

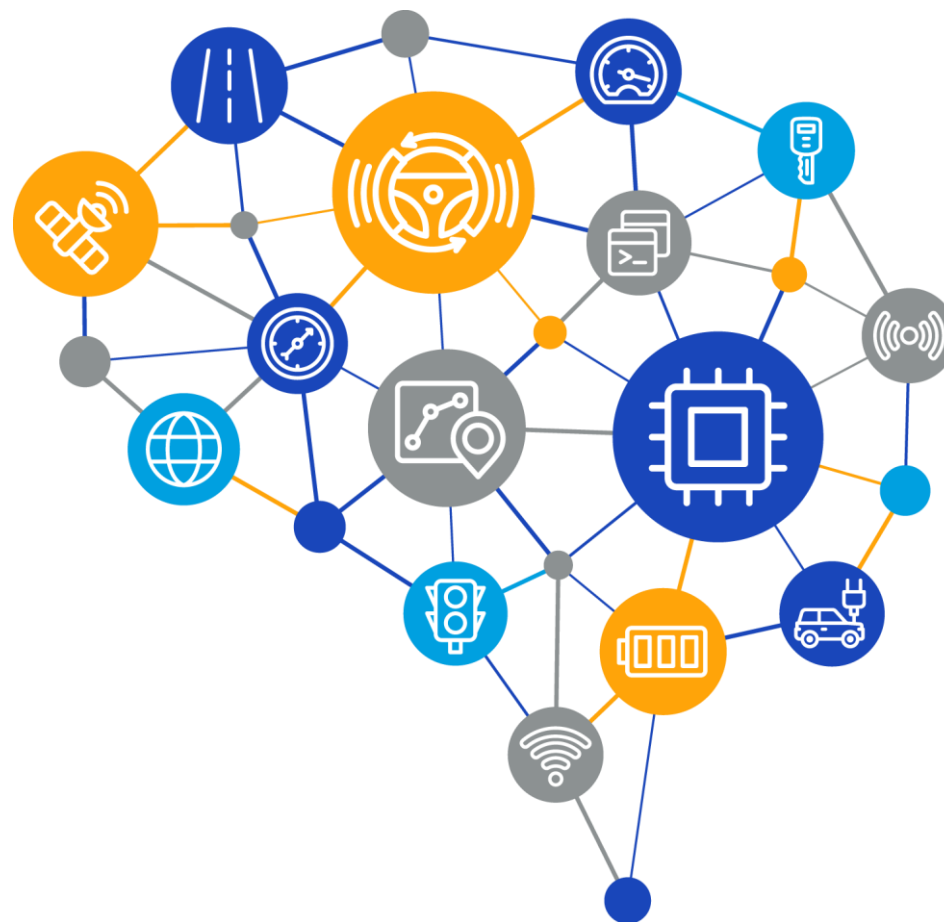
# AI+ 인공지능 인재양성을 위한 비교과프로그램 개발 및 운영

## 파이썬 위한 다양한 개발환경

2022.07.12(화) ~ 2022.07.15(금)

2022.07

강환수 교수



# AI Experts Who Lead The Future

## CONTENTS

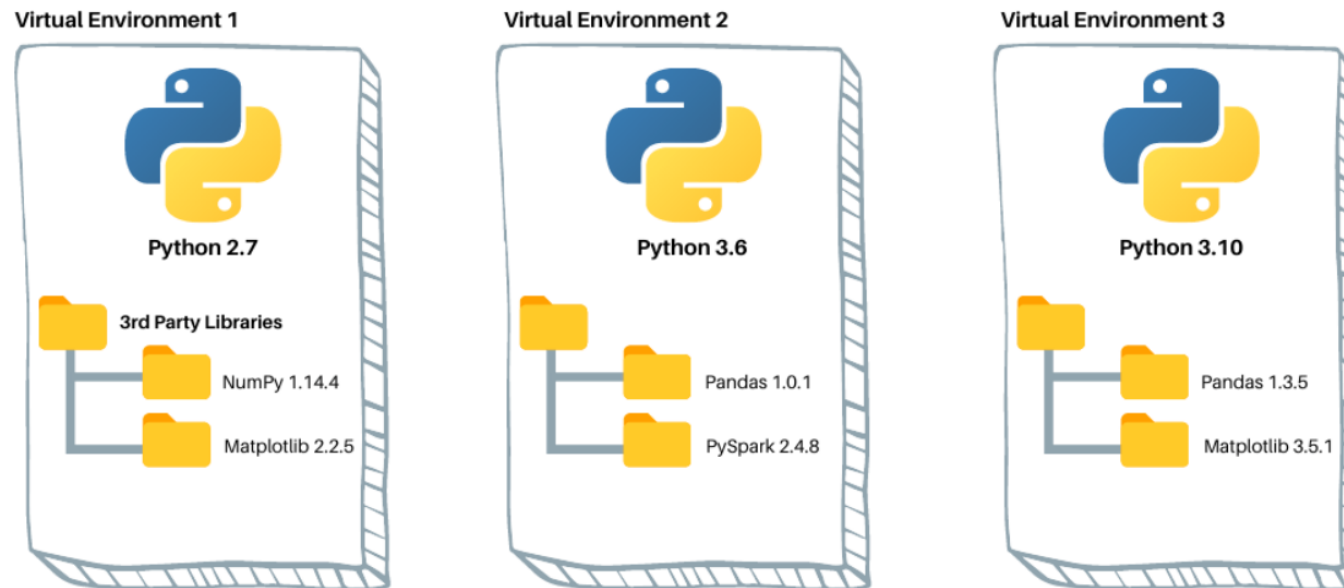
- 01 | 깃과 깃허브 이해
- 02 | 깃 설치
- 03 | 깃 실행과 기본 설정
- 04 | 깃허브 클론(복사)
- 05 | 오픈소스 소프트웨어 개요
- 06 | 오픈소스 소프트웨어 라이선스

AI Experts  
Who Lead  
The Future

# 01

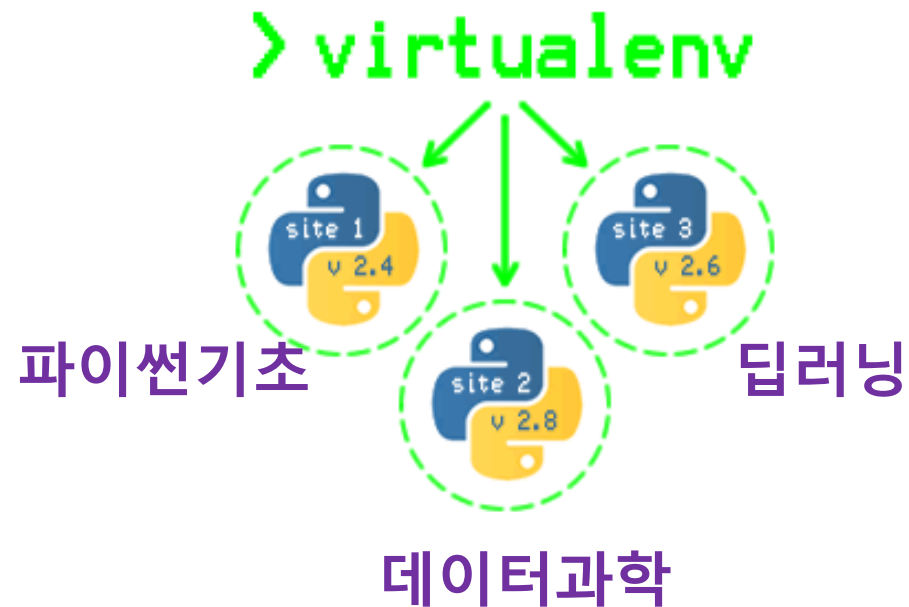
## 파이썬 가상환경 이해와 준비

- 가상환경(Virtual Environments)
  - 격리된 별도의 라이브러리 설치 폴더
    - 원하는 Python 환경을 구축하기 위해 필요한 모듈만 담아 놓는 바구니
  - 파이썬의 다른 버전이 필요한 경우 발생
    - python 3.10
      - Switch case
    - python 3.6



dataquest.io

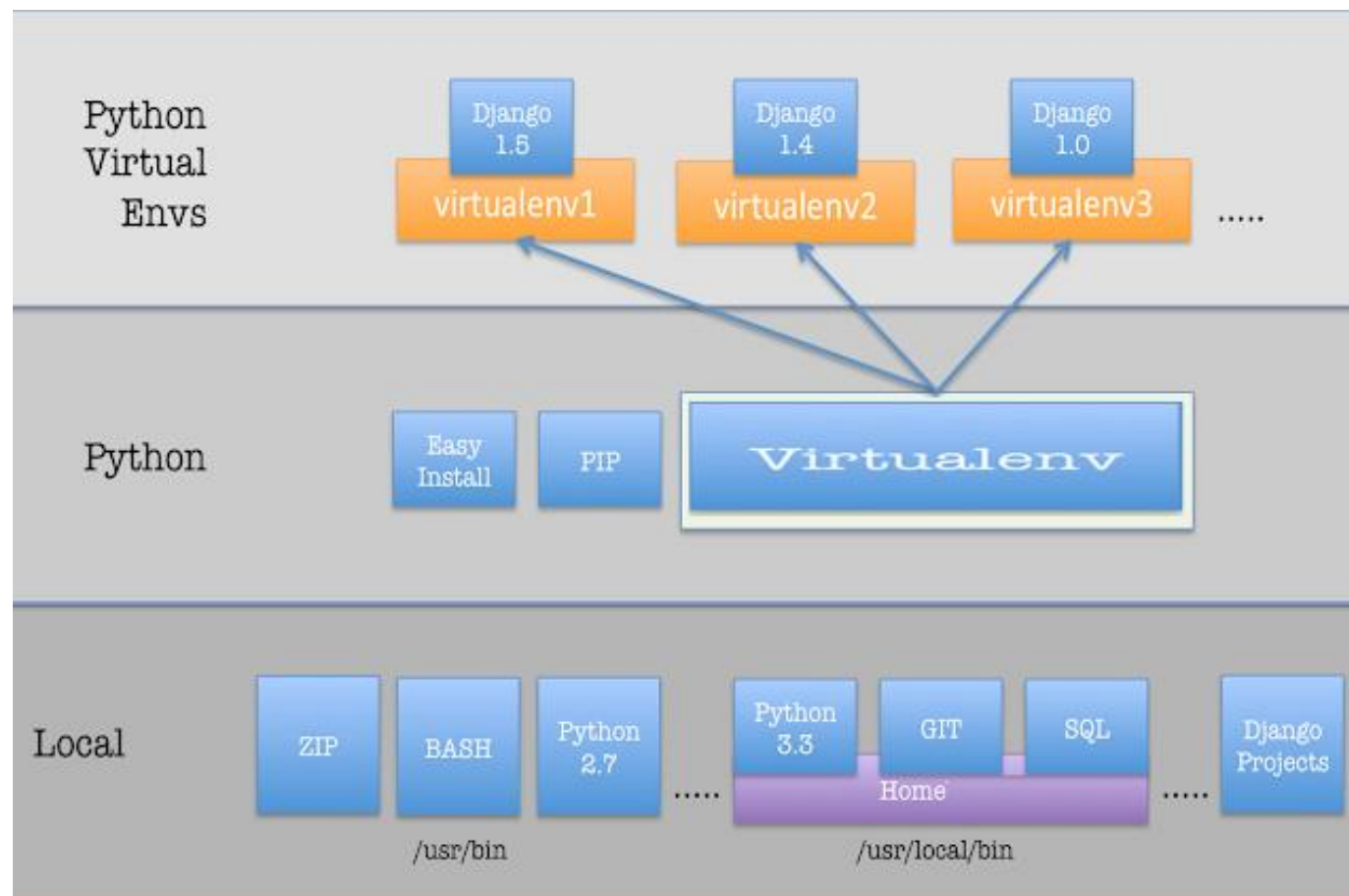
- 가상환경의 필요성
  - 하나의 환경에는 여러 버전의 설치를 할 수 없음
    - 여러 환경을 만들고 각각의 환경에 여러 버전 설치 가능
  - python 프로그램을 실행하기 위한 다양한 환경을 마련
- 가상환경의 주요 목적
  - Python 프로젝트를 위한 격리된 개발환경을 만드는 것
    - 다른 모든 프로젝트의 종속성에 관계없이 각 프로젝트마다 고유한 종속성을 가질 수 있음을 의미



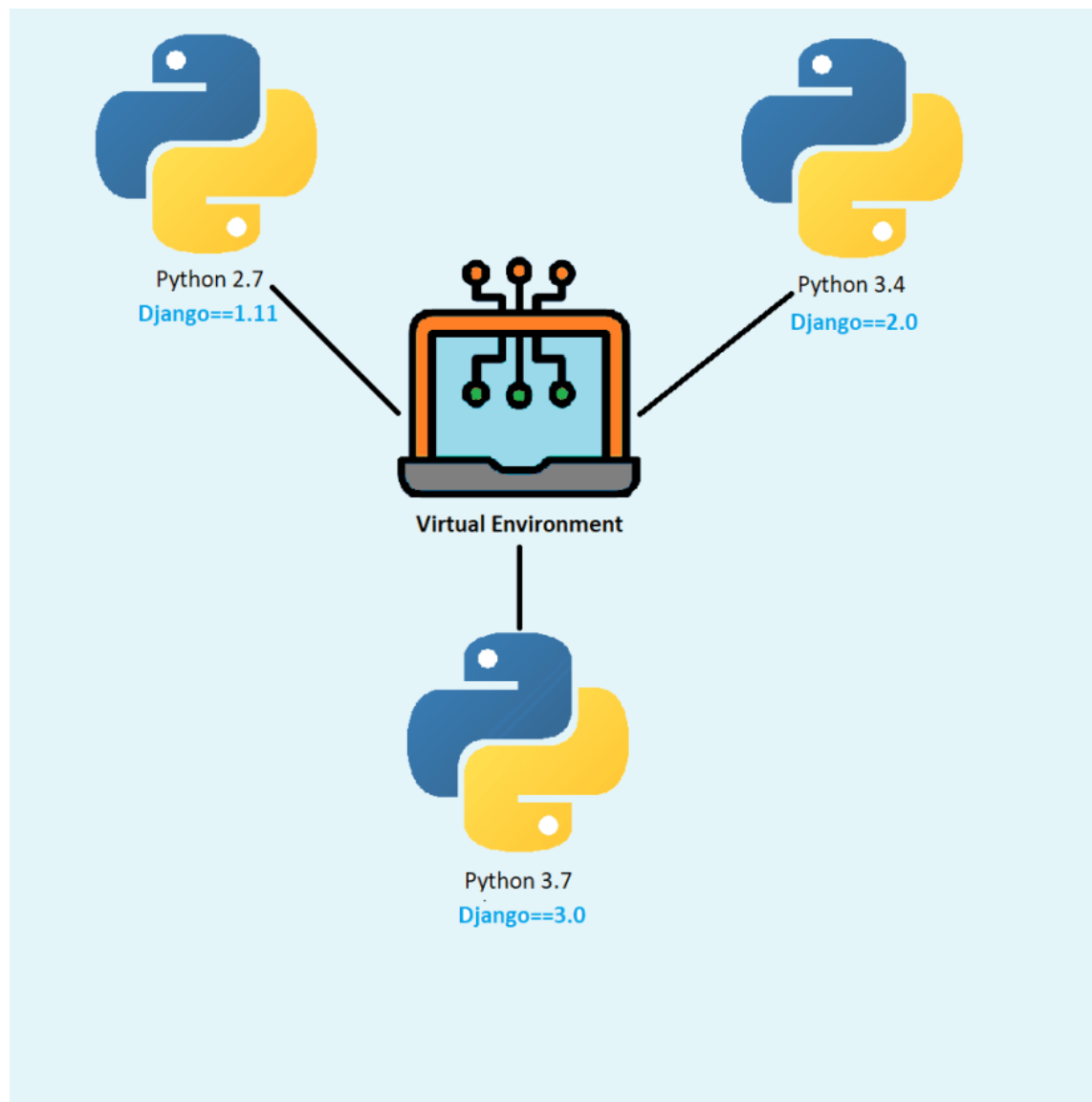
# 가상환경 생성 방법

가상환경 Python language

- Python에서 가상환경을 만드는 방법
  - 크게 2가지로 virtualenv와 conda를 사용
    - venv, pipenv 등 매우 다양



- 다른 패키지, 동일 패키지의 다른 버전 등 설치



## 자신의 PC에 여러 개의 개발환경 설치된 경우

가상환경 Python language

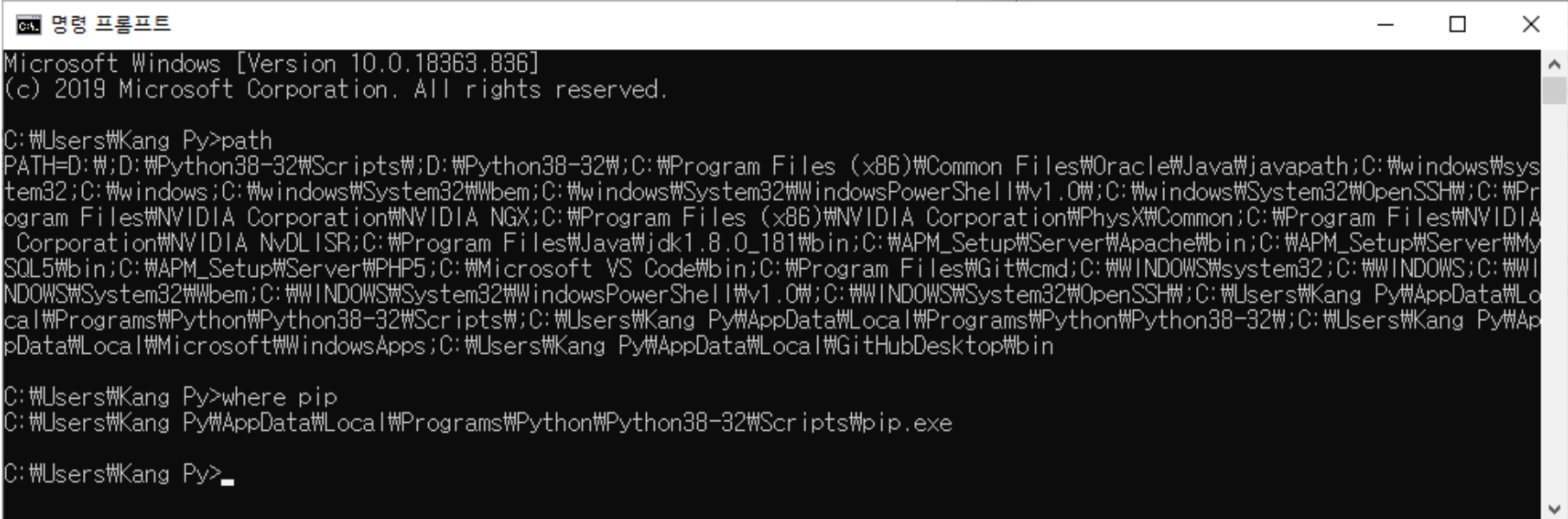
- 어느 개발환경의 PIP를 실행하는지 확인 필요



## 명령 프롬프트 path 점검

가상환경 Python language

- 명령어 pip 사용 위치 확인
  - 처음 나오는 pip 명령어가 실행됨
    - 여러 개가 있더라도 맨 위에 있는 것이 실행됨



```

C:\Users\Kang Py>path
PATH=D:\Python38-32\Scripts;D:\Python38-32;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\windows\system32;C:\windows;C:\windows\System32\WindowsPowerShell\v1.0;C:\windows\System32\OpenSSH;C:\Program Files\NVIDIA Corporation\NVIDIA NGX;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR;C:\Program Files\Java\jdk1.8.0_181\bin;C:\APM_Setup\Server\Apache\bin;C:\APM_Setup\Server\MySQL5\bin;C:\APM_Setup\Server\PHP5;C:\Microsoft VS Code\bin;C:\Program Files\Git\cmd;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0;C:\WINDOWS\System32\OpenSSH;C:\Users\Kang Py\AppData\Local\Programs\Python\Python38-32\Scripts;C:\Users\Kang Py\AppData\Local\Programs\Python\Python38-32;C:\Users\Kang Py\AppData\Local\Microsoft\WindowsApps;C:\Users\Kang Py\AppData\Local\GitHubDesktop\bin

C:\Users\Kang Py>where pip
C:\Users\Kang Py\AppData\Local\Programs\Python\Python38-32\Scripts\pip.exe

C:\Users\Kang Py>_
  
```

- 항상 pip를 사용하는 경우, 어느 개발환경의 pip가 실행되는지 확인
  - 해당되는 가상 개발환경에 pip의 패키지가 설치되는 것임

# pip 명령어 검사

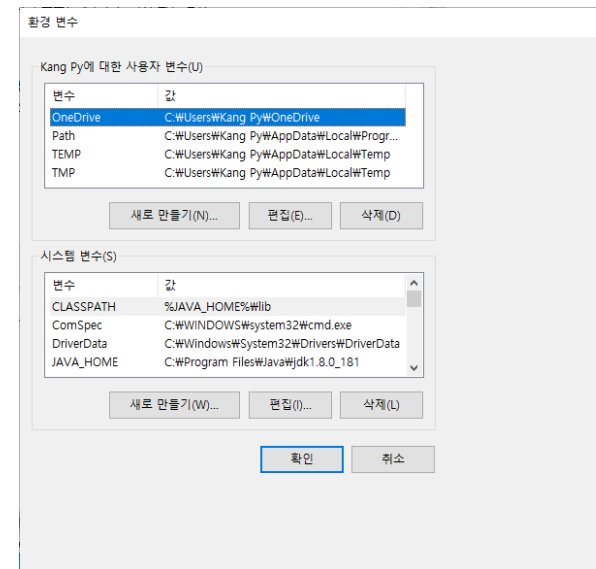
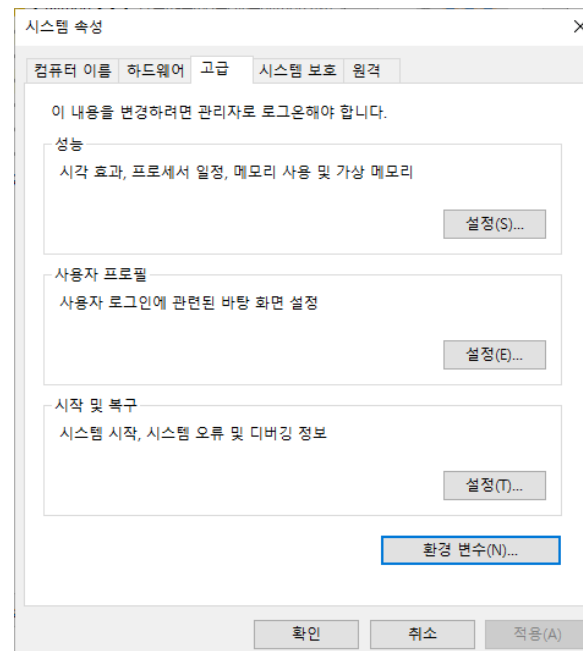
가상환경 Python language

- Pip 명령어가 있는 폴더
  - 표준 파이썬인 경우
    - D:\Python\Python38-32\Scripts\pip.exe
  - 아나콘다인 경우
    - D:\Python\Anaconda3\Scripts\pip.exe
- 표준 파이썬의 pip를 사용하려면
  - Where pip 명령어의 결과에서 다음이 먼저 나오도록 조정
    - D:\Python\Python38-32\Scripts\pip.exe
- [다음은 가능한가 다시 점검 필요]
- 먼저 표준 파이썬의 pip를 사용해 가상환경 생성
  - 표준 파이썬이 path에 먼저 설정되도록
  - 필요 시 path 직접 수정하는 방법
    - set path=D:\Python38-32\Scripts\;D:\Python38-32\;%path%
  - 다만 이 방법은 현재의 명령 프롬프트창에서만 path가 변경

# 필요 시 path 변경

가상환경 Python language

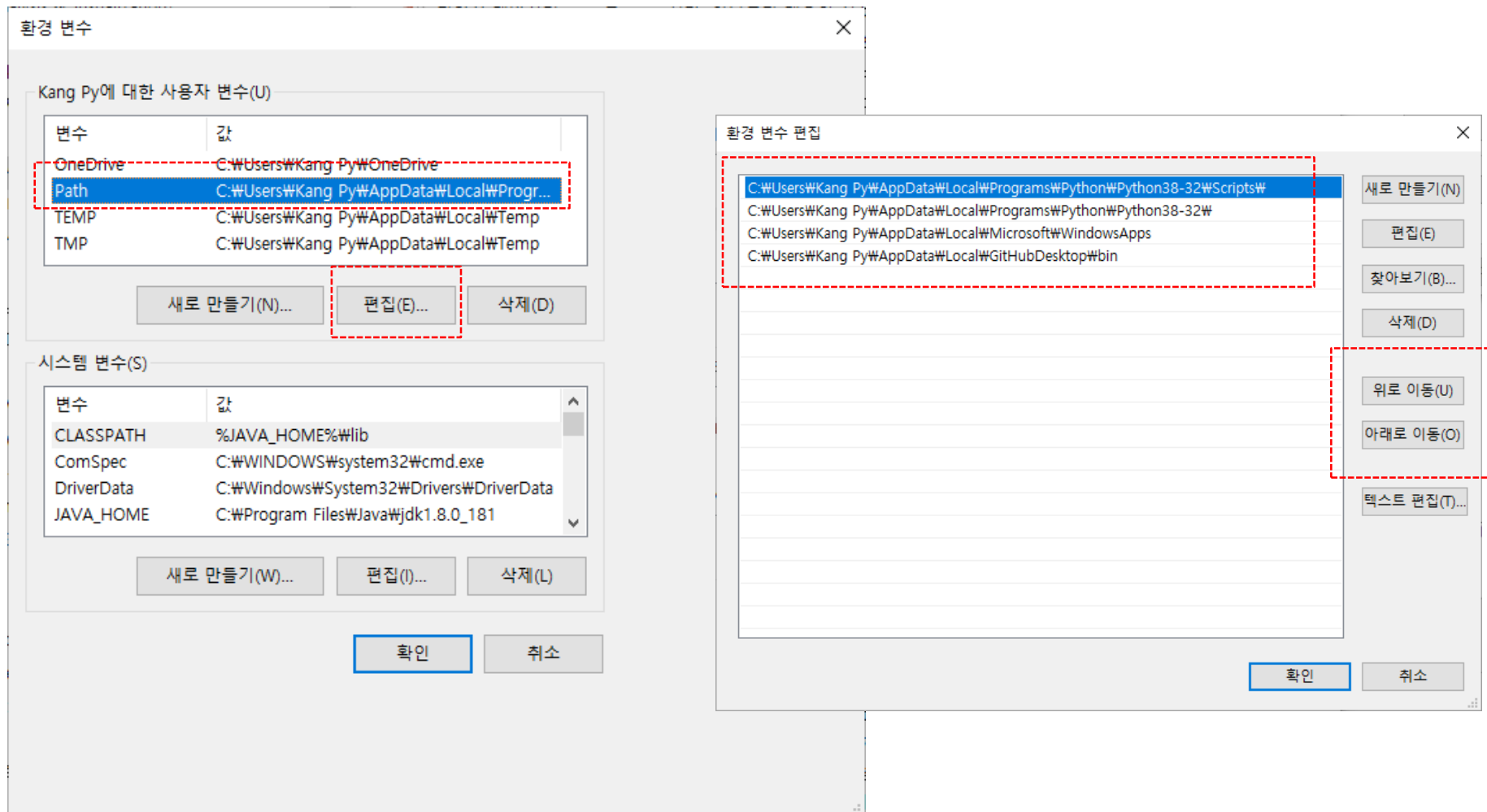
- 검색창에서 검색
  - '고급 시스템 설정 보기'
  - Path 경로 등록 순서의 중요성
    - 만약 동일한 이름의 파일이 각기 다른 폴더에 존재하고 또 해당 폴더들이 모두 Path에 등록되어 있는 경우
      - 먼저 등록된 폴더 안의 파일이 실행
    - 명령어 pip 사용 위치 확인 필요한 이유



# Path 확인: 사용자 변수

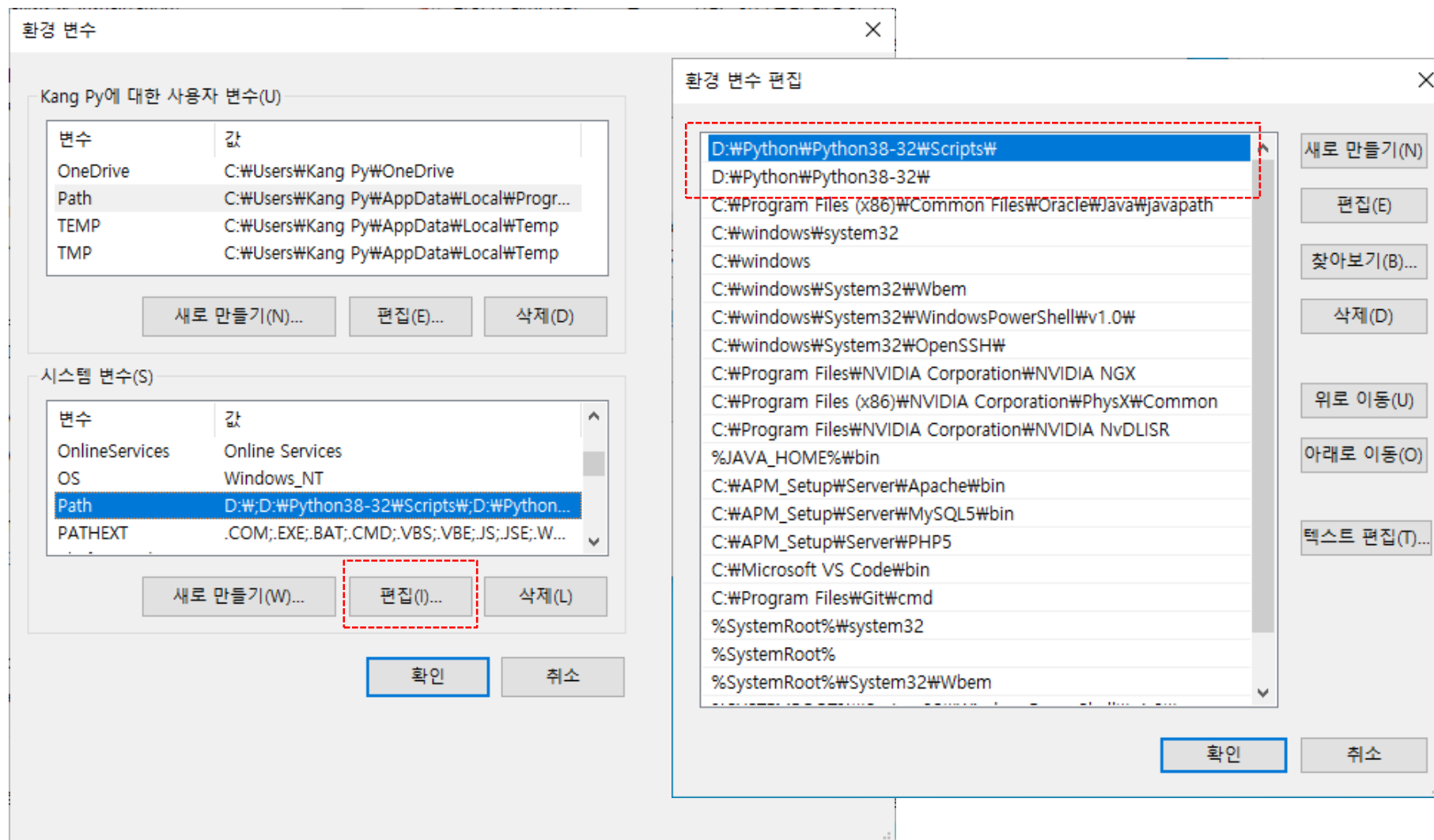
가상환경 Python language

- 사용자 변수
  - 필요하면 환경 변수 값을 이동



# Path 확인: 시스템 변수

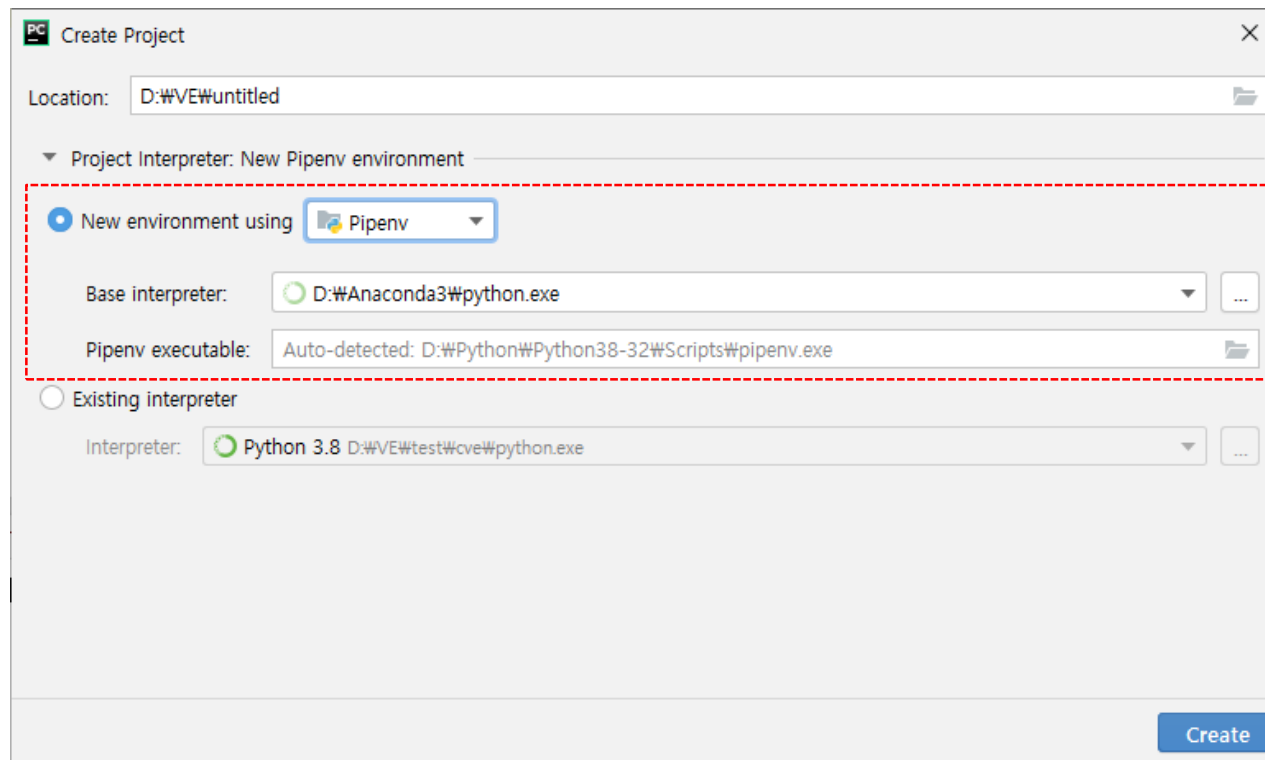
가상환경 Python language



# 파이참 가상환경 생성 도구 점검

가상환경 Python language

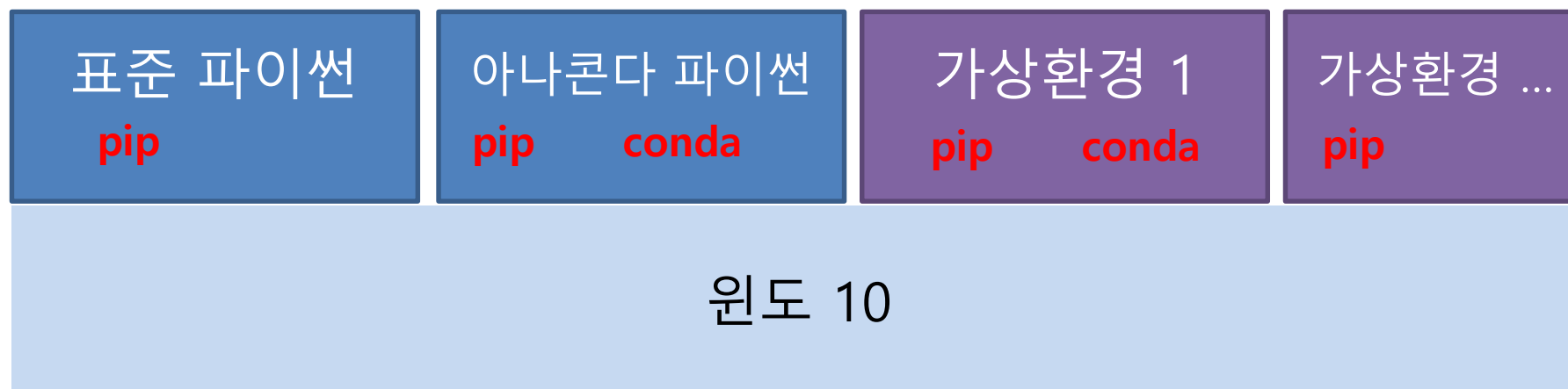
- Pipenv
- VirtualEnv



# 현재의 시스템

가상환경 Python language

- Python 인터프리터 기본 환경
  - 표준 python: pip
  - 아나콘다 배포판: conda pip



- 파이썬 환경에서 패키지 설치 명령어
  - pip(Package Installer for Python)
    - pip install 패키지명
    - pip list
  - 각각의 환경에서 pip를 실행하면 그 환경에 패키지가 설치
    - 서로 독립적
- 현재 시스템 확인

# 모듈 설치 명령 pip

가상환경 Python language

- pip 명령

pip 명령	설명
pip search {키워드}	키워드로 관련 패키지를 검색합니다.
pip install {패키지명}	지정한 패키지를 설치합니다.
pip uninstall {패키지명}	지정한 패키지를 삭제합니다.
pip install --upgrade {패키지}	지정한 패키지를 업데이트합니다.
pip show {패키지}	지정한 패키지의 버전을 출력합니다.
pip list	전체 패키지 목록을 출력합니다.



- 0. [준비]필요 가상환경 도구 설치
  - venv
    - 설치 불필요
  - conda
    - 아나콘다 설치
  - virtualenv
    - `pip install virtualenv`
  - pipenv
    - `pip install pipenv`
- 설치 전에 pip 업그레이드
  - 하지 않으면 오류 발생 있음
    - `> python -m pip install --upgrade pip`
  - 여러 상황에서 달리 발생
    - 설치해보고 안되면 upgrade하는 방법

## 가상환경 생성 4단계 절차

가상환경 Python language

- 1. 가상환경 생성
  - 도구에 따라 여러 방법
- 2. 생성된 가상환경을 활성화(들여가기)
  - activate, conda activate
  - Pipenv인 경우
    - pipenv shell
- 3. 활성화된 가상환경 내에서 필요 패키지를 설치, 개발을 진행
  - venv, virtualenv
    - pip install 모듈명
  - conda
    - conda install 모듈명
  - pipenv
    - pipenv install 모듈명
      - pip install 모듈명도 가능
- 4. 현재 활성화된 가상환경을 비활성화(빠져나오기)
  - deactivate, conda deactivate
  - Pipenv인 경우
    - exit

AI Experts  
Who Lead  
The Future

## 02

### 표준 파이썬 venv를 사용한 가상환경 생성

# 하나의 폴더에서 작업

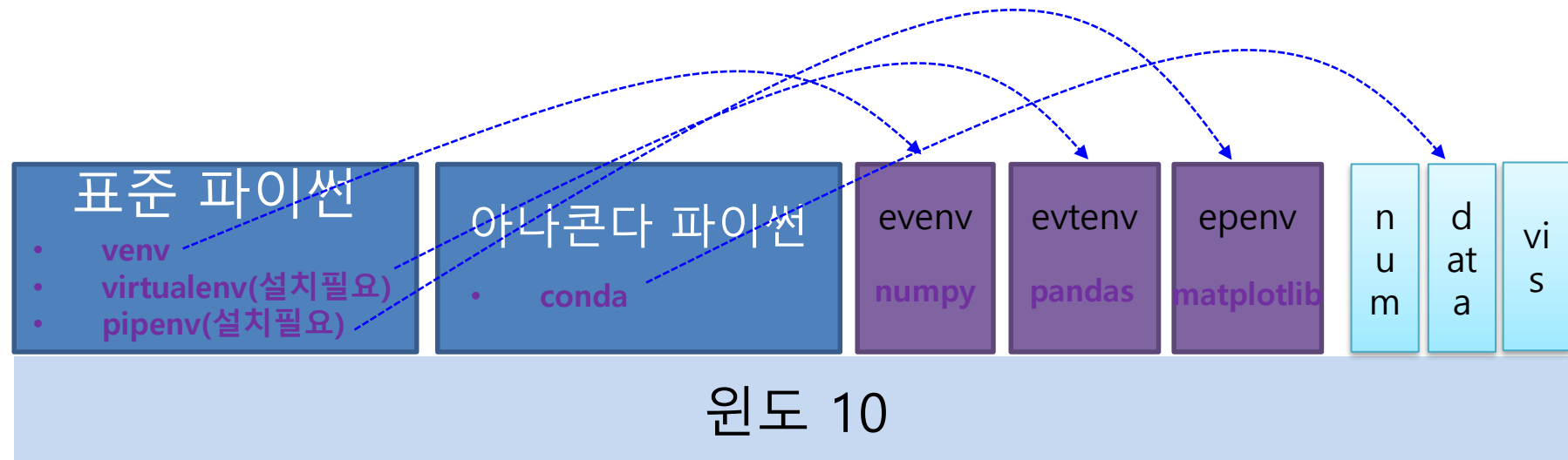
가상환경 Python language

- C 또는 D 드라이브에서
  - 폴더 VE 생성
- 가상환경 필요 모듈 설치
  - 먼저 외부에서 모듈 virtualenv 설치 필요
    - pip install virtualenv
    - pip install pipenv
  - 명령 pip 확인
    - 표준 파이썬에서 설치

```

명령 프롬프트
C:\Users\Kang Py>where pip
C:\Users\Kang Py\AppData\Local\Programs\Python\Python38-32\Scripts\pip.exe
C:\Users\Kang Py>_

```



- 지정 폴더 하부에 가상환경 생성
- 상위 폴더 ve 하부에 각각 생성
  - venv 로 생성, 자동 생성
    - Num

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

D:\WE>where pip
D:\Anaconda3\Scripts\pip.exe
D:\Python3102\Scripts\pip.exe
C:\Program Files\Python38\Scripts\pip.exe
C:\Program Files\Python310\Scripts\pip.exe

D:\WE>where python
D:\Anaconda3\python.exe
D:\Python3102\python.exe
C:\Program Files\Python38\python.exe
C:\Users\WPCW\AppData\Local\Microsoft\WindowsApps\python.exe

D:\WE>_
```

- 가상환경 venv 만들기
  - 적당한 폴더 ve에서 시작
    - D:
    - cd ve
  - 가상환경 venv 생성
    - python -m venv num
  - 가상환경 활성화
    - num\scripts\activate
  - 가상환경에 numpy 설치
    - pip install numpy
  - 설치 확인
    - pip show numpy
  - 가상환경에서 나오기
    - deactivate

```
C:\Windows\System32\cmd.exe

(num) D:\WEWnum>where python
D:\WEWnum\Scripts\python.exe
D:\Anaconda3\python.exe
D:\Python3102\python.exe
C:\Program Files\Python38\python.exe
C:\Users\WPCW\AppData\Local\Microsoft\WindowsApps\python.exe

(num) D:\WEWnum>pip list
Package      Version
-----
pip          22.0.4
setuptools  58.1.0
WARNING: You are using pip version 22.0.4; however, version 22.1.2 is available.
You should consider upgrading via the 'D:\WEWnum\Scripts\python.exe -m pip install --upgrade pip' command.

(num) D:\WEWnum>pip install numpy
Collecting numpy
  Downloading numpy-1.23.0-cp39-cp39-win_amd64.whl (14.7 MB)
    ----- 14.7/14.7 MB 72.6 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-1.23.0
WARNING: You are using pip version 22.0.4; however, version 22.1.2 is available.
You should consider upgrading via the 'D:\WEWnum\Scripts\python.exe -m pip install --upgrade pip' command.

(num) D:\WEWnum>python -m pip install --upgrade pip
Requirement already satisfied: pip in d:\wewnum\lib\site-packages (22.0.4)
Collecting pip
  Using cached pip-22.1.2-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.0.4
    Uninstalling pip-22.0.4:
      Successfully uninstalled pip-22.0.4
Successfully installed pip-22.1.2

(num) D:\WEWnum>
```

```
C:\Windows\System32\cmd.exe

(num) D:\WVEWnum>pip list
Package      Version
-----
numpy        1.23.0
pip          22.1.2
setuptools   58.1.0

(num) D:\WVEWnum>python
Python 3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> import site
>>> site.getsitepackages()
['D:\WVEWnum', 'D:\WVEWnum\lib\site-packages']
>>> exit()

(num) D:\WVEWnum>
```



# 가상환경 확인 방법

가상환경 Python language

## 가상환경의 인터프리터에서 다음 확인

- 가상환경 폴더 확인
- 외부모듈 설치폴더 확인

```
>>> import site
>>> site.getsitepackages()
```

```
>>> import sys
>>> sys.prefix
'D:\\ve\\evenv'
>>> from distutils.sysconfig import get_python_lib
>>> print(get_python_lib())
D:\\ve\\evenv\\Lib\\site-packages
```

```
(evenv) D:\\VE>path
PATH=D:\\VE\\evenv\\Scripts;D:\\;D:\\Python38-32\\Scripts\\;D:\\Python38-32\\;C:\\Program Files (x86)\\Common Files\\Oracle\\Java\\javapath;C:\\windows\\system32;C:\\windows\\System32\\Wbem;C:\\windows\\System32\\WindowsPowerShell\\v1.0\\;C:\\windows\\System32\\OpenSSH\\;C:\\Program Files\\NVIDIA Corporation\\NVIDIA NGX;C:\\Program Files (x86)\\NVIDIA Corporation\\PhysX\\Common;C:\\Program Files\\NVIDIA Corporation\\NVIDIA NvDLISR;C:\\Program Files\\Java\\jdk1.8.0_181\\bin;C:\\APM_Setup\\Server\\Apache\\bin;C:\\APM_Setup\\Server\\MySQL5\\bin;C:\\APM_Setup\\Server\\PHP5;C:\\Microsoft VS Code\\bin;C:\\Program Files\\Git\\cmd;C:\\WINDOWS\\system32;C:\\WINDOWS;C:\\WINDOWS\\System32\\Wbem;C:\\WINDOWS\\System32\\WindowsPowerShell\\v1.0\\;C:\\WINDOWS\\System32\\OpenSSH\\;C:\\Users\\Kang Py\\AppData\\Local\\Programs\\Python\\Python38-32\\Scripts\\;C:\\Users\\Kang Py\\AppData\\Local\\Programs\\Python\\Python38-32\\;C:\\Users\\Kang Py\\AppData\\Local\\Microsoft\\WindowsApps;C:\\Users\\Kang Py\\AppData\\Local\\GitHubDesktop\\bin

(evenv) D:\\VE>python
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> sys.path
['', 'C:\\Users\\Kang Py\\AppData\\Local\\Programs\\Python\\Python38-32\\python38.zip', 'C:\\Users\\Kang Py\\AppData\\Local\\Programs\\Python\\Python38-32\\DLLs', 'C:\\Users\\Kang Py\\AppData\\Local\\Programs\\Python\\Python38-32\\lib', 'C:\\Users\\Kang Py\\AppData\\Local\\Programs\\Python\\Python38-32', 'D:\\VE\\evenv', 'D:\\VE\\evenv\\Lib\\site-packages']
>>> sys.prefix
'D:\\VE\\evenv'
>>> from distutils.sysconfig import get_python_lib
>>> get_python_lib()
'D:\\VE\\evenv\\Lib\\site-packages'
>>> import site
>>> site.getsitepackages()
['D:\\VE\\evenv', 'D:\\VE\\evenv\\Lib\\site-packages']
```

AI Experts  
Who Lead  
The Future

# 03

## 아나콘다의 conda 개요

# 아나콘다 프롬프트 실행

가상환경 Python language

- 아나콘다가 제공하는 명령 프롬프트에서 실행되는 명령어

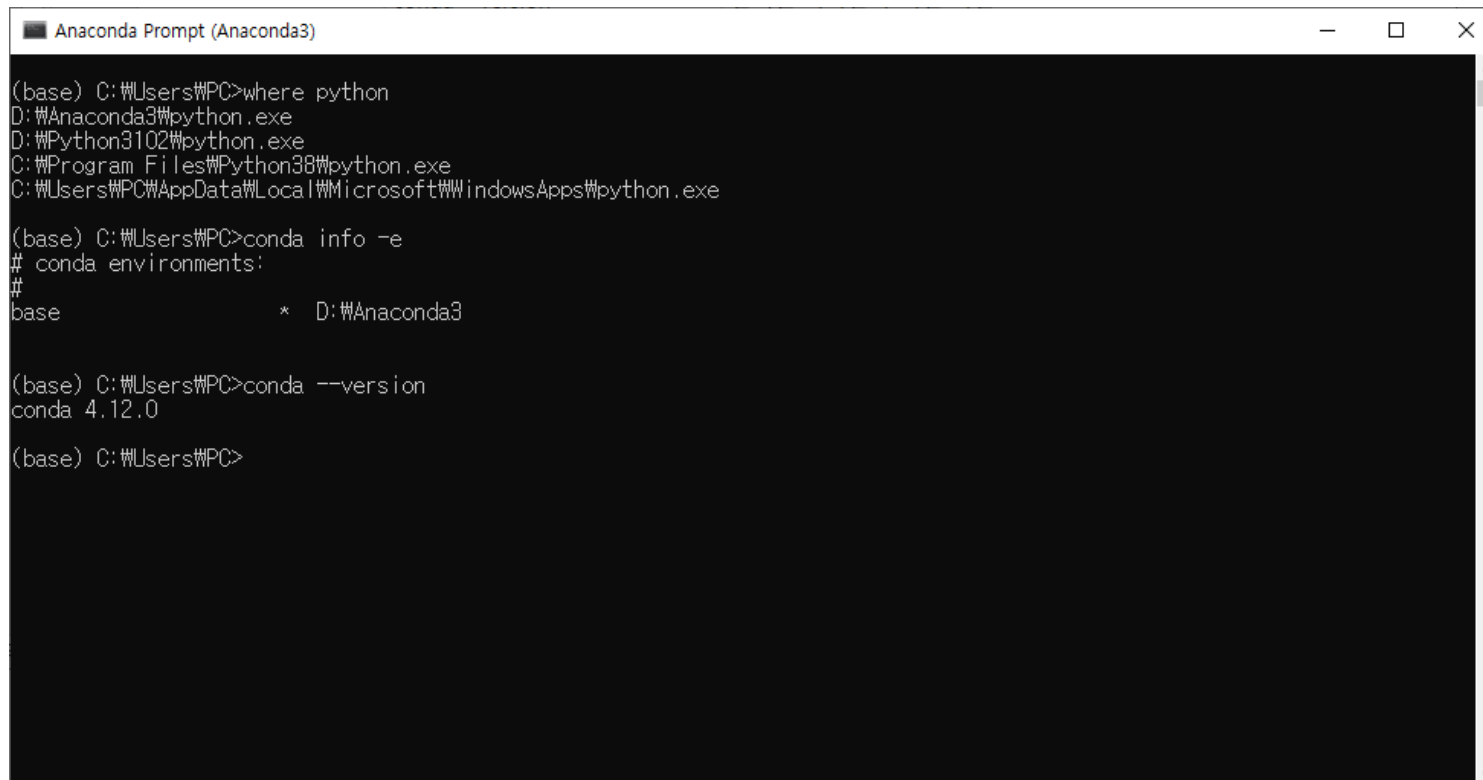
- conda 프롬프트 열기

- 설치 메뉴 Anaconda Prompt(Anaconda3) 실행

- %windir%\System32\cmd.exe "/K" C:\Anaconda3\Scripts\activate.bat C:\Anaconda3

- 기본

- (base) 가상 환경



```

Anaconda Prompt (Anaconda3)

(base) C:\Users\WPC>where python
D:\Anaconda3\python.exe
D:\Python3102\python.exe
C:\Program Files\Python38\python.exe
C:\Users\WPC\AppData\Local\Microsoft\WindowsApps\python.exe

(base) C:\Users\WPC>conda info -e
# conda environments:
#
base                * D:\Anaconda3

(base) C:\Users\WPC>conda --version
conda 4.12.0

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

# 아나콘다 conda

가상환경 Python language

- 가상환경이나 모듈을 관리하는 명령어
  - 항상 conda로 시작

명령어	설명
conda --version	설치된 아나콘다 버전 확인
conda clean	설치된 패키지를 모두 삭제
conda create	새로운 가상환경 생성
conda config	설정 보기, 신규 설정, 수정등
conda info	설치된 아나콘다 정보
conda install	패키지 설치
conda list	설치된 패키지 정보(환경 별로 다름)
conda remove	설치된 패키지 삭제
conda search	설치된 패키지 조회
conda uninstall	Alias for conda remove
conda update	최신 버전으로 업데이트
conda upgrade	Alias for conda update

# 아나콘다의 conda 명령 기본

가상환경 Python language

- **conda --version**
  - 현재 콘다 버전
- **conda update conda**
  - 콘다 최신 버전으로
- **conda info -e**
  - 현재 가상환경 목록

```

Anaconda Prompt (Anaconda3)

(base) C:\Users\WPC>conda update conda
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: D:\Anaconda3

added / updated specs:
- conda

The following packages will be downloaded:

package | build | size
-----|-----|-----
conda-4.13.0 | py39haa95532_0 | 923 KB
Total: 923 KB

The following packages will be UPDATED:

conda 4.12.0-py39haa95532_0 --> 4.13.0-py39haa95532_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.13.0 | 923 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: failed

EnvironmentNotWritableError: The current user does not have write permissions to the target environment.
environment location: D:\Anaconda3

(base) C:\Users\WPC>

```

# 아나콘다의 conda 명령어

가상환경 Python language

- **conda list**
  - conda list 패키지명
    - 패키지 관련 설치 모듈과 버전 정보 제공
- **conda install**
- **conda search 패키지명**
  - 저장소에 있는 패키지 검색
  - 설치전에 설치할 패키지를 한번 찾아본 후 버전 등을 보고 설치

```

Anaconda Prompt (Anaconda3)

(base) C:\Users\WPC>conda list tensorflow
# packages in environment at D:\Anaconda3:
#
# Name                      Version           Build Channel
#
(base) C:\Users\WPC>conda search tensorflow
Loading channels: done
# Name                      Version           Build Channel
tensorflow                  1.7.0             0 pkgs/main
tensorflow                  1.7.1             0 pkgs/main
tensorflow                  1.8.0             0 pkgs/main
tensorflow                  1.8.0             haa95532_0 pkgs/main
tensorflow                  1.9.0 eigen_py35hb0e21f4_1 pkgs/main
tensorflow                  1.9.0 eigen_py36h0b764b7_1 pkgs/main
tensorflow                  1.9.0 gpu_py35h0075c17_1 pkgs/main
tensorflow                  1.9.0 gpu_py36hfddee9c2_1 pkgs/main
tensorflow                  1.10.0 eigen_py35h38c8211_0 pkgs/main
tensorflow                  1.10.0 eigen_py36h849fbd8_0 pkgs/main
tensorflow                  1.10.0 gpu_py35ha5d5ef7_0 pkgs/main
tensorflow                  1.10.0 gpu_py36h3514669_0 pkgs/main
tensorflow                  1.10.0 mkl_py35h4a0f5c2_0 pkgs/main
tensorflow                  1.10.0 mkl_py36hb361250_0 pkgs/main
tensorflow                  1.11.0 eigen_py36h346fd36_0 pkgs/main
  
```

# 아나콘다의 conda 명령어

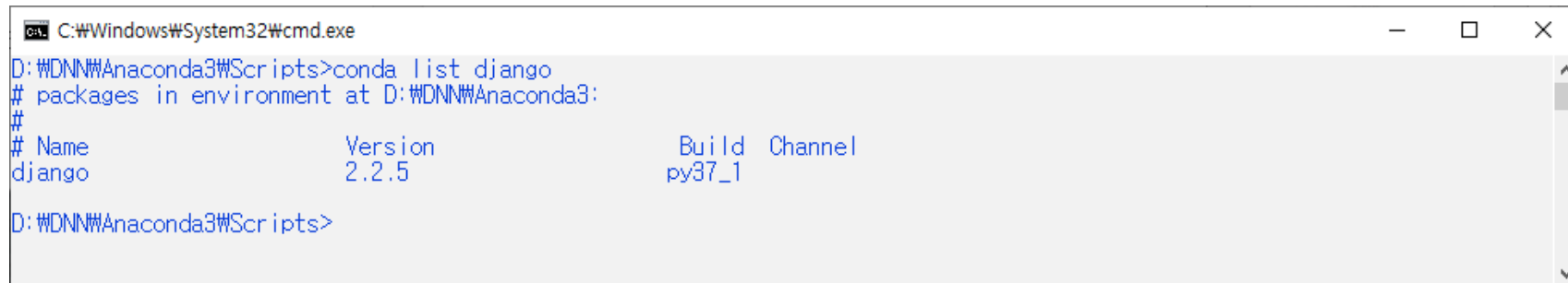
가상환경 Python language

명령어	설명
conda --version	설치된 아나콘다 버전 확인
conda clean	설치된 패키지를 모두 삭제
conda create	새로운 가상환경 생성
conda config	설정 보기, 신규 설정, 수정등
conda info	설치된 아나콘다 정보
conda install	패키지 설치
conda list	설치된 패키지 정보(환경 별로 다름)
conda remove	설치된 패키지 삭제
conda search	설치된 패키지 조회
conda uninstall	Alias for conda remove
conda update	최신 버전으로 업데이트
conda upgrade	Alias for conda update

## 장고 설치 및 삭제

가상환경 Python language

- `conda list django`
- `conda search django`
- `conda install django`
- `conda remove django`



```
C:\Windows\System32\cmd.exe
D:\DNN\Anaconda3\Scripts>conda list django
# packages in environment at D:\DNN\Anaconda3:
#
# Name            Version           Build    Channel
django            2.2.5             py37_1
D:\DNN\Anaconda3\Scripts>
```



AI Experts  
Who Lead  
The Future

## 04

### 아나콘다의 conda를 사용한 가상환경 생성

- DMU*Ai*** 동양미래대학교  
인공지능소프트웨어학과

## 모듈과 가상환경 삭제

가상환경 Python language

- `conda uninstall numpy`
- `conda deactivate`
- `conda env remove -n kang`
- `conda remove --name kang --all`  
- 가상환경명 kang 삭제
- `conda info -e`
- `conda env list`

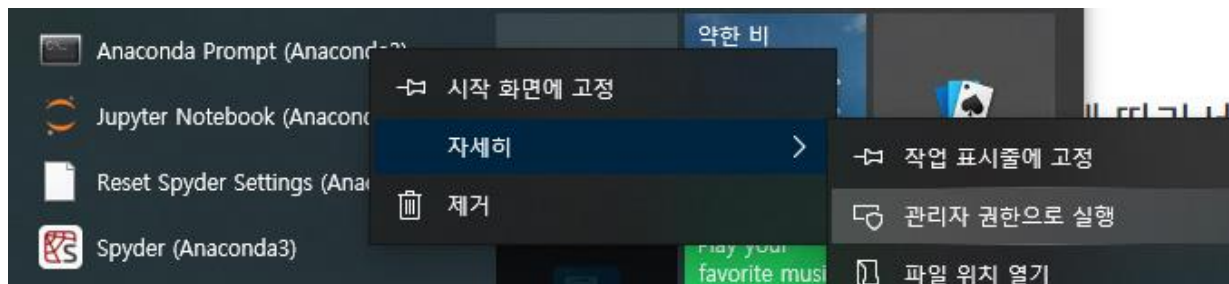
```

C:\Windows\System32\cmd.exe - conda install numpy - conda uninstall nump...
(kang) D:\Users>conda list numpy
# packages in environment at D:\DNN\Anaconda3\envs\kang:
#
# Name                               Version                               Build Channel
(kang) D:\Users>conda deactivate
(base) D:\Users>conda env remove -n kang
Remove all packages in environment D:\DNN\Anaconda3\envs\kang:
(base) D:\Users>conda info -e
# conda environments:
#
base                                * C:\Python\Anaconda3
gpu1.13                             D:\DNN\Anaconda3\envs\gpu1.13
kgpu14                              D:\DNN\Anaconda3\envs\kgpu14
ktf14                               D:\DNN\Anaconda3\envs\ktf14
ktf14gpu                            D:\DNN\Anaconda3\envs\ktf14gpu
tf20                                D:\DNN\Anaconda3\envs\tf20

(base) D:\Users>
  
```

- Only me로 설치한 경우에는 일반 모드/ 관리자 모드 모두 동일
- 전체 사용자(all users)로 설치한 경우
  - 명령어를 사용하는 셸(도스 프롬프트)에 따라 폴더 상이
  - 따라서, anaconda 설치 시 all user로 설치한 경우에는
    - 가상환경 생성시 관리자 모드로 생성하는 것을 권장
- Conda 설치 시
  - 생성되는 모듈이 최소

- Case: 전체 사용자로 설치 후
- 메뉴 바로 쉘을 실행 해 가상환경 생성
  - 프롬프트가 다음으로 실행
    - (base) C:\Users\WPC>
  - > conda create -n ml
    - 가상환경 ml이 다음 폴더로 생성
    - C:\사용자\<사용자명>\.conda\ml(가상환경이름)
- 메뉴 '관리자 권한으로 실행' 으로 쉘을 실행 해 가상환경 생성
  - 프롬프트가 다음으로 실행
    - (base) C:\Windows\system32>
  - > conda create -n ml
    - 가상환경 ml이 다음 폴더로 생성
    - D:\Anaconda3\envs\ml(가상환경이름)



```
선택 Anaconda Prompt (Anaconda3)
(base) C:\Users\PC>conda info -e
# conda environments:
#
test                C:\Users\PC\Anaconda3\envs\test
base                * D:\Anaconda3
num                D:\Anaconda3\envs\num
test2              D:\Anaconda3\envs\test2

(base) C:\Users\PC>conda create -n ml
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.12.0
  latest version: 4.13.0

Please update conda by running

    $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\PC\Anaconda3\envs\ml

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate ml
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#

(base) C:\Users\PC>conda info -e
# conda environments:
#
ml                  C:\Users\PC\Anaconda3\envs\ml
test                C:\Users\PC\Anaconda3\envs\test
base                * D:\Anaconda3
num                D:\Anaconda3\envs\num
test2              D:\Anaconda3\envs\test2

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

```
관리자: Anaconda Prompt (Anaconda3)
test C:\Users\PC\Anaconda3\envs\test
base * D:\Anaconda3
test2 D:\Anaconda3\envs\test2

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

==> WARNING: A newer version of conda exists. <==
  current version: 4.12.0
  latest version: 4.13.0

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: D:\Anaconda3\envs\num

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate num
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#

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

# 가상환경 test로 연습

- 생성
  - conda create -n test
- 설치 모듈 조회
  - conda activate test
  - conda list
  - conda deactivate
- 제거
  - conda env remove -n test

```
Anaconda Prompt (Anaconda3) - conda deactivate

(base) C:\Users\PC>conda create -n test
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.12.0
  latest version: 4.13.0

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

  environment location: C:\Users\PC\conda\envs\test

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate test
#
# To deactivate an active environment, use
#
#   $ conda deactivate
#

(base) C:\Users\PC>conda info -e
# conda environments:
#
test                  C:\Users\PC\conda\envs\test
base                  * D:\Anaconda3

(base) C:\Users\PC>conda activate test

(test) C:\Users\PC>conda list
# packages in environment at C:\Users\PC\conda\envs\test:
#
# Name                      Version            Build Channel
(test) C:\Users\PC>conda deactivate

(base) C:\Users\PC>conda env remove -n test

Remove all packages in environment C:\Users\PC\conda\envs\test:

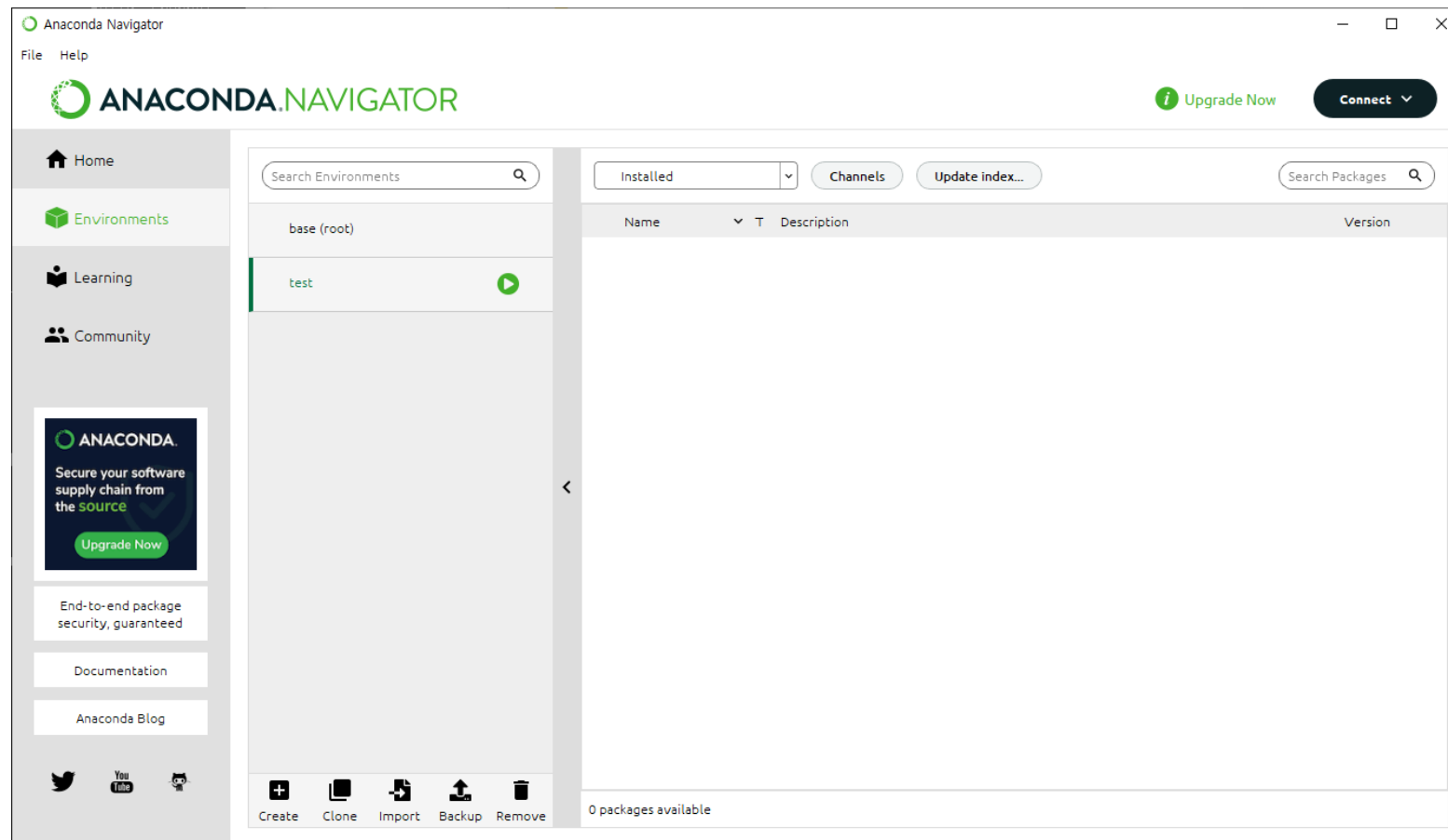
(base) C:\Users\PC>conda info -e
# conda environments:
#
base                  * D:\Anaconda3

(base) C:\Users\PC>_
```

가상환경 Python language



- 생성된 폴더
  - C:\사용자\<계정이름>\.conda\<가상환경>



- **conda 프롬프트 열기**
  - 설치 메뉴 Anaconda Prompt(Anaconda3) 실행
    - %windir%\System32\cmd.exe "/K" C:\Anaconda3\Scripts\activate.bat C:\Anaconda3
    - cmd /k cd %homepath% 로 열어서 다음 실행
      - D:\Anaconda3\Scripts\activate.bat
      - 또는 conda activate
- **가상환경 생성 및 모듈 설치 동시**
  - conda create -n 가상환경 설치모듈1 설치모듈2 설치모듈3 ...
- **가상환경 활성화**
  - conda activate 가상환경
- **필요하면 모듈 추가 설치**
  - conda install 설치모듈4 설치모듈5 설치모듈6
- **가상환경 비활성화**
  - conda deactivate

윈도 기본 기본변수

> echo %windir%  
C:\windows

## 수업에서의 가상환경 3개

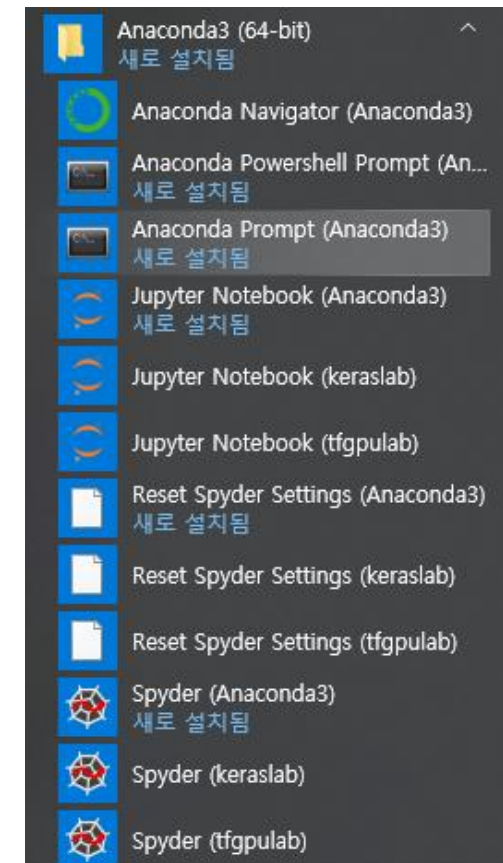
가상환경 Python language

- 다음 폴더에서 작업, 폴더는 큰 의미는 없음
  - C:WVEWcenv
- 가상환경 num
  - 설치 모듈
    - numpy
- 가상환경 data
  - 설치 모듈
    - pandas
- 가상환경 vis
  - 설치 모듈
    - bokeh

# 아나콘다 프롬프트에서 가상환경 생성

가상환경 Python language

- 도스 창 실행됨
  - 프롬프트 앞에 '(가상환경이름)'이 있음
    - 기본적으로 (base)
  - (base) ...>
- 가상환경 생성
  - 이름
    - num
  - 설치 모듈
    - numpy
  - 명령어
    - `conda create -n num numpy`
    - `conda create -n 가상환경 설치모듈1 설치모듈2`



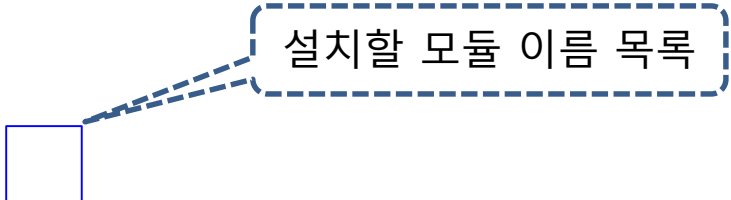
- 현재 가상환경 확인
  - `conda info -e`
- 가상환경 생성과 모듈 `numpy` 설치 명령
  - `conda create -n num numpy`
- 다시 가상환경 확인, 만들어진 가상환경 `num` 확인 필요
  - `conda info -e`
    - 지금 새로 생성된 가상환경 `num` 확인
- 가상환경으로 진입, 활성화
  - `conda activate num`
    - 가상환경 활성화해 내부로 진입, 가상환경 이름이 맨 앞에 표시
    - `(num) D:\WVEW\cenv>`
- 새로이 생성된 가상환경 내부에서 설치된 모듈 점검
  - `conda list numpy`

설치할 모듈 이름 목록

- **모듈 설치**
  - 모듈 matplotlib, pandas, jupyter도 함께 설치
- **필요 명령**
  - conda info -e
  - conda create -n data matplotlib pandas jupyter
  - conda info -e
    - 지금 새로 생성된 가상환경 data 확인
  - conda activate data
    - 가상환경 활성화해 내부로 진입
    - 가상환경 이름이 맨 앞에 표시
- **새로이 생성된 가상환경 내부에서 설치된 모듈 점검**
  - conda list matplotlib
  - conda list pandas
  - conda list numpy
  - conda list jupyter

설치할 모듈 이름 목록

- 모듈 설치
  - 모듈 bokeh 설치
- 필요 명령
  - conda info -e
  - conda create -n vis
  - conda info -e
    - 지금 새로 생성된 가상환경 vis 확인
  - conda activate vis
    - 가상환경 활성화해 내부로 진입
    - 가상환경 이름이 맨 앞에 표시
- 가상환경 생성 후 패키지 bokeh 설치
  - conda install bokeh
  - conda로 생성되는 빈 가상환경에는 모듈은 아무 것도 설치되지 않음
    - pip를 사용 불가능
- 새로이 생성된 가상환경 내부에서 설치된 모듈 점검
  - conda list bokeh
  - conda list numpy



설치할 모듈 이름 목록

```
선택 Anaconda Prompt (Anaconda3) - conda install python==3.6.12 - conda deactivate
done
#
# To activate this environment, use
#
#     $ conda activate num
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) C:\Users\PC>conda info -e
# conda environments:
#
num                C:\Users\PC\conda\envs\num
test               C:\Users\PC\conda\envs\test
base               * D:\Anaconda3

(base) C:\Users\PC>conda activate num

(num) C:\Users\PC>conda list
# packages in environment at C:\Users\PC\conda\envs\num:
#
# Name                                Version                                Build                                Channel
blas                                  1.0                                    mkl
bz2ip2                                1.0.8                                he774522_0
ca-certificates                       2022.4.26                             haa95532_0
certifi                               2022.6.15                             py310haa95532_0
intel-openmp                          2021.4.0                              haa95532_3556
libffi                                3.4.2                                hd77b12b_4
mkl                                    2021.4.0                              haa95532_640
mkl-service                           2.4.0                                py310h2bbff1b_0
mkl_fft                               1.3.1                                py310ha0764ea_0
mkl_random                            1.2.2                                py310h4edbf06_0
numpy                                  1.22.3                                py310h6d2d95c_0
numpy-base                            1.22.3                                py310h206c741_0
openssl                                1.1.1p                                h2bbff1b_0
pip                                    21.2.4                                py310haa95532_0
python                                3.10.4                                hbb2ffb3_0
setuptools                             61.2.0                                py310haa95532_0
six                                    1.16.0                               pyhd3eb1b0_1
sqlite                                 3.38.5                                h2bbff1b_0
tk                                     8.6.12                                h2bbff1b_0
tzdata                                 2022a                                  hda174b7_0
vc                                      14.2                                  h21ffa451_1
vs2015_runtime                        14.27.29016                           h5e58377_2
wheel                                  0.37.1                               pyhd3eb1b0_0
wincertstore                           0.2                                  py310haa95532_2
xz                                      5.2.5                                h8cc25b3_1
zlib                                    1.2.12                               h8cc25b3_2

(num) C:\Users\PC>
```



# Conda로 만든 빈 가상환경

가상환경 Python language

- Pip는 가상환경의 pip가 아님
  - 설치되지 않음
- 다른 패키지를 설치하면 pip가 설치
  - 이후 pip 사용 가능

```

Anaconda Prompt (Anaconda3)

(base) C:\Users\Kang Py>d:

(base) D:\>conda create -n vis
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: D:\Anaconda3\envs\vis

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate vis
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#

(base) D:\>conda info --e
# conda environments:
#
#
#          C:\Users\Kang Py\Anaconda3
#          C:\Users\Kang Py\Anaconda3\envs\cprj
#          * D:\Anaconda3
#          base          D:\Anaconda3\envs\1 Python Books)
#          bok          D:\Anaconda3\envs\bok
#          data          D:\Anaconda3\envs\data
#          data2         D:\Anaconda3\envs\data2
#          data3         D:\Anaconda3\envs\data3
#          dnn           D:\Anaconda3\envs\dnn
#          k1data        D:\Anaconda3\envs\k1data
#          num           D:\Anaconda3\envs\num
#          vis           D:\Anaconda3\envs\vis
#          D:\VE\test\cve

(base) D:\>conda activate vis

(vis) D:\>conda list
# packages in environment at D:\Anaconda3\envs\vis:
#
# Name                               Version           Build Channel
#
(vis) D:\>where pip
D:\Python\Python38-32\Scripts\pip.exe
C:\Users\Kang Py\AppData\Local\Programs\Python\Python38-32\Scripts\pip.exe

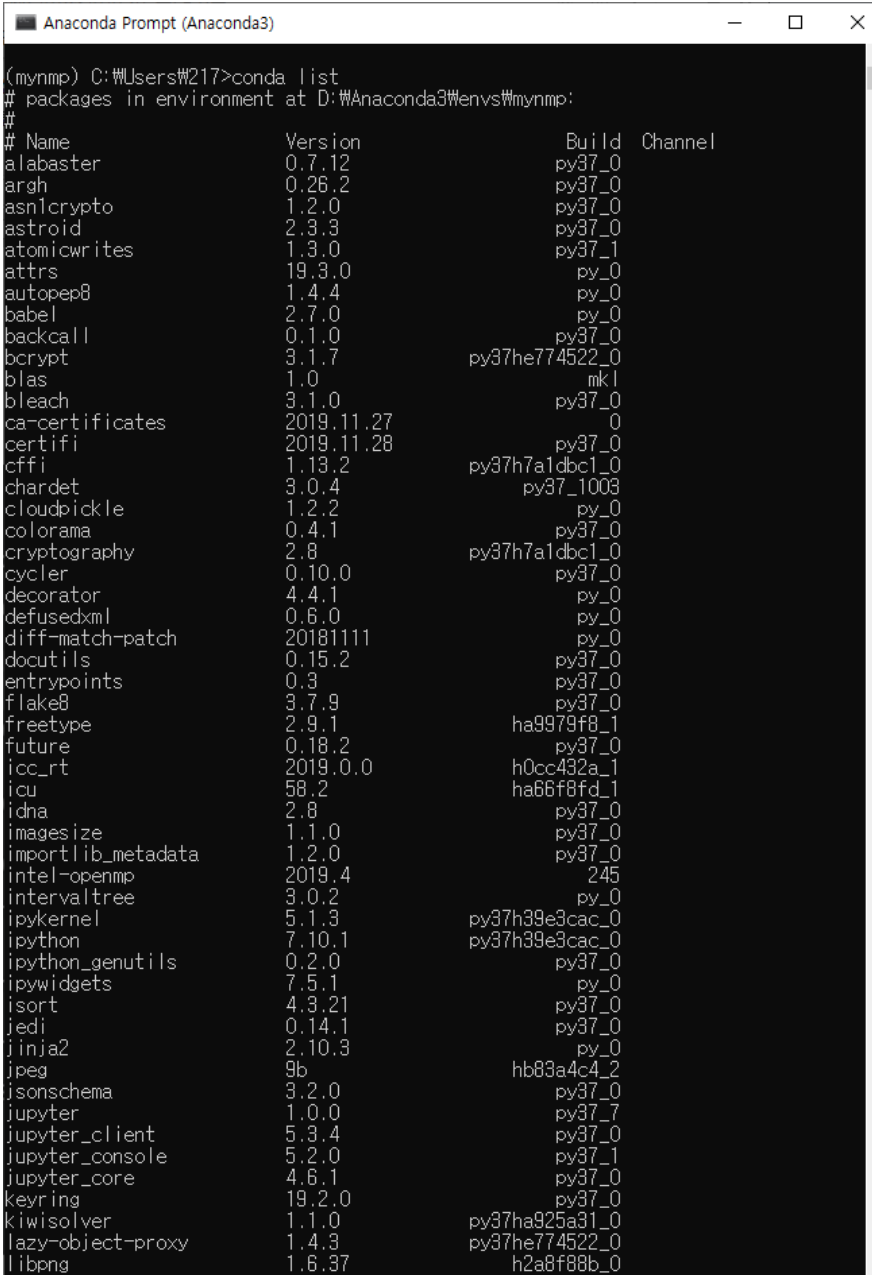
(vis) D:\>

```

# 설치 모듈 전체 확인

가상환경 Python language

- conda list



```

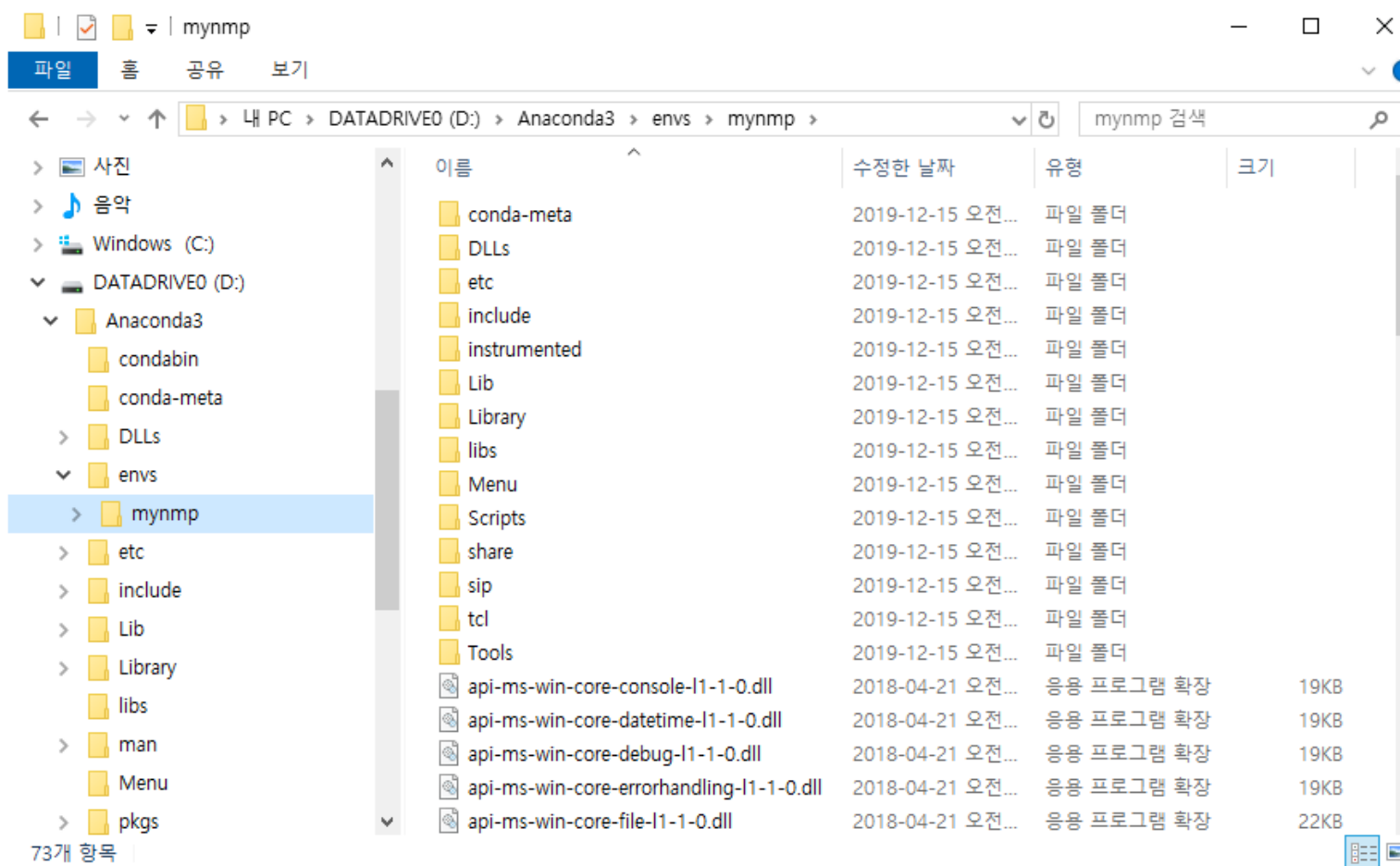
(mynmp) C:\Users\217>conda list
# packages in environment at D:\Anaconda3\envs\mynmp:
#
# Name                                Version                                Build                                Channel
alabaster                             0.7.12                                py37_0                              py37_0
argh                                  0.26.2                                py37_0                              py37_0
asn1crypto                             1.2.0                                py37_0                              py37_0
astroid                                2.3.3                                py37_0                              py37_0
atomicwrites                           1.3.0                                py37_1                              py37_1
attrs                                  19.3.0                                py_0                                py_0
autopep8                               1.4.4                                py_0                                py_0
babel                                  2.7.0                                py_0                                py_0
backcall                               0.1.0                                py37_0                              py37_0
bcrypt                                 3.1.7                                py37he774522_0                      py37he774522_0
blas                                   1.0                                    mkl                                  mkl
bleach                                 3.1.0                                py37_0                              py37_0
ca-certificates                       2019.11.27                            0                                    0
certifi                               2019.11.28                            py37_0                              py37_0
cffi                                   1.13.2                                py37h7a1dbc1_0                      py37h7a1dbc1_0
chardet                               3.0.4                                py37_1003                           py37_1003
cloudpickle                             1.2.2                                py_0                                py_0
colorama                              0.4.1                                py37_0                              py37_0
cryptography                           2.8                                    py37h7a1dbc1_0                      py37h7a1dbc1_0
cyclor                                 0.10.0                                py37_0                              py37_0
decorator                              4.4.1                                py_0                                py_0
defusedxml                             0.6.0                                py_0                                py_0
diff-match-patch                       20181111                              py_0                                py_0
docutils                               0.15.2                                py37_0                              py37_0
entrypoints                             0.3                                    py37_0                              py37_0
flake8                                 3.7.9                                py37_0                              py37_0
freetype                               2.9.1                                ha9979f8_1                          ha9979f8_1
future                                 0.18.2                                py37_0                              py37_0
icc_rt                                 2019.0.0                              h0cc432a_1                          h0cc432a_1
icu                                    58.2                                   ha66f8fd_1                          ha66f8fd_1
idna                                   2.8                                    py37_0                              py37_0
imagesize                              1.1.0                                py37_0                              py37_0
importlib_metadata                     1.2.0                                py37_0                              py37_0
intel-openmp                           2019.4                                245                                  245
intervaltree                           3.0.2                                py_0                                py_0
ipykernel                              5.1.3                                py37h39e3cac_0                      py37h39e3cac_0
ipython                                7.10.1                                py37h39e3cac_0                      py37h39e3cac_0
ipython_genutils                       0.2.0                                py37_0                              py37_0
ipywidgets                             7.5.1                                py_0                                py_0
isort                                  4.3.21                                py37_0                              py37_0
jedi                                    0.14.1                                py37_0                              py37_0
jinja2                                 2.10.3                                py_0                                py_0
jpeg                                   9b                                    hb83a4c4_2                          hb83a4c4_2
jsonschema                             3.2.0                                py37_0                              py37_0
jupyter                                1.0.0                                py37_7                              py37_7
jupyter_client                         5.3.4                                py37_0                              py37_0
jupyter_console                        5.2.0                                py37_1                              py37_1
jupyter_core                           4.6.1                                py37_0                              py37_0
keyring                                19.2.0                                py37_0                              py37_0
kiwisolver                             1.1.0                                py37ha925a31_0                      py37ha925a31_0
lazy-object-proxy                       1.4.3                                py37he774522_0                      py37he774522_0
libpng                                 1.6.37                                h2a8f88b_0                          h2a8f88b_0

```

# 가상환경 확인

가상환경 Python language

- 폴더 생성 확인
  - [아나콘다설치폴더]/envs/가상환경이름



- 하단 Create 버튼

**Create new environment**

Name:

Location:

Packages: ☒ Python   
☐ R

**Installed** Channels Update index... Search Packages

Name	Description	Version
✓ bzip2	High-quality data compressor	1.0.8
✓ ca-certificates	Certificates for use with other packages.	2022.4.26
✓ certifi	Python package for providing mozilla's ca bundle.	2022.6.15
✓ libffi	A portable Foreign Function interface library.	3.4.2
✓ openssl	Openssl is an open-source implementation of the ssl and tls protocols	1.1.1p
✓ pip	Pypa recommended tool for installing python packages	21.2.4
✓ python	General purpose programming language	3.10.4
✓ setuptools	Download, build, install, upgrade, and uninstall python packages	61.2.0
✓ sqlite	Implements a self-contained, zero-configuration, sql database engine	3.38.5
✓ tk	A dynamic programming language with gui support. bundles tcl and tk.	8.6.12
✓ tzdata	The time zone database (called tz, tzdb or zoneinfo)	2022a
✓ vc	A meta-package to impose mutual exclusivity among software built with different vs versions	14.2
✓ vs2015_runtime	Msvc runtimes associated with cl.exe version 19.27.29111 (vs 2019 update 5)	14.27.29...
✓ wheel	A built-package format for python.	0.37.1
✓ wincertstore	Python module to extract ca and crt certs from windows' cert store (ctypes based).	0.2
✓ xz	Data compression software with high compression ratio	5.2.5
✓ zlib	Massively spiffy yet delicately unobtrusive compression library	1.2.12

# Matplotlib 설치

가상환경 Python language

Not installed

Channels

Update index...

matplotlib X

Name	Description	Version
<input checked="" type="checkbox"/> matplotlib	Publication quality figures in python	3.5.1
<input type="checkbox"/> basemap	Plot on map projections using matplotlib	1.3.2
<input type="checkbox"/> basemap-data	Plot on map projections (with coastlines and political boundaries) using matplotlib.	1.3.2
<input type="checkbox"/> descartes	Use geometric objects as matplotlib paths and patches.	1.1.0
<input type="checkbox"/> ipympl	Matplotlib jupyter extension	
<input type="checkbox"/> matplotlib-base	Publication quality figures in python	
<input type="checkbox"/> matplotlib-inline	Inline matplotlib backend for jupyter	
<input type="checkbox"/> mpl-scatter-density	Matplotlib helpers to make density scatter plots	
<input type="checkbox"/> mpld3	D3 viewer for matplotlib.	
<input type="checkbox"/> basemap-data-hires	Plot on map projections (with coastlines and political boundaries) using matplotlib.	

10 packages available matching "matplotlib" 1 package selected

Install Packages

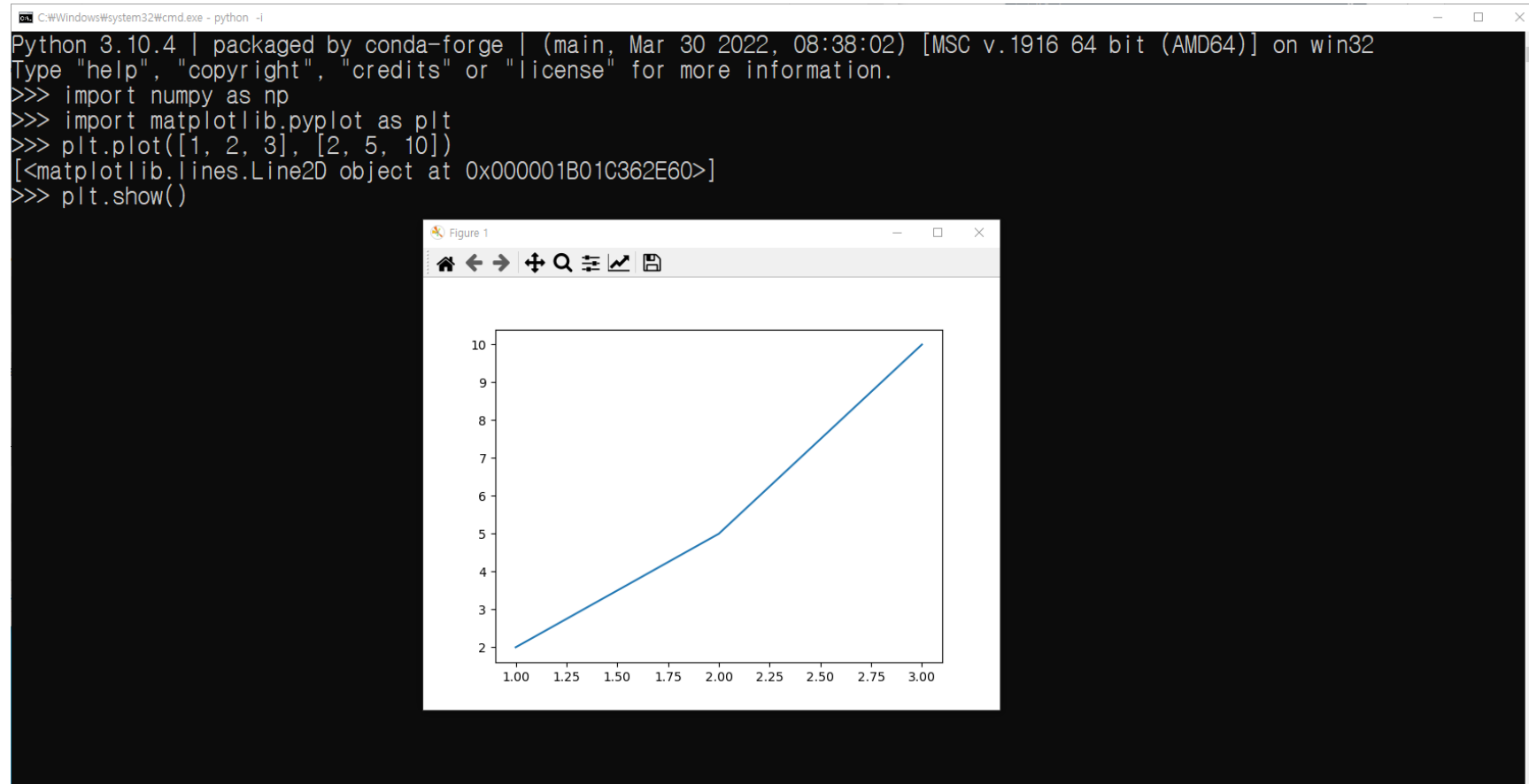
32 packages will be installed

	Name	Unlink	Link	Channel	Action
1	*zstd	-	1.5.2	pkgs/main	Install
2	*tornado	-	6.1	pkgs/main	Install
3	*six	-	1.16.0	pkgs/main	Install
4	*sip	-	4.19.13	pkgs/main	Install
5	*qt	-	5.9.7	pkgs/main	Install
6	*python-dateutil	-	2.8.2	pkgs/main	Install

\* indicates the package is a dependency of a selected package

Cancel

Apply



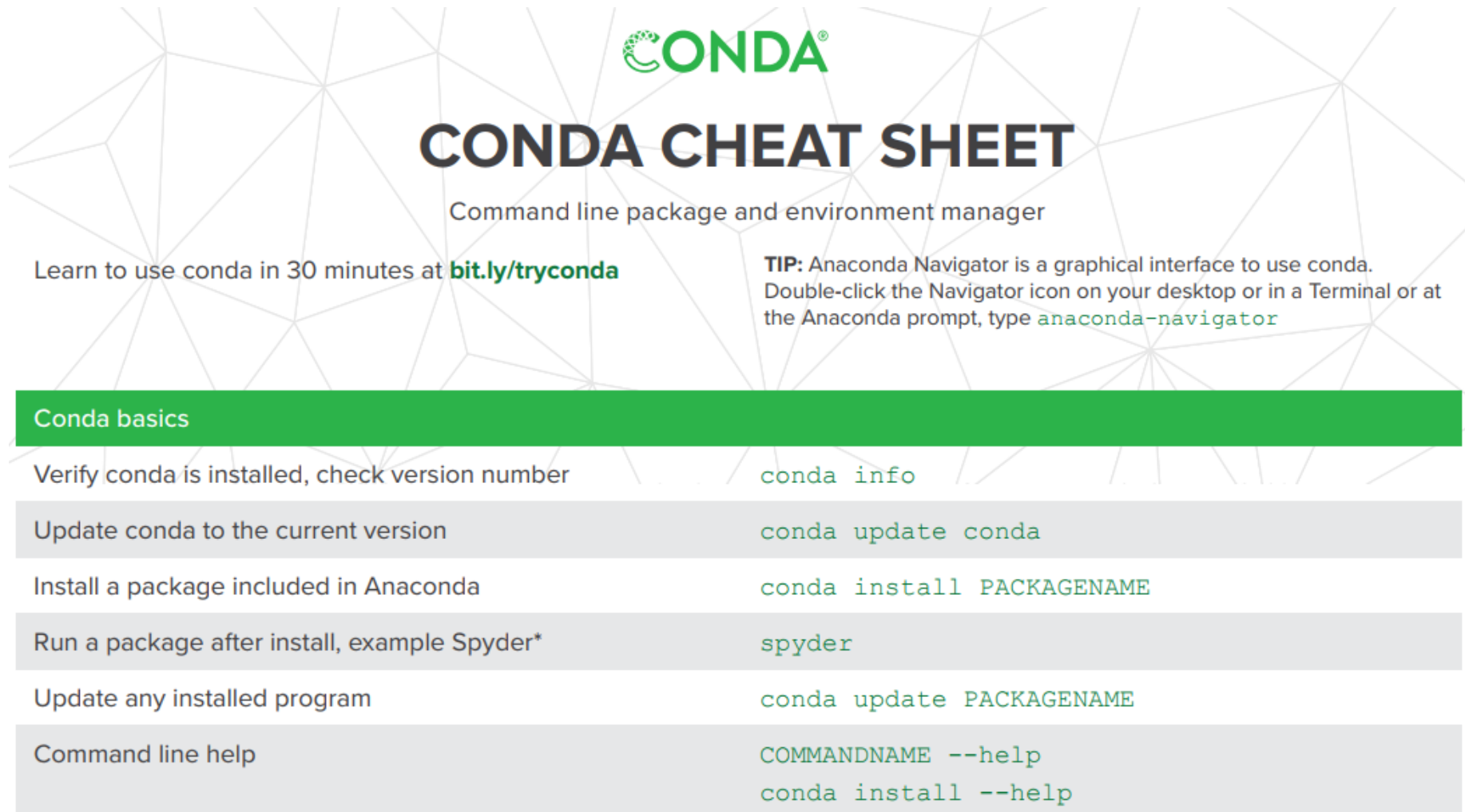
## 가상환경 생성 실습(20분)

가상환경 Python language

- 각각 지정한 가상환경 3개를 conda로 생성
  - 가상환경 vnp 생성, numpy 설치
    - `conda create -n vnp numpy`
  - 가상환경 vpd 생성, pandas 설치
    - `conda create -n vpd pandas`
  - 가상환경 vtf 생성, tensorflow 설치
    - `conda create -n vmat matplotlib`

# Conda cheat sheet





**CONDA**  
**CONDA CHEAT SHEET**  
 Command line package and environment manager

Learn to use conda in 30 minutes at [bit.ly/tryconda](https://bit.ly/tryconda)

**TIP:** Anaconda Navigator is a graphical interface to use conda. Double-click the Navigator icon on your desktop or in a Terminal or at the Anaconda prompt, type `anaconda-navigator`

### Conda basics

Verify conda is installed, check version number	<code>conda info</code>
Update conda to the current version	<code>conda update conda</code>
Install a package included in Anaconda	<code>conda install PACKAGENAME</code>
Run a package after install, example Spyder*	<code>spyder</code>
Update any installed program	<code>conda update PACKAGENAME</code>
Command line help	<code>COMMANDNAME --help</code> <code>conda install --help</code>

\*Must be installed and have a deployable command,  
 usually PACKAGENAME

### Using environments

Create a new environment named py35, install Python 3.5	<code>conda create --name py35 python=3.5</code>
Activate the new environment to use it	WINDOWS: <code>activate py35</code> Linux, macOS: <code>source activate py35</code>
Get a list of all my environments, active environment is shown with *	<code>conda env list</code>
Make exact copy of an environment	<code>conda create --clone py35 --name py35-2</code>
List all packages and versions installed in active environment	<code>conda list</code>
List the history of each change to the current environment	<code>conda list --revisions</code>
Restore environment to a previous revision	<code>conda install --revision 2</code>
Save environment to a text file	<code>conda list --explicit &gt; bio-env.txt</code>
Delete an environment and everything in it	<code>conda env remove --name bio-env</code>
Deactivate the current environment	WINDOWS: <code>deactivate</code> macOS, LINUX: <code>source deactivate</code>
Create environment from a text file	<code>conda env create --file bio-env.txt</code>
Stack commands: create a new environment, name it bio-env and install the biopython package	<code>conda create --name bio-env biopython</code>

### Finding conda packages

Use conda to search for a package	<code>conda search PACKAGENAME</code>
See list of all packages in Anaconda	<a href="https://docs.anaconda.com/anaconda/packages/pkg-docs">https://docs.anaconda.com/anaconda/packages/pkg-docs</a>

### Installing and updating packages

Install a new package (Jupyter Notebook)  
in the active environment

```
conda install jupyter
```

Run an installed package (Jupyter Notebook)

```
jupyter-notebook
```

Install a new package (toolz) in a different environment  
(bio-env)

```
conda install --name bio-env toolz
```

Update a package in the current environment

```
conda update scikit-learn
```

Install a package (boltons) from a specific channel  
(conda-forge)

```
conda install --channel conda-forge  
boltons
```

Install a package directly from PyPI into the current active  
environment using pip

```
pip install boltons
```

Remove one or more packages (toolz, boltons)  
from a specific environment (bio-env)

```
conda remove --name bio-env toolz boltons
```

## Managing multiple versions of Python

Install different version of Python in a new environment named py34

```
conda create --name py34 python=3.4
```

Switch to the new environment that has a different version of Python

```
Windows: activate py34
Linux, macOS: source activate py34
```

Show the locations of all versions of Python that are currently in the path

```
Windows: where python
Linux, macOS: which -a python
```

**NOTE:** The first version of Python in the list will be executed.

Show version information for the current active Python

```
python --version
```

## Specifying version numbers

Ways to specify a package version number for use with conda create or conda install commands, and in meta.yaml files.

Constraint type	Specification	Result
Fuzzy	<code>numpy=1.11</code>	1.11.0, 1.11.1, 1.11.2, 1.11.18 etc.
Exact	<code>numpy==1.11</code>	1.11.0
Greater than or equal to	<code>"numpy&gt;=1.11"</code>	1.11.0 or higher
OR	<code>"numpy=1.11.1 1.11.3"</code>	1.11.1, 1.11.3
AND	<code>"numpy&gt;=1.8,&lt;2"</code>	1.8, 1.9, not 2.0

**NOTE:** Quotation marks must be used when your specification contains a space or any of these characters: > < | \*

conda --version	아나콘다 버전확인
conda install numpy	numpy 설치
conda install numpy selenium scipy	numpy, selenium, scipy 동시 설치
conda install numpy=1.2	numpy 특정 버전(1.2) 설치
conda remove numpy	numpy 제거
conda update conda	conda 업데이트
conda update -all	모든 패키지 업데이트
conda list	설치된 패키지 목록 보여주기
conda list numpy	numpy 설치유무, 버전확인
conda search numpy	(conda 저장소에 패키지유무) numpy 검색
conda create -n py37 python=3.7	python3.7 버전의 파이썬 가상환경 생성
conda create -n py37 python=3.7 anaconda	python3.7 가상환경 생성 + 마지막 anacodna는 아나콘다의 기본패키지 설치
set CONDA_FORCE_32BIT=1 conda create -n py36_32 python=3.6 anaconda conda activate py36_32	base가 64비트일 경우 python3.6 32비트의 가상환경을 지정한 이름(py36_32)으로 설치하고자 할 때
conda env list	가상환경 목록 확인
conda info --envs	가상환경 목록 확인
conda info	현재 conda에 대한 정보
conda remove --name py36_32 --all	py36_32 가상환경 삭제

# 가상환경 설치모듈과 버전 복사 pip 사용

# pip로 패키지 목록 관리하기

가상환경 Python language

- pip는 파이썬 환경의 모든 라이브러리를 조회하여 출력 기능을 제공
  - 파일 "requirements" 파일을 읽어서 기술된 모든 패키지를 설치하는 기능도 제공

```
$ pip freeze > requirements.txt
```

"requirements" 파일에는 현재 설치된 모든 라이브러리 목록이 저장됩니다. "requirements" 파일은 다음과 같은 형태의 정보를 포함합니다.

```
ipython==6.1.0
ipython-genutils==0.2.0
ipywidgets==7.0.0
jedi==0.10.2
Jinja2==2.9.6
jsonschema==2.6.0
jupyter==1.0.0
jupyter-client==5.1.0
jupyter-console==5.2.0
jupyter-core==4.3.0
MarkupSafe==1.0
mistune==0.7.4
nbconvert==5.3.1
nbformat==4.4.0
notebook==5.0.0
pandocfilters==1.4.2
pexpect==4.2.1
pickleshare==0.7.4
prompt-toolkit==1.0.15
```

"requirements" 파일에 기술된 모든 라이브러리를 설치하는 pip 명령은 다음과 같습니다.

```
pip install -r requirements.txt
```

- 가상환경 A를 가상환경 B와 같이
  - 가상환경 B에서 req.txt 생성
    - `pip freeze > req.txt`
  - 가상환경 A를 만들어
    - req.txt를 반영하여 설치
      - `pip install -r req.txt`