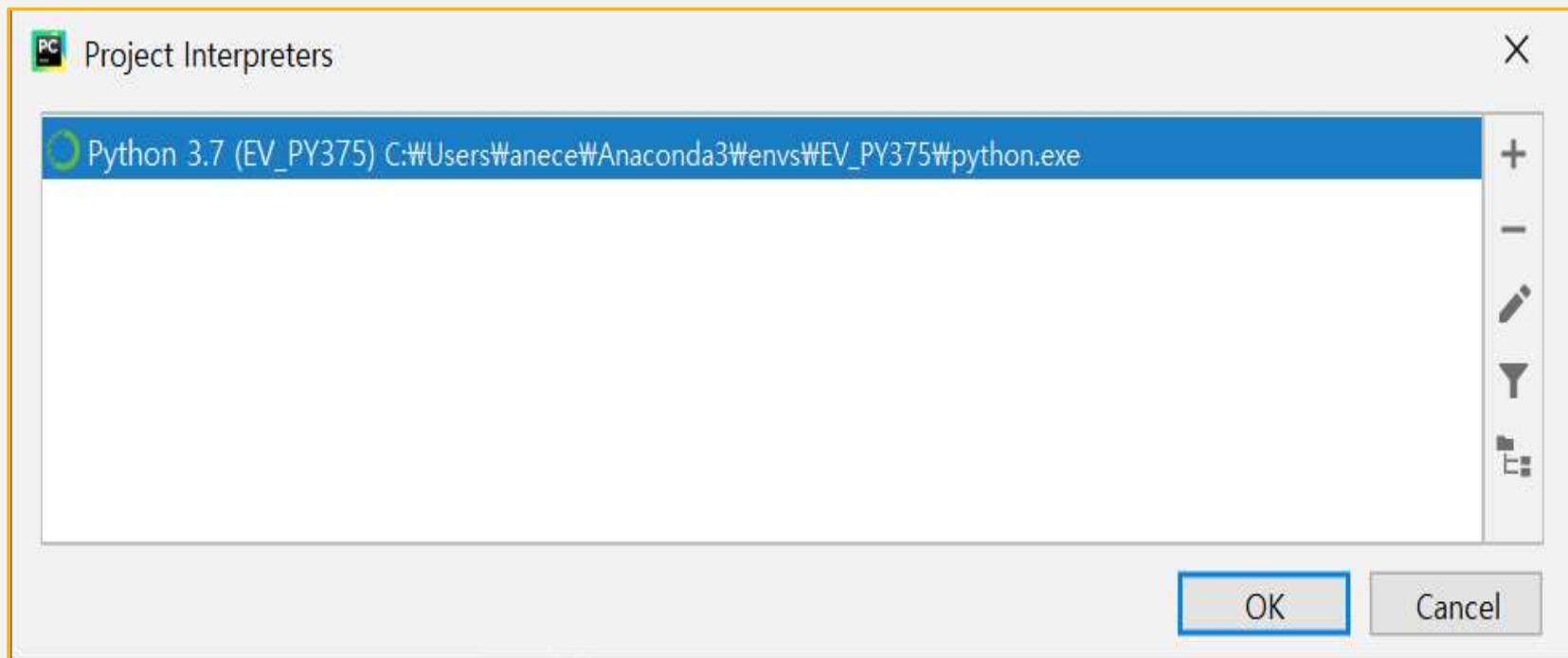
➤ Python 인터프리터 설정



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCharm 연동

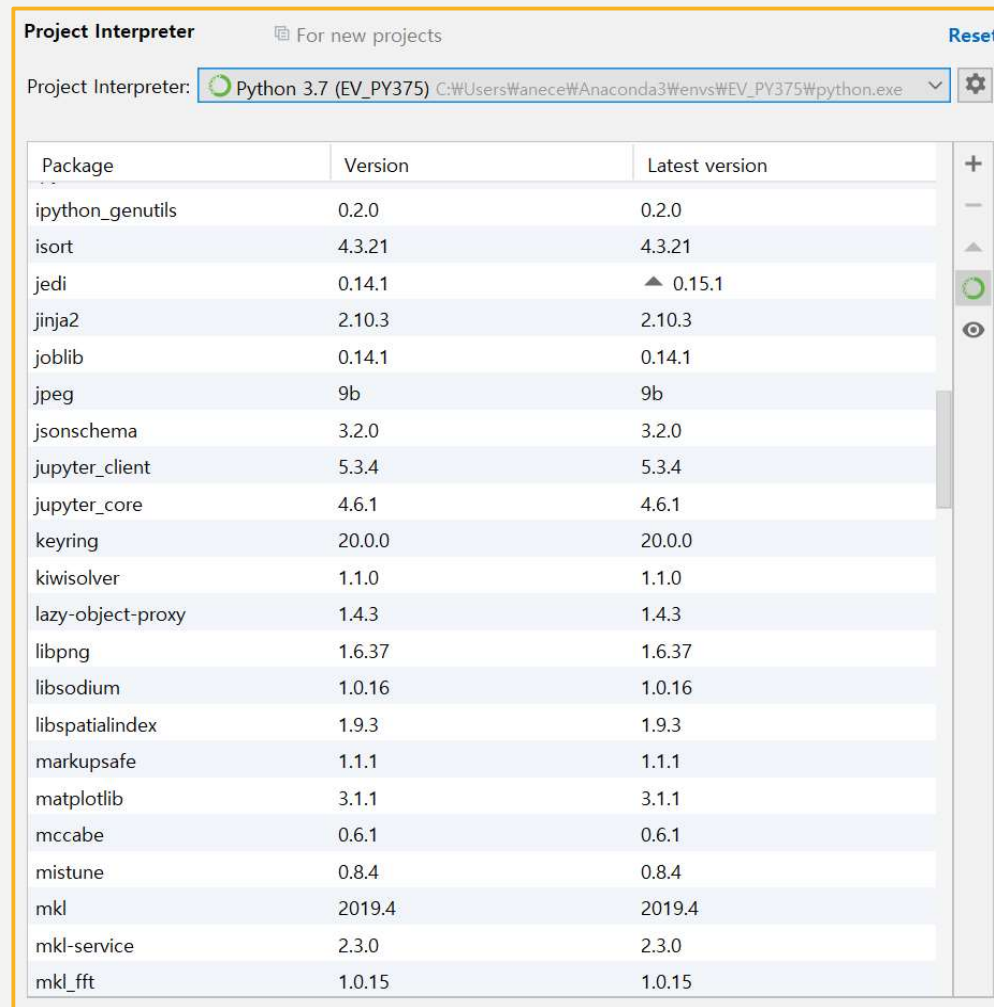
➤ python 인터프리터 설정



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCharm 연동

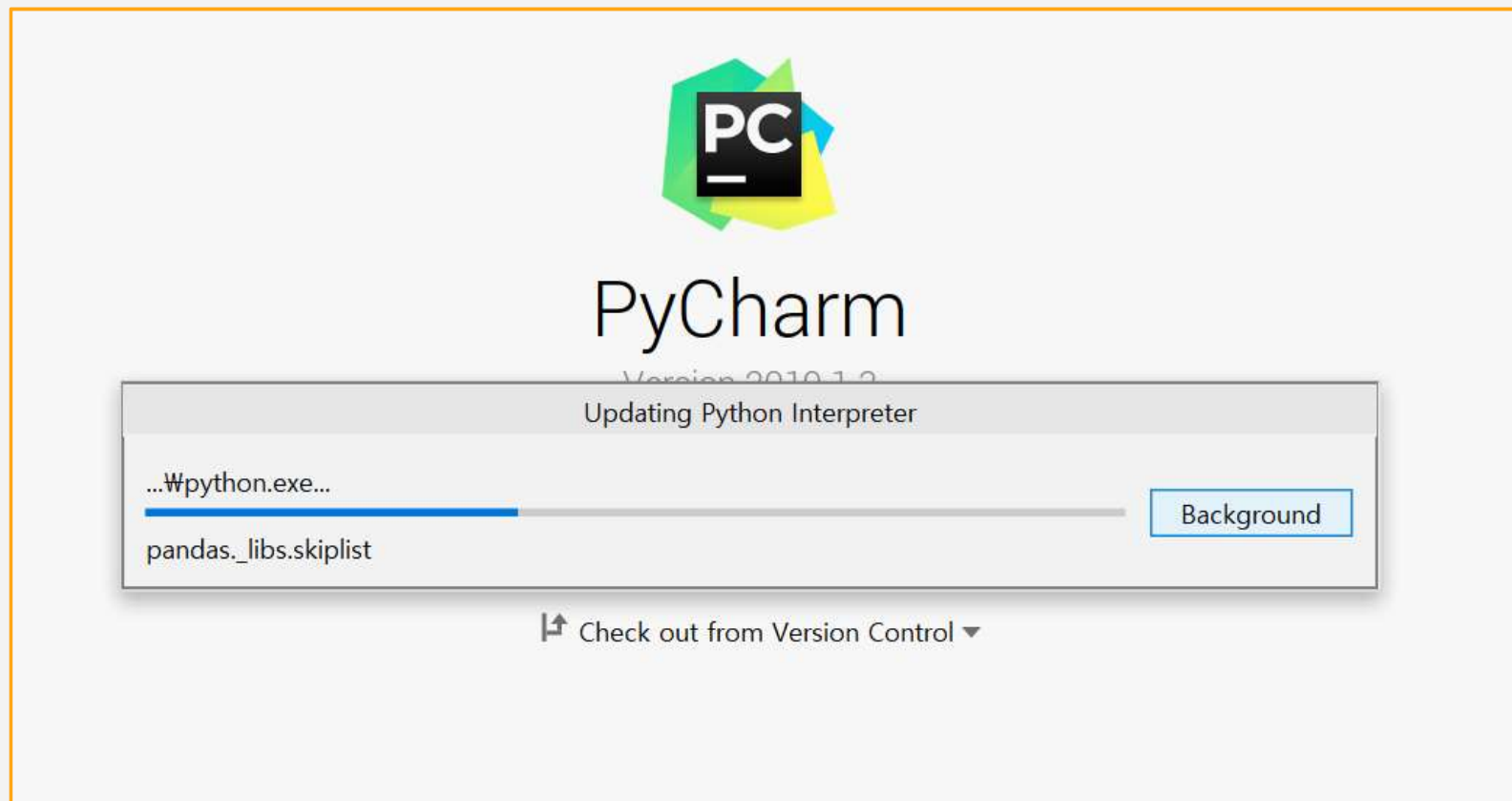
➤ Python 인터프리터 설정



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCharm 연동

➤ Python 인터프리터 설정

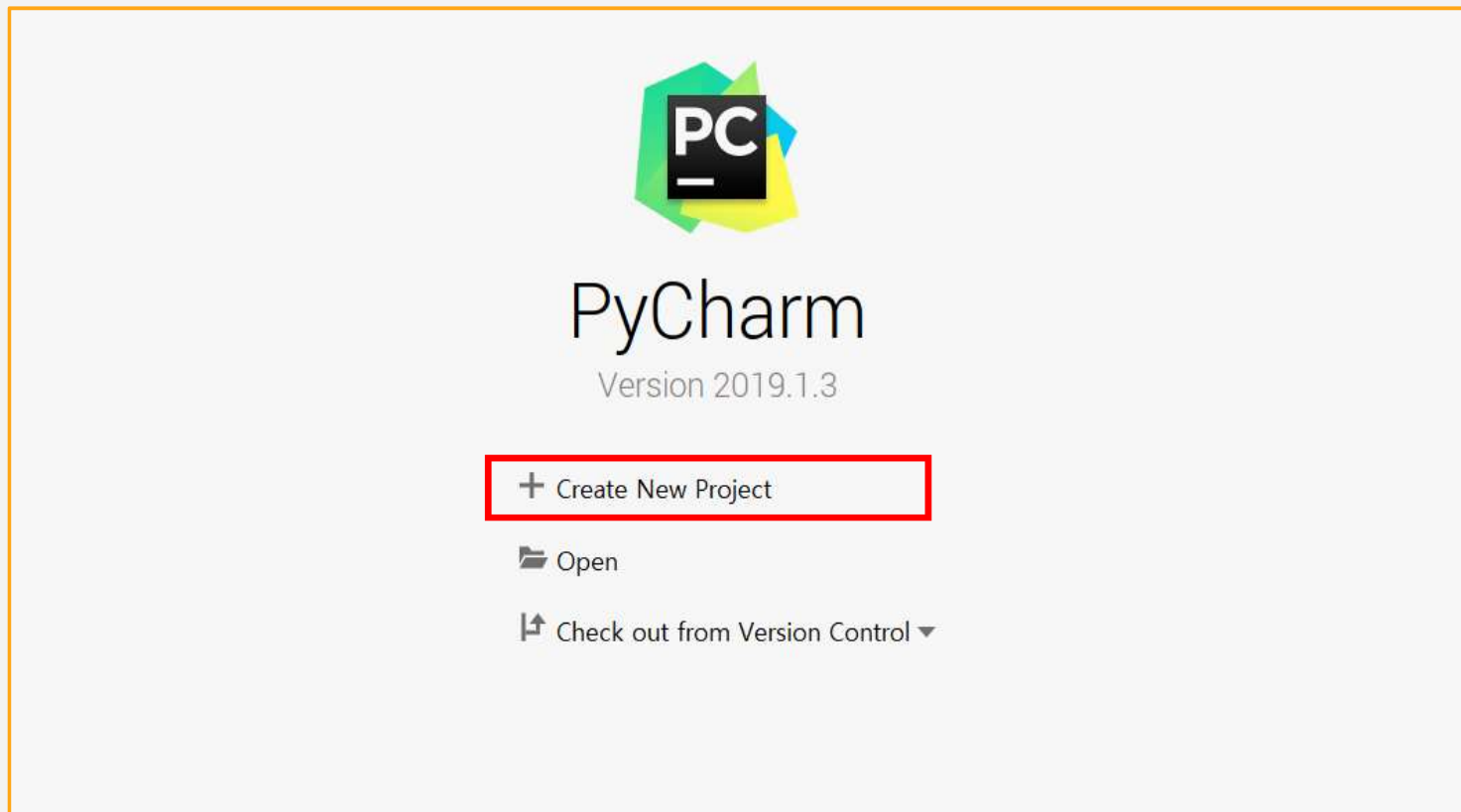


PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCharm 연동

➤ 프로젝트 생성

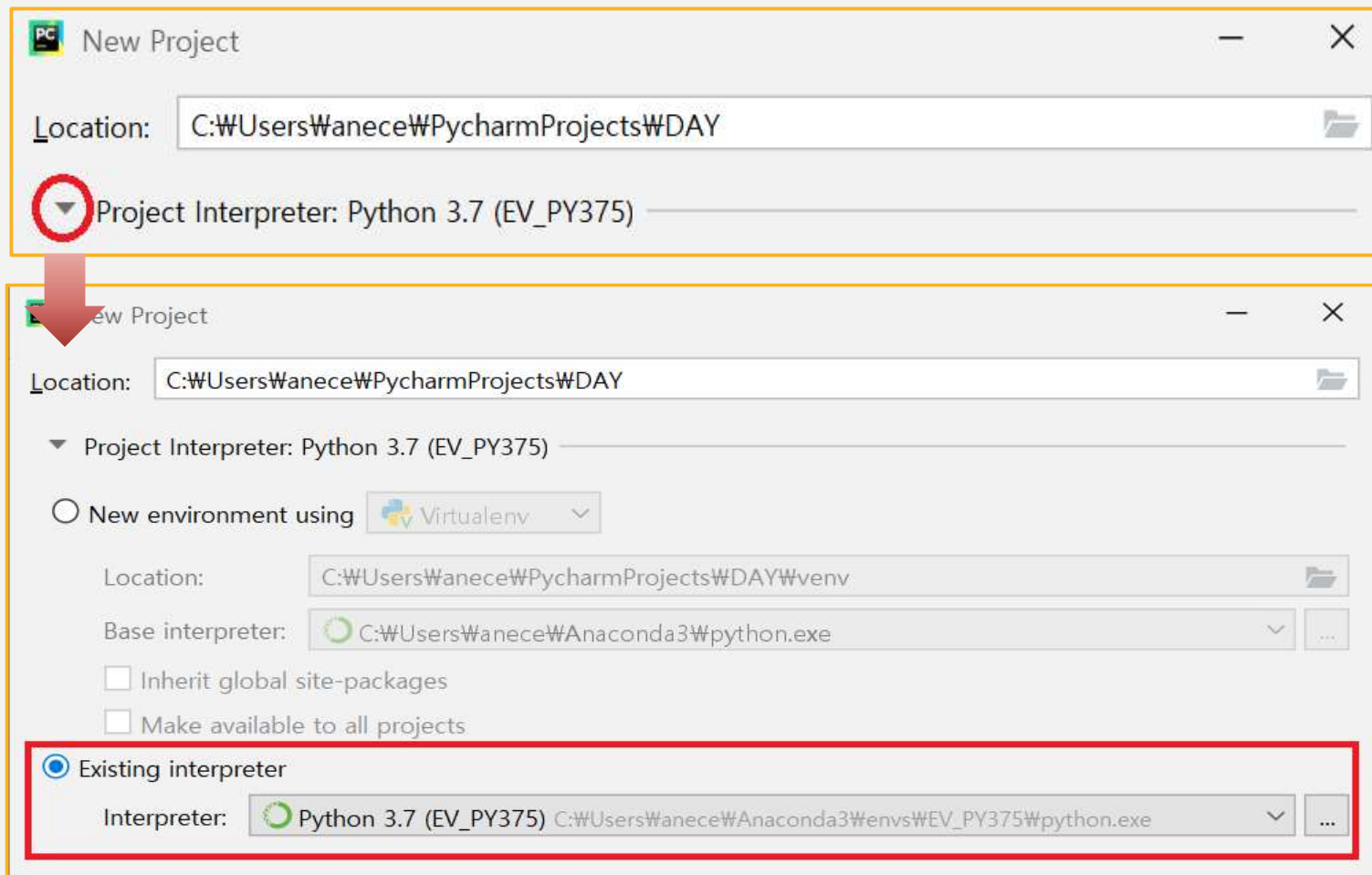
- PyCharm 종료 후 다시 실행 => Create New Project



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCharm 연동

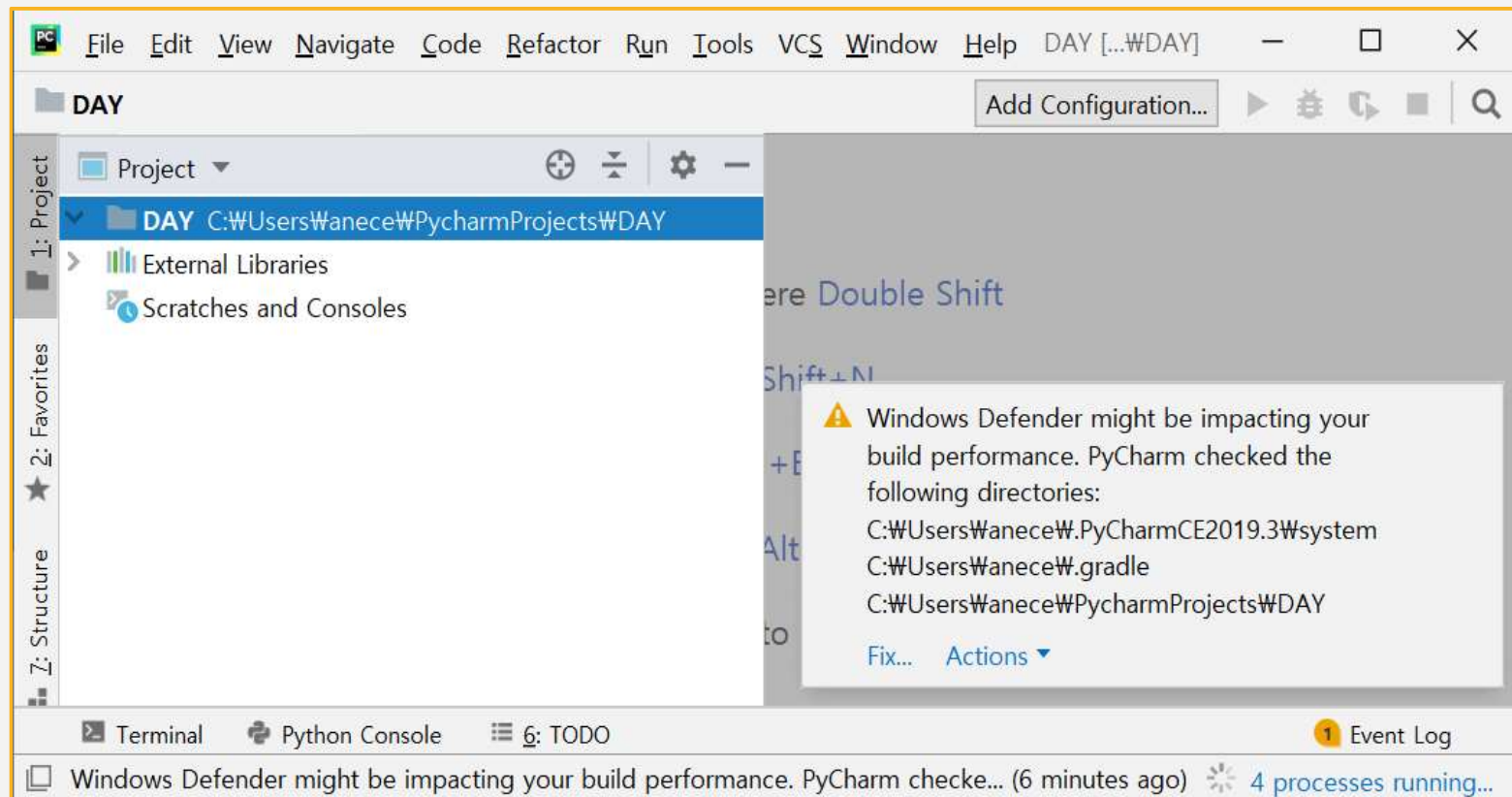
➤ 프로젝트 생성



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCharm 연동

➤ 프로젝트 생성



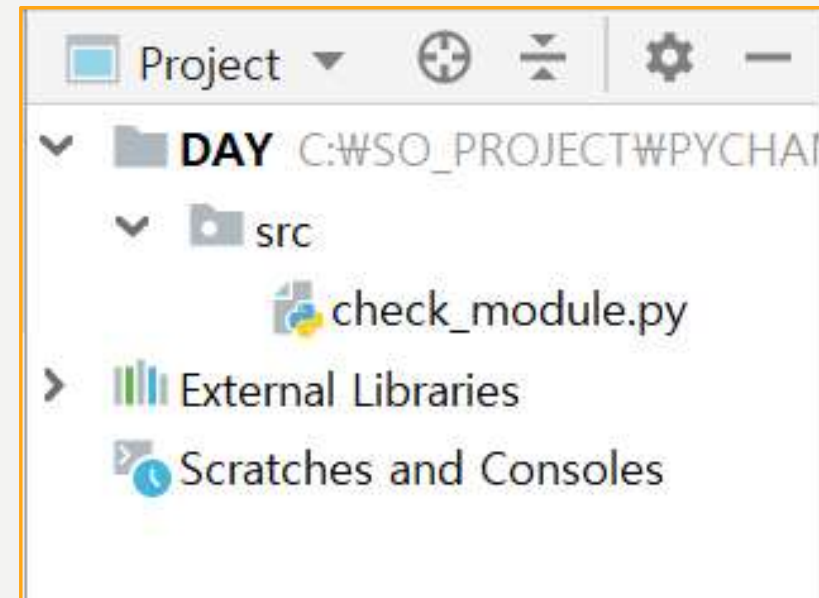
Windows Defender 혹은 그외 Anti Virus 백신에서 프로젝트 디렉토리를 제외 처리 해야 성능이 좋아진다 ➔ **Fix...**

PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCham 연동

➤ 프로젝트 생성

- 소스 코드 폴더 생성 → src
- 파일 생성 → check_module.py



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCham 연동

➤ 프로젝트 생성



The screenshot shows the PyCharm IDE interface. On the left, the 'Project' tool window displays the project structure: a folder named 'DAY' at 'C:\Users\Wanece\Py', containing a sub-folder 'src' with a file 'check_module.py'. Below this are 'External Libraries' and 'Scratches and Consoles'. The main editor area shows the content of 'check_module.py' with line numbers 1 through 14. The code imports several Python libraries and prints their versions.

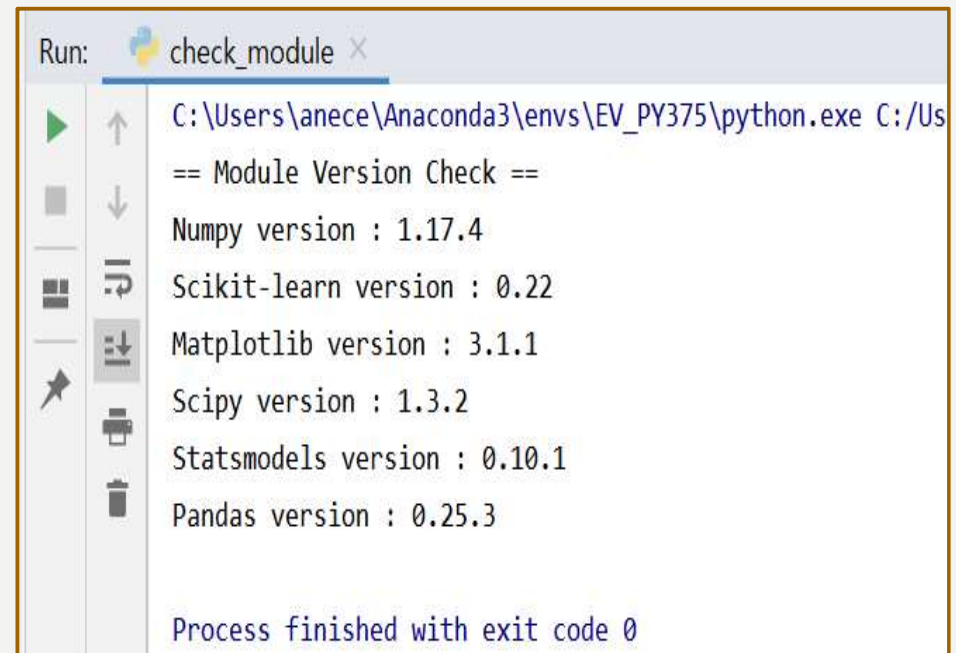
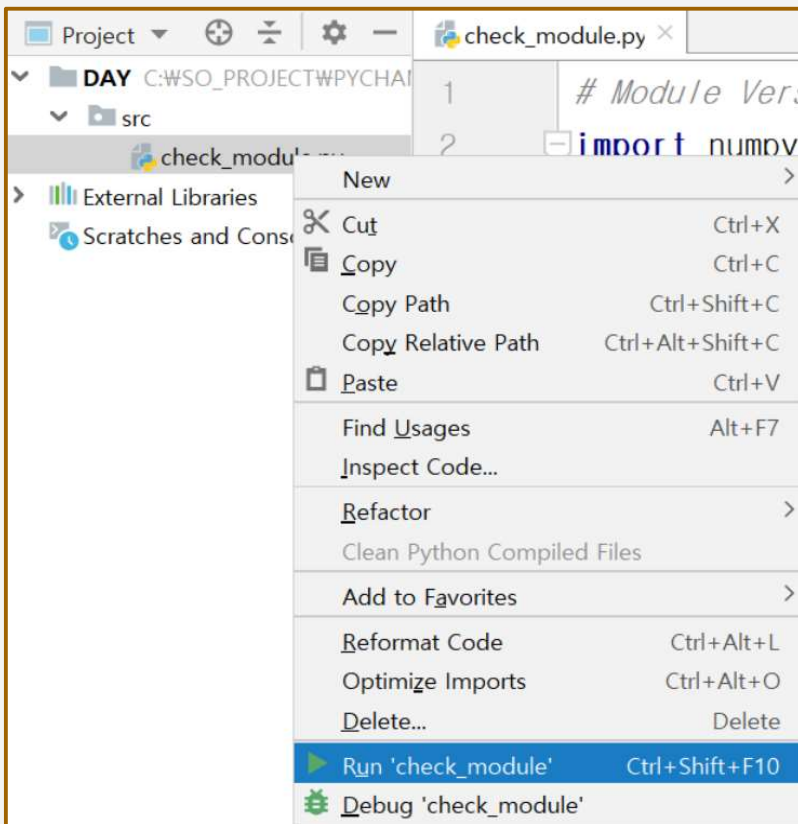
```
1 import numpy
2 import sklearn
3 import matplotlib as mp
4 import scipy as sp
5 import statsmodels as sm
6 import pandas as pd
7
8 print("== Module Version Check ==")
9 print("Numpy version : {}".format(numpy.__version__))
10 print("Scikit-learn version : {}".format(sklearn.__version__))
11 print("Matplotlib version : {}".format(mp.__version__))
12 print("Scipy version : {}".format(sp.__version__))
13 print("Statsmodels version : {}".format(sm.__version__))
14 print("Pandas version : {}".format(pd.__version__))
```


PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) & PyCham 연동

➤ 프로젝트 생성

- 파일 선택 > 오른쪽 클릭 > Run '파일명' 클릭



PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) 포함 IDE

➤ Spyder

- Anaconda 배포본에 포함된 IDE(Integrated Development Environment)
- Python 과학 프로그래밍 위한 오픈 소스 크로스 플랫폼 통합 개발 환경
- 에디터와 표준 Python 콘솔을 확장한 IPython 콘솔 지원
- 프로젝트, 디버깅, 변수 보기 등 다양한 편의기능 제공
- 편집 중 스크립트 전체 수행 또는 일부 영역 선택 수행 가능

PYTHON ML&DL 환경구축

◆ 아나콘다(Anaconda) 포함 IDE

➤ Jupyter Notebook

- 웹 브라우저에서 Python 코드 작성 및 실행해주는 IDE
- Anaconda 메뉴중 Jupyter Notebook 선택 또는 Anaconda Navigator에서 Jupyter Notebook 메뉴 선택

```
Jupyter Notebook
[ I 09:52:25.263 NotebookApp] JupyterLab application directory is D:\Anaconda3\share\jupyter\lab
[ I 09:52:25.265 NotebookApp] Serving notebooks from local directory: D:\EXAM_JUPYTER
[ I 09:52:25.266 NotebookApp] The Jupyter Notebook is running at:
[ I 09:52:25.266 NotebookApp] http://localhost:8888/?token=875ecbdceb8760ddf4a14758e0947e9fa50287b2ceef96e7
[ I 09:52:25.266 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

To access the notebook, open this file in a browser:
file:///C:/Users/PNJ/AppData/Roaming/jupyter/runtime/nbserver-22356-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=875ecbdceb8760ddf4a14758e0947e9fa50287b2ceef96e7
[W 09:53:07.758 NotebookApp] delete /ex_test.ipynb
[W 09:53:08.000 NotebookApp] delete /Untitled.ipynb
[W 09:53:08.172 NotebookApp] delete /test.ipynb
```

[Sever] Python 실행

[Client] Python 코드 입력

