데이터프로그래밍 기초 5일차

2026-1 DS Bootcamp

부산대학교
데이터사이언스전문대학원
석사과정 박민서

PUSAN NATIONAL UNIVERSITY
Graduate School of Data Science

CONTENTS

Module

▪ 모듈 (Module)

- 여러 변수와 함수를 가지고 있는 집합체 (.py 파일)
- 표준 모듈: Python에 내장된 모듈　　외부 모듈: 다른 사람이 만들어서 공개한 모듈
- **import '모듈 이름'**

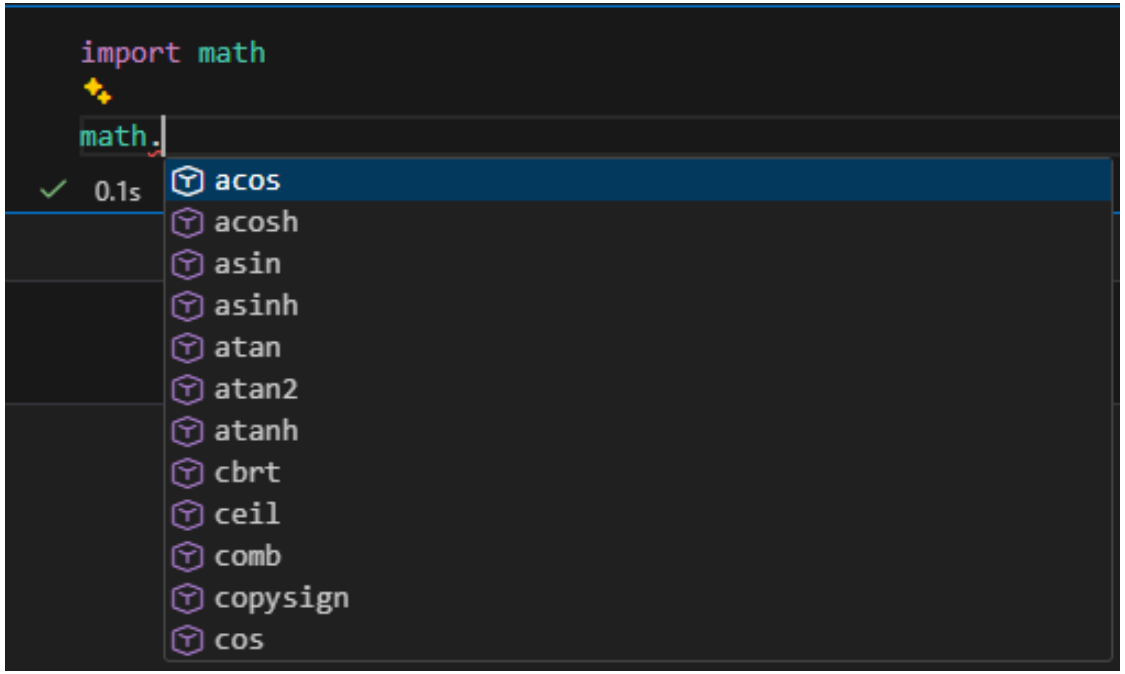
```
import math
✓ 0.1s
```

```
import sys
from _typeshed import SupportsMul, SupportsRMul
from collections.abc import Iterable
from typing import Any, Final, Literal, Protocol, SupportsFloat, SupportsIndex, TypeVar,
from typing_extensions import TypeAlias

_T = TypeVar("_T")
_T_co = TypeVar("_T_co", covariant=True)

_SupportsFloatOrIndex: TypeAlias = SupportsFloat | SupportsIndex

e: Final[float]
pi: Final[float]
inf: Final[float]
nan: Final[float]
tau: Final[float]

def acos(x: _SupportsFloatOrIndex, /) -> float: ...
def acosh(x: _SupportsFloatOrIndex, /) -> float: ...
def asin(x: _SupportsFloatOrIndex, /) -> float: ...
def asinh(x: _SupportsFloatOrIndex, /) -> float: ...
def atan(x: _SupportsFloatOrIndex, /) -> float: ...
def atan2(y: _SupportsFloatOrIndex, x: _SupportsFloatOrIndex, /) -> float: ...
def atanh(x: _SupportsFloatOrIndex, /) -> float: ...

if sys.version_info >= (3, 11):
    def cbrt(x: _SupportsFloatOrIndex, /) -> float: ...
```

math.pyi

▪ 표준 모듈

- math: 수학과 관련된 기능을 가진 모듈



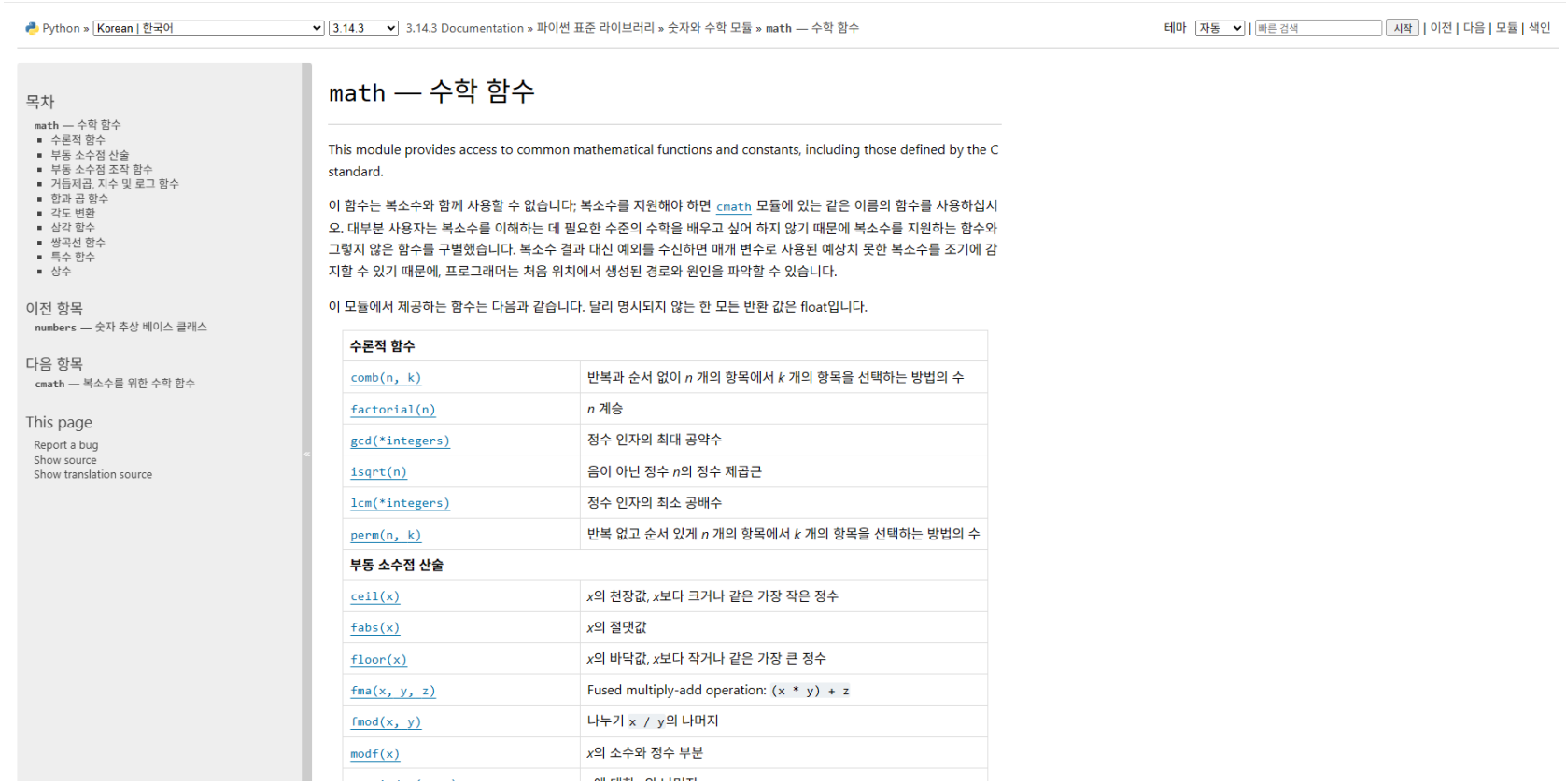

```python
import math

print(math.pi)  # 원주율
print(math.log(16, 2))  # 밑이 2, 진수가 16인 로그값
print(math.ceil(4.12))  # 올림함수 -> 5
print(math.floor(4.832)) # 내림함수 -> 4
```

0.0s

```
3.141592653589793
4.0
5
4
```

▪ 표준 모듈

- math: 수학과 관련된 기능을 가진 모듈
- https://docs.python.org/ko/3/library/math.html



Python » Korean | 한국어 3.14.3 3.14.3 Documentation » 파이썬 표준 라이브러리 » 숫자와 수학 모듈 » math — 수학 함수

테마 자동 | 빠른 검색 시작 | 이전 | 다음 | 모듈 | 색인

목차

math — 수학 함수
- 수론적 함수
- 부동 소수점 산술
- 부동 소수점 조작 함수
- 거듭제곱, 지수 및 로그 함수
- 합과 곱 함수
- 각도 변환
- 삼각 함수
- 쌍곡선 함수
- 특수 함수
- 상수

이전 항목
numbers — 숫자 추상 베이스 클래스

다음 항목
cmath — 복소수를 위한 수학 함수

This page
Report a bug
Show source
Show translation source

math — 수학 함수

This module provides access to common mathematical functions and constants, including those defined by the C standard.

이 함수는 복소수와 함께 사용할 수 없습니다; 복소수를 지원해야 하면 cmath 모듈에 있는 같은 이름의 함수를 사용하십시오. 대부분 사용자는 복소수를 이해하는 데 필요한 수준의 수학을 배우고 싶어 하지 않기 때문에 복소수를 지원하는 함수와 그렇지 않은 함수를 구별했습니다. 복소수 결과 대신 예외를 수신하면 매개 변수로 사용된 예상치 못한 복소수를 조기에 감지할 수 있기 때문에, 프로그래머는 처음 위치에서 생성된 경로와 원인을 파악할 수 있습니다.

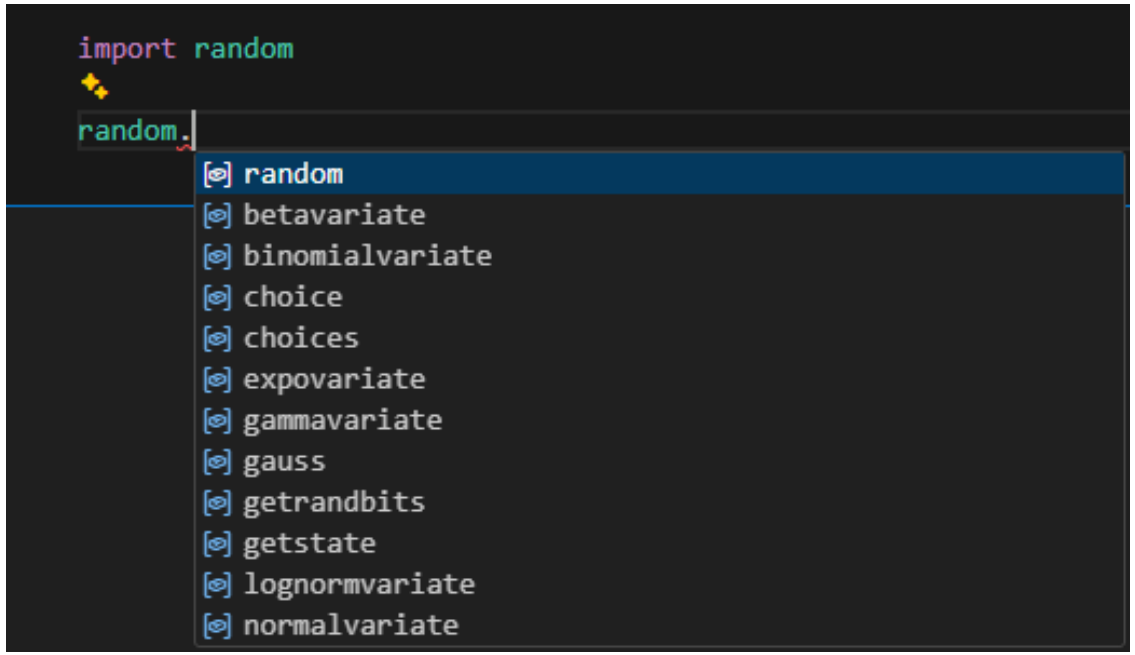
이 모듈에서 제공하는 함수는 다음과 같습니다. 달리 명시되지 않는 한 모든 반환 값은 float입니다.

수론적 함수	
comb(n, k)	반복과 순서 없이 *n* 개의 항목에서 *k* 개의 항목을 선택하는 방법의 수
factorial(n)	*n* 계승
gcd(*integers)	정수 인자의 최대 공약수
isqrt(n)	음이 아닌 정수 *n*의 정수 제곱근
lcm(*integers)	정수 인자의 최소 공배수
perm(n, k)	반복 없고 순서 있게 *n* 개의 항목에서 *k* 개의 항목을 선택하는 방법의 수
부동 소수점 산술	
ceil(x)	*x*의 천장값, *x*보다 크거나 같은 가장 작은 정수
fabs(x)	*x*의 절댓값
floor(x)	*x*의 바닥값, *x*보다 작거나 같은 가장 큰 정수
fma(x, y, z)	Fused multiply-add operation: (x * y) + z
fmod(x, y)	나누기 x / y의 나머지
modf(x)	*x*의 소수와 정수 부분

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽

▪ 표준 모듈

- random: 무작위 값을 생성할 때 사용하는 모듈




```python
import random

# 0~1 사이의 랜덤 수
print(f"random(): {random.random()}")
# 10~20 사이의 랜덤 수
print(f"uniform(10, 20): {random.uniform(10, 20)}")
# 0~10 사이의 랜덤 수
print(f"randrange(10): {random.randrange(10)}")
# 리스트 내에서 k개 만큼 랜덤 샘플링
print(f"sample([1, 2, 3, 4, 5], k=2): {random.sample([1, 2, 3, 4, 5], k=2)}")
```

✓ 0.0s

```
random(): 0.74292644069585
uniform(10, 20): 15.859987424253289
randrange(10): 6
sample([1, 2, 3, 4, 5], k=2): [5, 2]
```

▪ 표준 모듈

- random: 무작위 값을 생성할 때 사용하는 모듈
- https://docs.python.org/ko/3/library/random.html#module-random

Python » Korean | 한국어 3.14.3 3.14.3 Documentation » 파이썬 표준 라이브러리 » 숫자와 수학 모듈 » random — 의사 난수 생성

테마 자동 | 빠른 검색 시작 | 이전 | 다음 | 모듈 | 색인

목차

random — 의사 난수 생성
- 관리 함수
- 바이트열 함수
- 정수 함수
- 시퀀스 함수
- 이산 분포
- 실수 분포
- 대체 생성기
- 재현성에 대한 참고 사항
- 예제
- 조리법
- Command-line usage
- Command-line example

이전 항목
fractions — 유리수

다음 항목
statistics — 수학 통계 함수

This page
Report a bug
Show source
Show translation source

random — 의사 난수 생성

소스 코드: Lib/random.py

이 모듈은 다양한 분포에 대한 의사 난수 생성기를 구현합니다.

정수에 대해서는, 범위에서 균일한 선택이 있습니다. 시퀀스에 대해서는, 무작위 요소의 균일한 선택, 리스트를 제자리(in-place)에서 임의 순열을 생성하는 함수 및 중복 없는(without replacement) 무작위 표본 추출(sampling)을 위한 함수가 있습니다.

실수에 대해서는, 균일(uniform), 정규(normal) (가우시안(Gaussian)), 로그 정규(lognormal), 음의 지수(negative exponential), 감마(gamma) 및 베타(beta) 분포를 계산하는 함수가 있습니다. 각도 분포를 생성하기 위해, 폰 미제스(von Mises) 분포를 사용할 수 있습니다.

거의 모든 모듈 함수는 기본 함수 random()에 의존하는데, 이 함수는 반 열린 구간 0.0 <= X < 1.0 에서 무작위 float를 균일하게 생성합니다. 파이썬은 메르센 트위스터(Mersenne Twister)를 핵심 생성기로 사용합니다. 53비트 정밀도의 float를 생성하며, 주기는 2**19937-1 입니다. C로 작성된 하부 구현은 빠르고 스레드 안전합니다. 메르센 트위스터는 가장 광범위하게 테스트 된 난수 생성기 중 하나입니다. 그러나, 완전히 결정적이므로, 모든 목적에 적합하지는 않으며, 암호화 목적에는 전혀 적합하지 않습니다.

이 모듈에서 제공하는 함수는 실제로는 random.Random 클래스의 숨겨진 인스턴스에 대해 연결된 메서드입니다. Random 인스턴스를 직접 인스턴스 화하여 상태를 공유하지 않는 생성기를 얻을 수 있습니다.

Class Random can also be subclassed if you want to use a different basic generator of your own devising: see the documentation on that class for more details.

random 모듈은 운영 체제에서 제공하는 소스에서 난수를 생성하는 시스템 함수 os.urandom()을 사용하는 SystemRandom 클래스도 제공합니다.

경고: 이 모듈의 의사 난수 생성기를 보안 목적으로 사용해서는 안 됩니다. 보안이나 암호화 용도를 위해서는, secrets 모듈을 참조하십시오.

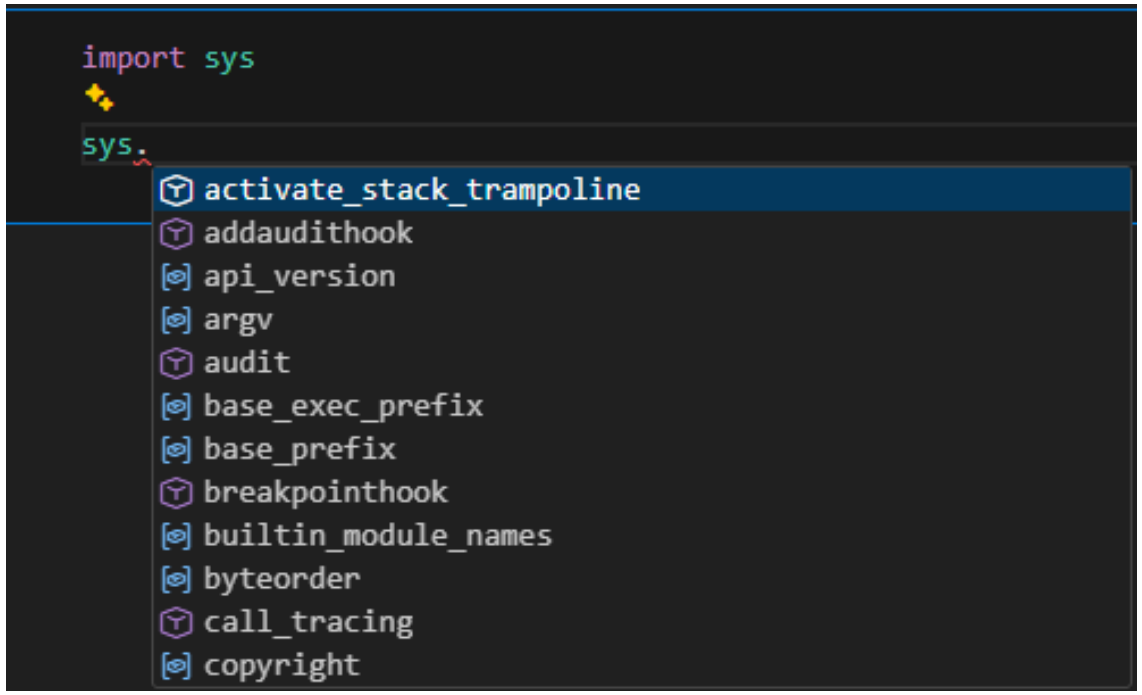
더 보기: M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator", ACM Transactions on Modeling and Computer Simulation Vol. 8, No. 1,

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽

Module

▪ 표준 모듈

- sys: 시스템과 관련된 모듈
- https://docs.python.org/ko/3/library/sys.html#module-sys




```
import sys

# 명령 매개변수 출력
print(sys.argv)
# 컴퓨터 환경 정보 출력
print(sys.getwindowsversion())
print(sys.copyright)
print(sys.version)

# 프로그램 종료
sys.exit()
```

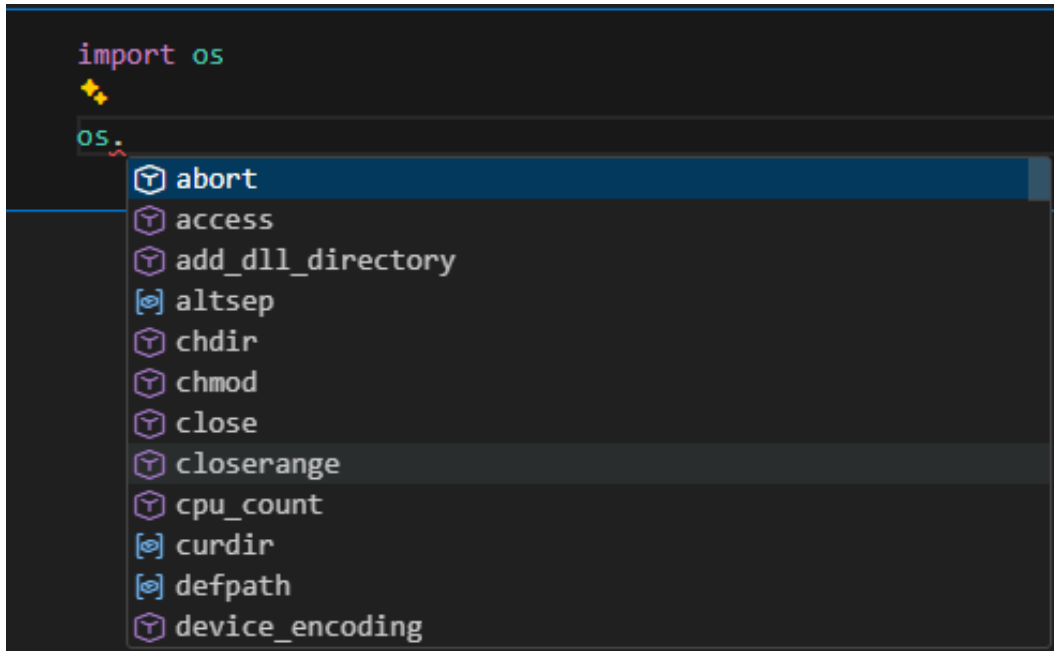
0.0s

```
['C:\\Users\\minsu\\AppData\\Roaming\\Python\\Python312\\site-packages\\ipykernel_launcher.py',
sys.getwindowsversion(major=10, minor=0, build=26200, platform=2, service_pack='')
Copyright (c) 2001-2023 Python Software Foundation.
All Rights Reserved.

Copyright (c) 2000 BeOpen.com.
All Rights Reserved.

Copyright (c) 1995-2001 Corporation for National Research Initiatives.
All Rights Reserved.

Copyright (c) 1991-1995 Stichting Mathematisch Centrum, Amsterdam.
All Rights Reserved.
3.12.6 (tags/v3.12.6:a4a2d2b, Sep  6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)]

An exception has occurred, use %tb to see the full traceback.

SystemExit
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽

▪ 표준 모듈

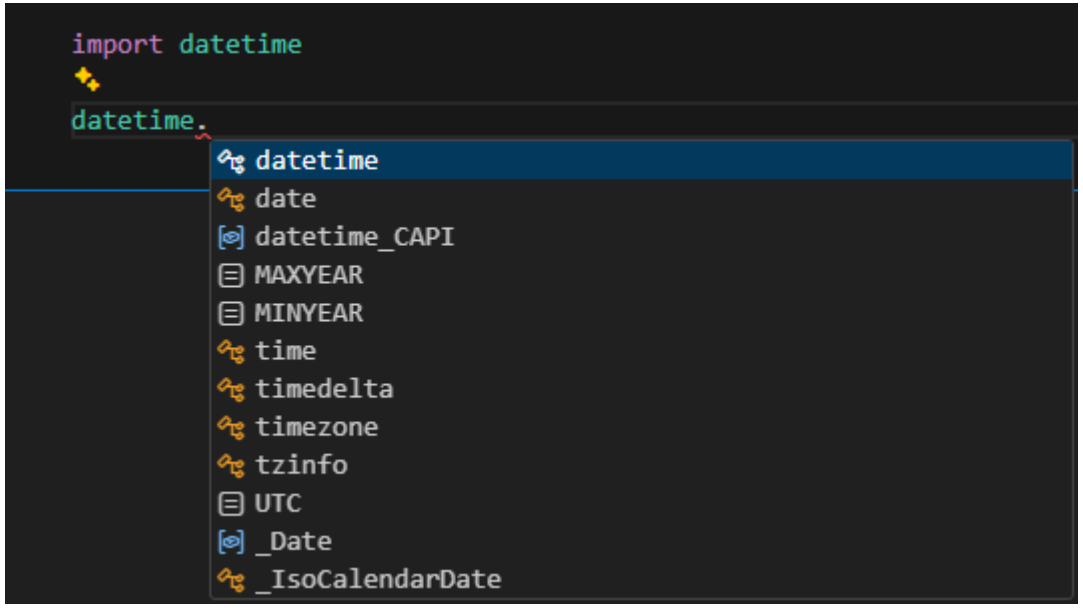
- os: 운영체제와 관련된 모듈 ☞ 폴더 생성, 폴더 내부 파일 탐색
- https://docs.python.org/ko/3/library/os.html#module-os




```python
import os

# 기본 정보
print(f"현재 폴더: {os.getcwd()}")
print(f"현재 폴더 내부: {os.listdir()}")

# 폴더 만들고 제거 (제거는 폴더가 비어있을 때)
os.mkdir("Hello")
os.rmdir("Hello")

# 명령 프롬프트 입력
os.system("dir")
```

▪ 표준 모듈

- datetime: 날짜와 시간과 관련된 모듈
- https://docs.python.org/ko/3/library/datetime.html#module-datetime




```python
import datetime

# 시각 출력
right_now = datetime.datetime.now()
print(f"현재 시각: {right_now}")
print(f"현재 날짜: {right_now.year}년 {right_now.month}월 {right_now.day}일")

# 시간 출력
print(right_now.strftime("%Y-%m-%d %H:%M:%S"))

# 시간 연산
print(f"3시간 뒤: {right_now + datetime.timedelta(hours=3)}")
```

```
0.0s
현재 시각: 2026-02-06 03:23:56.181337
현재 날짜: 2026년 2월 6일
2026-02-06 03:23:56
3시간 뒤: 2026-02-06 06:23:56.181337
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽

▪ from ~ import ~ as

- from ~ import ~: 모듈을 import 할 때, 전체 함수 대신 특정 함수만 불러오는 구문
- as: 함수, 변수명을 새롭게 지정 ☞ 불러온 함수들끼리 이름 충돌 / 이름 축약을 하고 싶을 때

```
from math import log, sin, exp, pow
# math 모듈에서 함수 import
# 여러 개의 함수 동시에 import 가능
```

```
import math as m
# math 모듈을 m으로도 사용 가능

print(math.sin(1))
print(m.sin(1))
```

✓ 0.0s

```
0.8414709848078965
0.8414709848078965
```

```
from datetime import datetime as dt
# datetime 모듈에서 datetime 객체(class)를 dt라는 이름으로 사용

print(dt.now())
```

✓ 0.0s

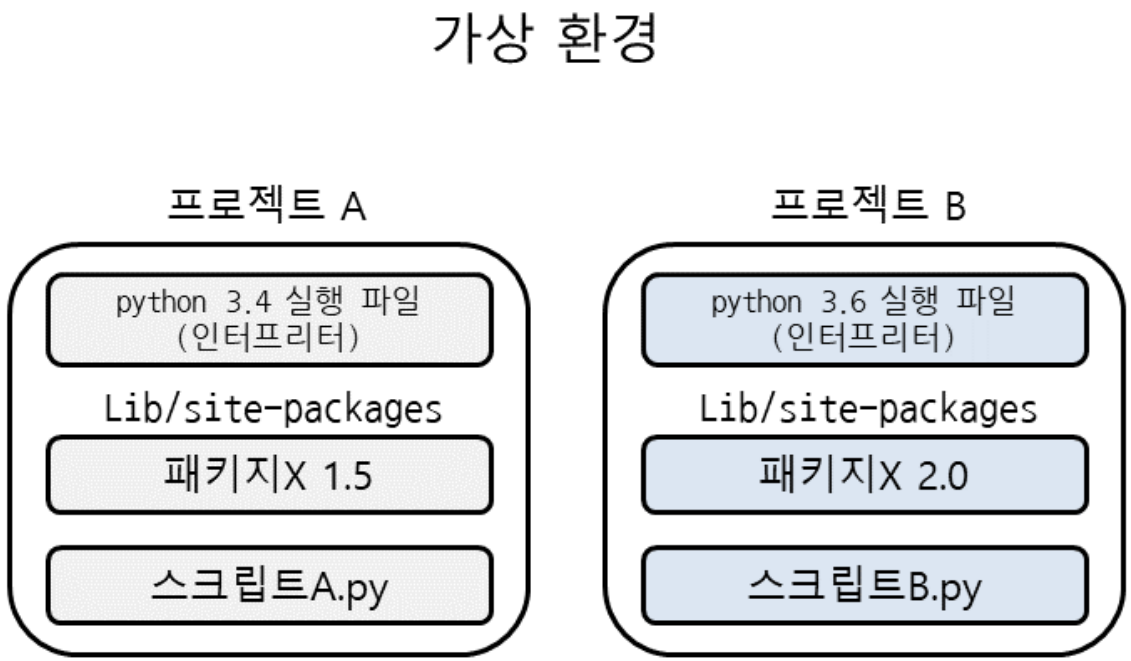
```
2026-02-06 03:29:35.329021
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽

Virtual Environment

▪ 가상환경 (Virtual Environment)

- 독립적인 Python 개발 환경 ☞ 서로 다른 환경에 영향을 줄 수 없음

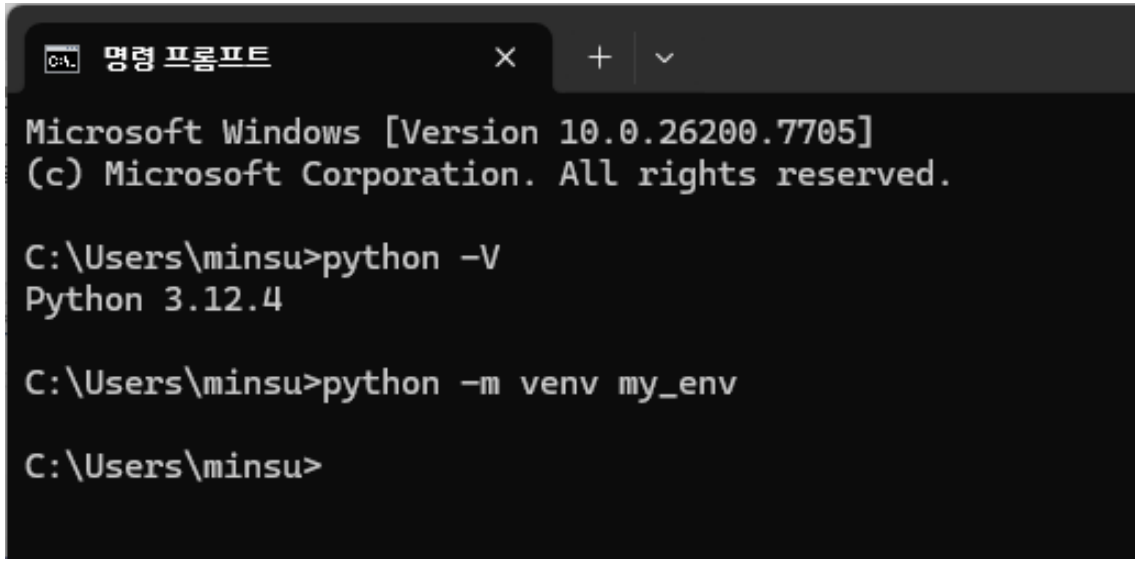


윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ 가상환경 설치

- Window cmd 명령 프롬프트 or VS Code 내부 terminal 이용
- **python –m venv [가상환경이름]**

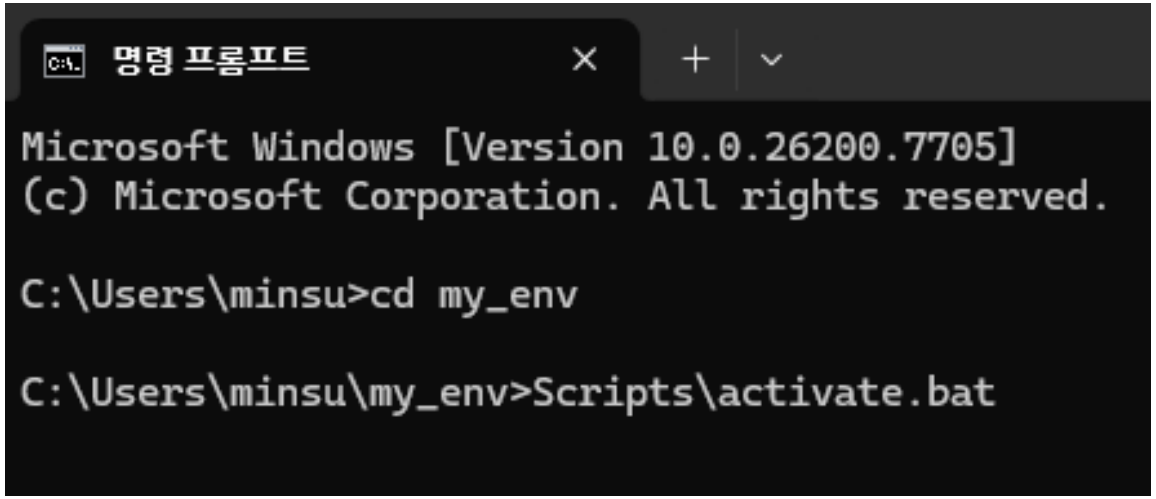


윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ 가상환경 설치

- Window cmd 명령 프롬프트 or VS Code 내부 terminal 이용
- **python –m venv [가상환경이름]**

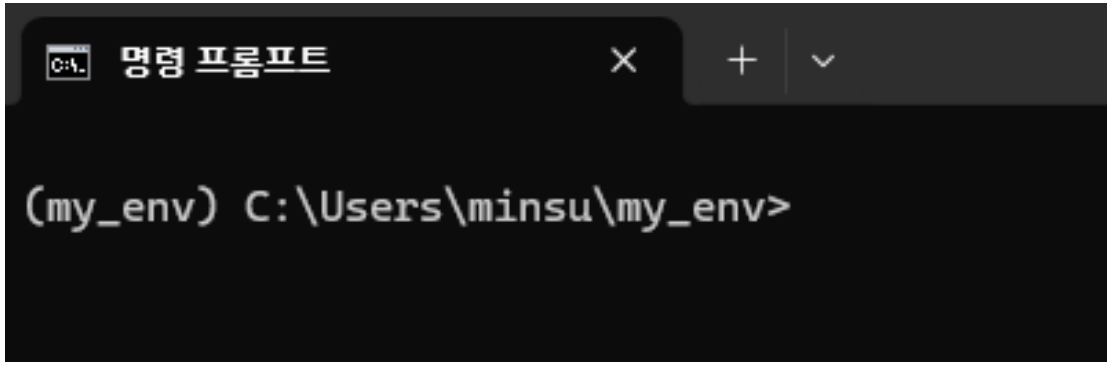


윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ 가상환경 설치

- Window cmd 명령 프롬프트 or VS Code 내부 terminal 이용
- **python –m venv [가상환경이름]**

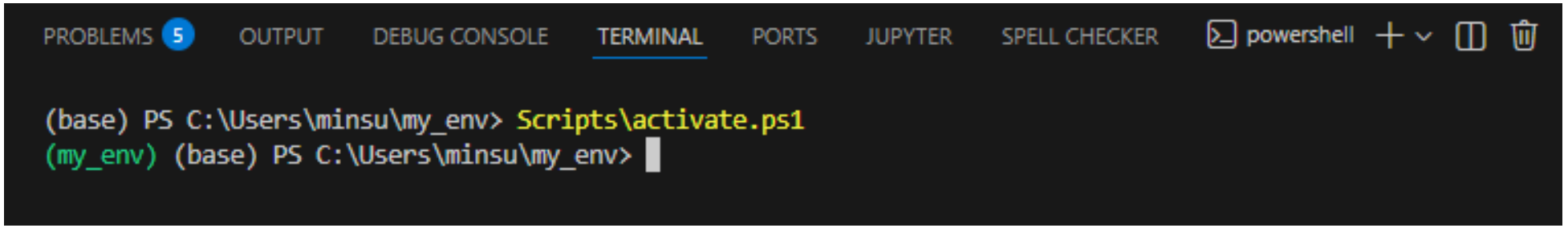


윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ 가상환경 설치

- Window cmd 명령 프롬프트 or VS Code 내부 terminal 이용
- **python –m venv [가상환경이름]**

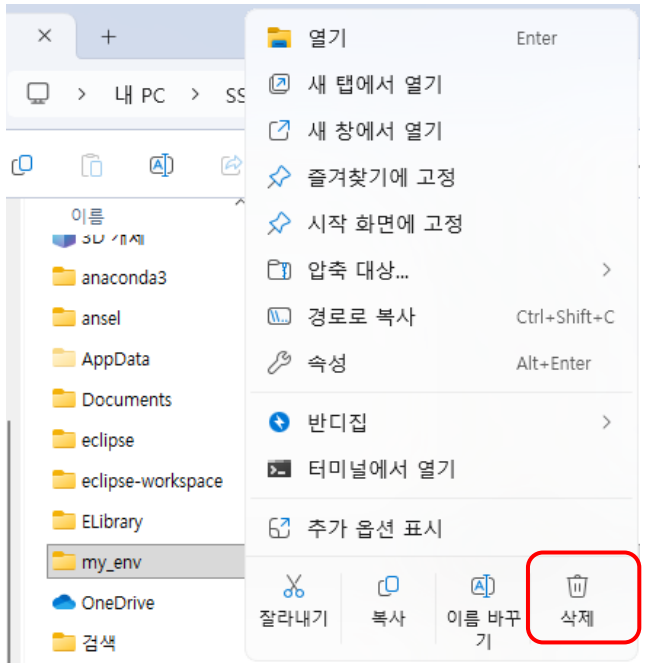


Windows Powershell

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

▪ 가상환경 설치

- Window cmd 명령 프롬프트 or VS Code 내부 terminal 이용
- **python –m venv [가상환경이름]**



가상환경 관리 / 삭제

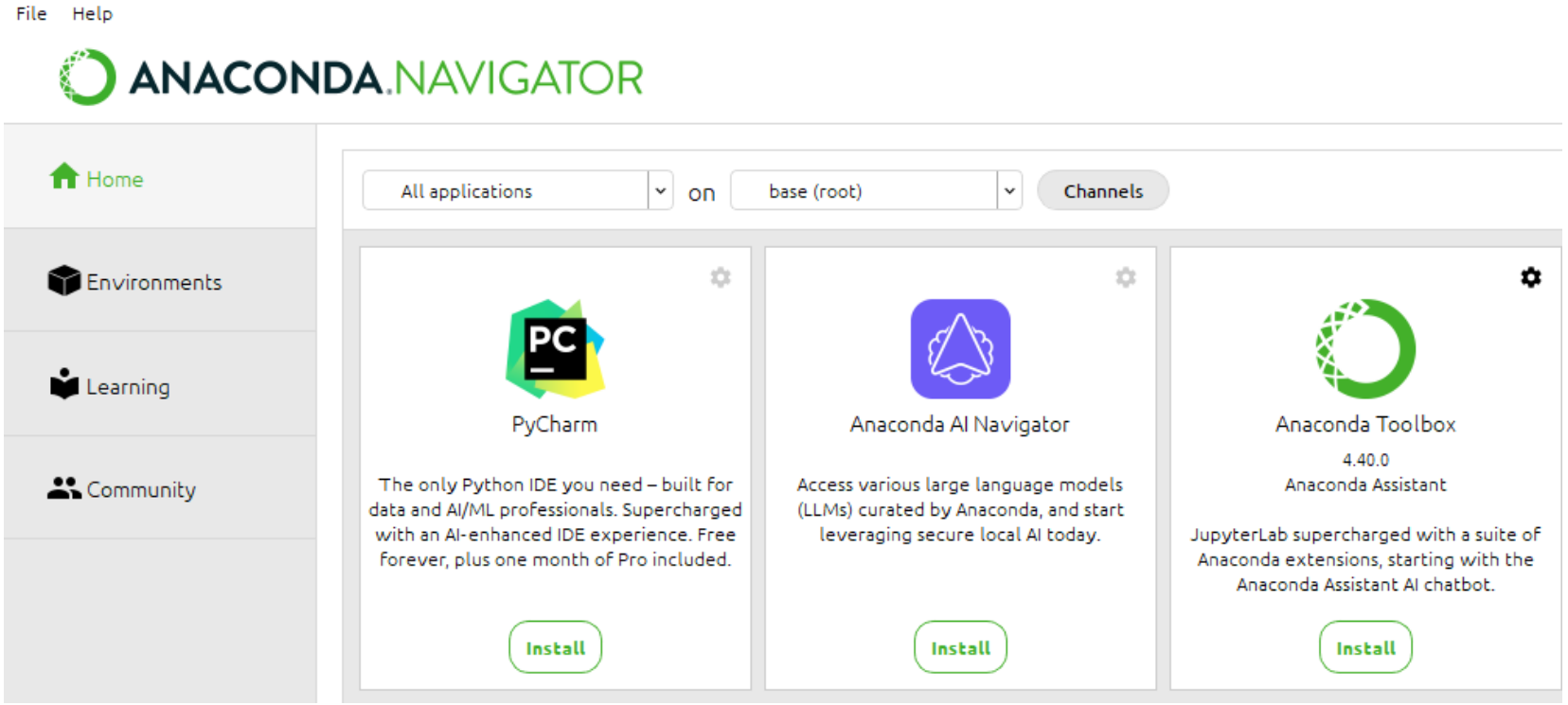
윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Q&A

Virtual Environment

▪ Anaconda 활용 가상환경 설치

- Anaconda Prompt or Anaconda Navigator 활용 or VS Code 이용
- **프롬프트 / 터미널 이용 시: conda create –n [가상환경 이름] [python=[파이썬 버전], 선택]**

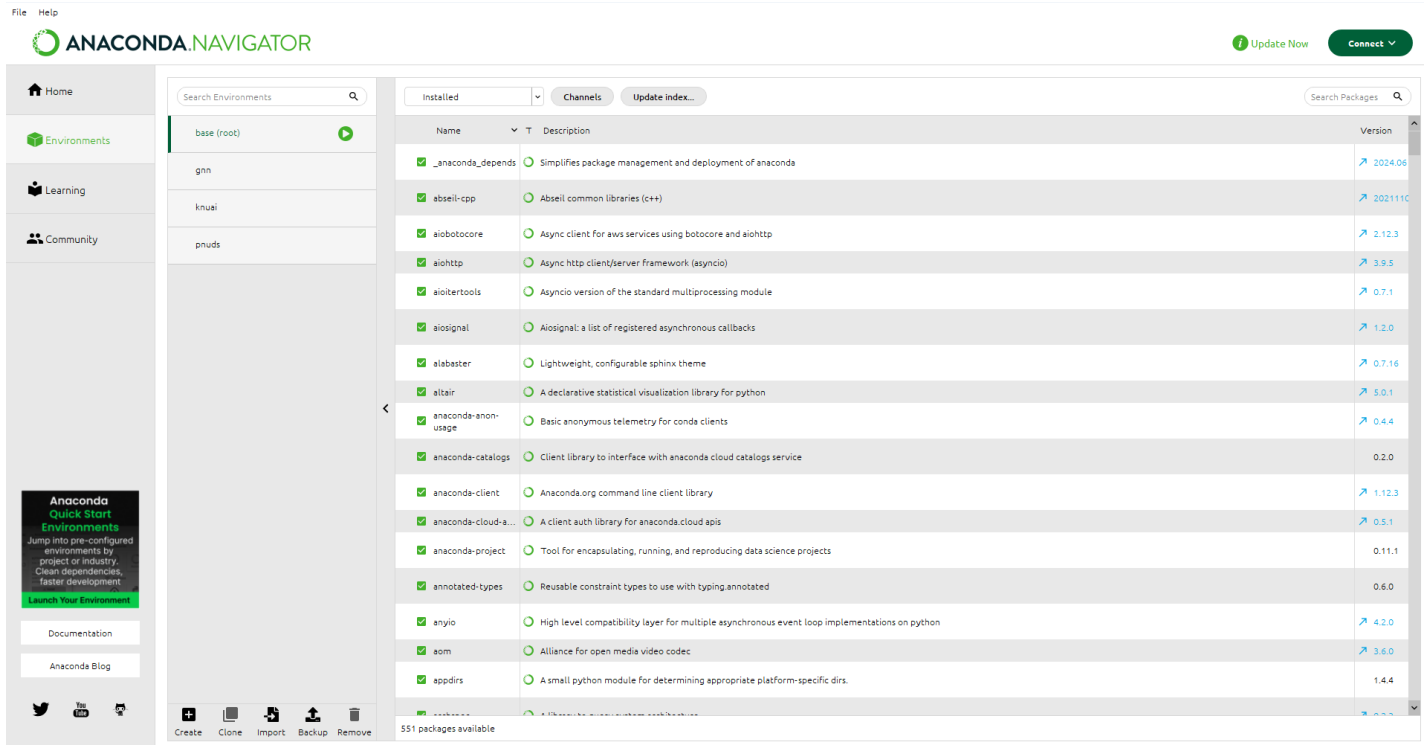


윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ Anaconda 활용 가상환경 설치

- Anaconda Prompt or Anaconda Navigator 활용 or VS Code 이용
- **프롬프트 / 터미널 이용 시: conda create –n [가상환경 이름] [python=[파이썬 버전], 선택]**

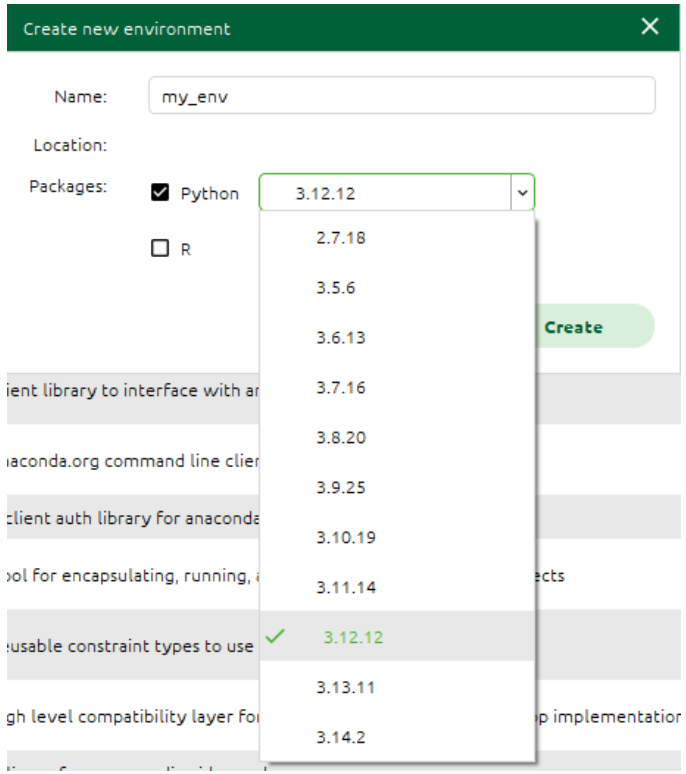


윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ Anaconda 활용 가상환경 설치

- Anaconda Prompt or Anaconda Navigator 활용 or VS Code 이용
- **프롬프트 / 터미널 이용 시: conda create –n [가상환경 이름] [python=[파이썬 버전], 선택]**

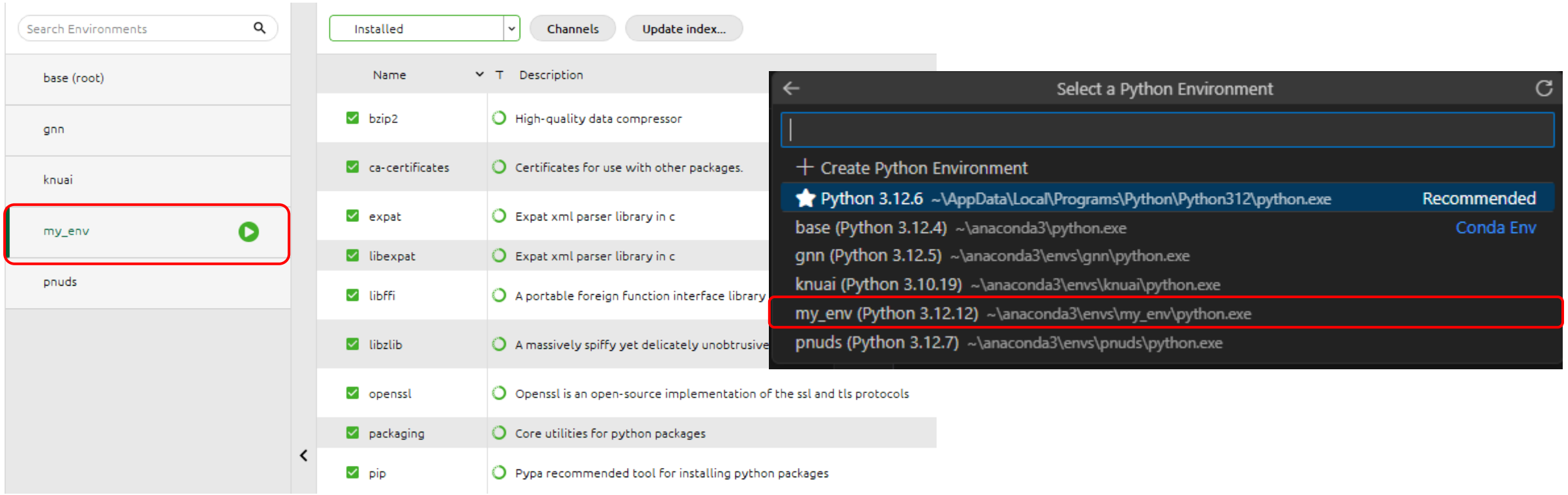


윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ Anaconda 활용 가상환경 설치

- Anaconda Prompt or Anaconda Navigator 활용 or VS Code 이용
- **프롬프트 / 터미널 이용 시: conda create –n [가상환경 이름] [python=[파이썬 버전], 선택]**

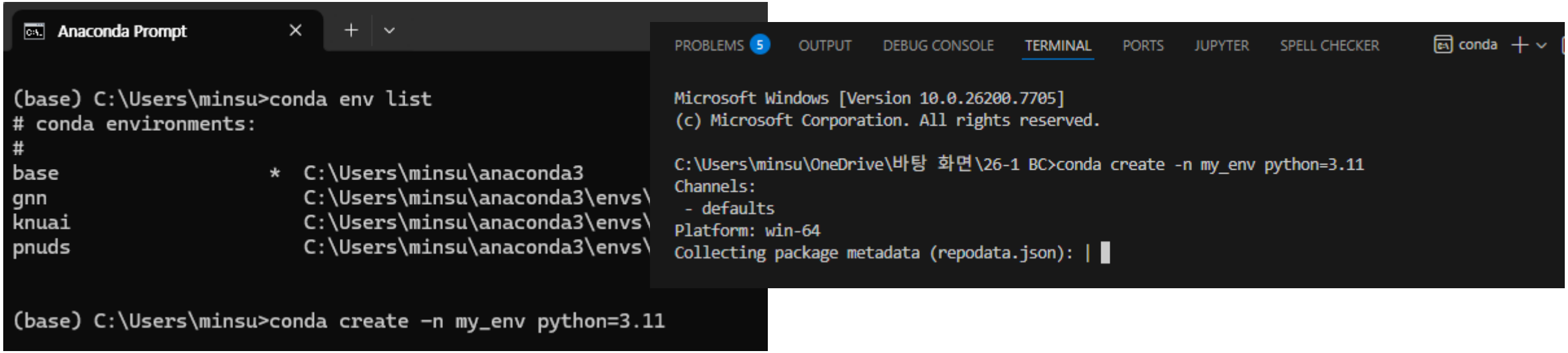


윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ Anaconda 활용 가상환경 설치

- Anaconda Prompt or Anaconda Navigator 활용 or VS Code 이용
- **프롬프트 / 터미널 이용 시: conda create –n [가상환경 이름] [python=[파이썬 버전], 선택]**

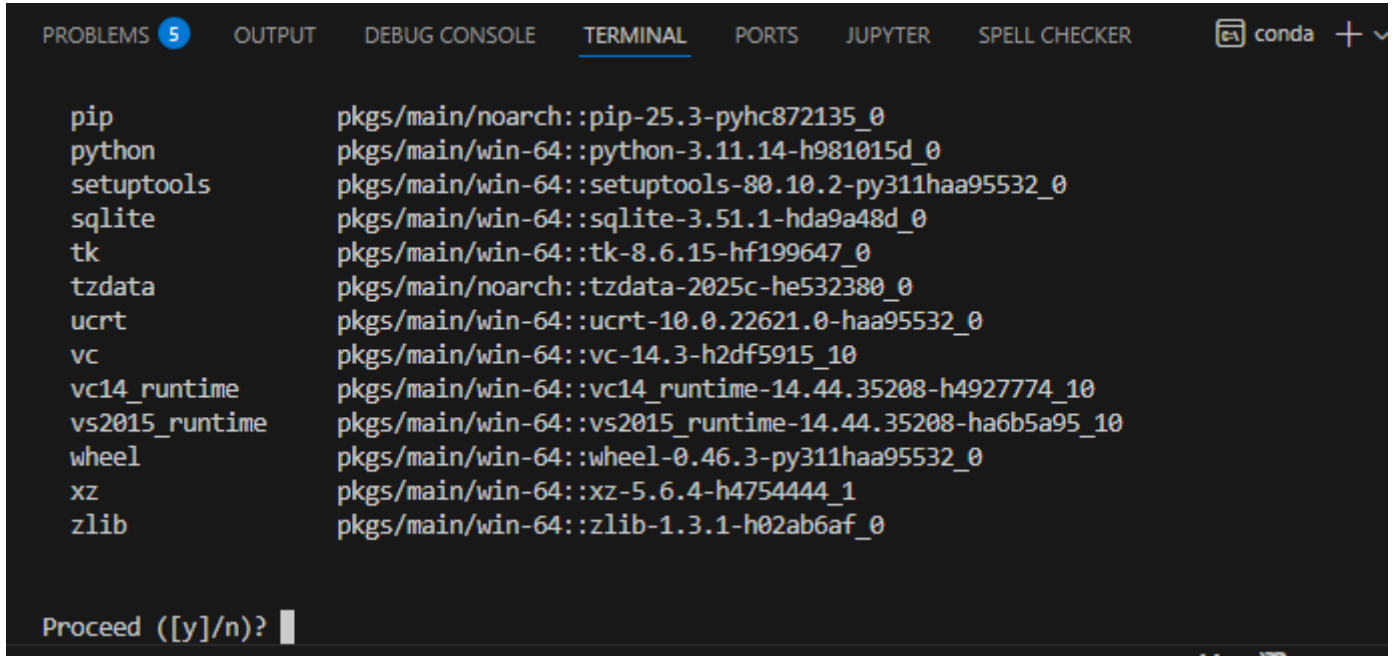


윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ Anaconda 활용 가상환경 설치

- Anaconda Prompt or Anaconda Navigator 활용 or VS Code 이용
- **프롬프트 / 터미널 이용 시: conda create –n [가상환경 이름] [python=[파이썬 버전], 선택]**

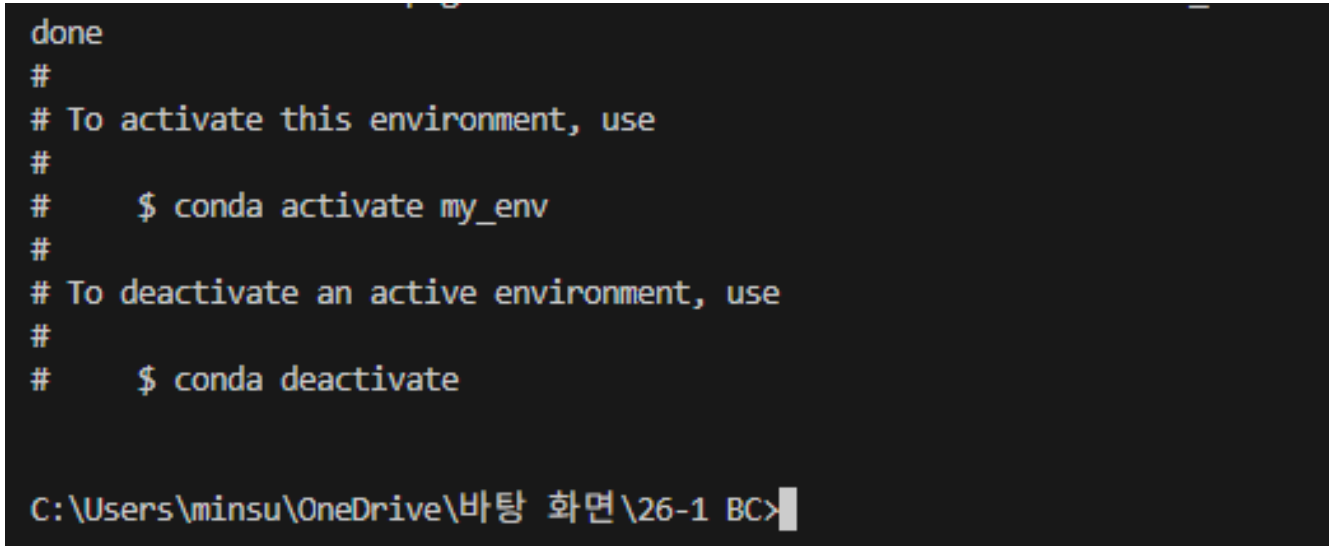


윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ Anaconda 활용 가상환경 설치

- Anaconda Prompt or Anaconda Navigator 활용 or VS Code 이용
- **프롬프트 / 터미널 이용 시: conda create –n [가상환경 이름] [python=[파이썬 버전], 선택]**



윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ Anaconda 활용 가상환경 설치

- Anaconda Prompt or Anaconda Navigator 활용 or VS Code 이용
- **프롬프트 / 터미널 이용 시: conda create –n [가상환경 이름] [python=[파이썬 버전], 선택]**

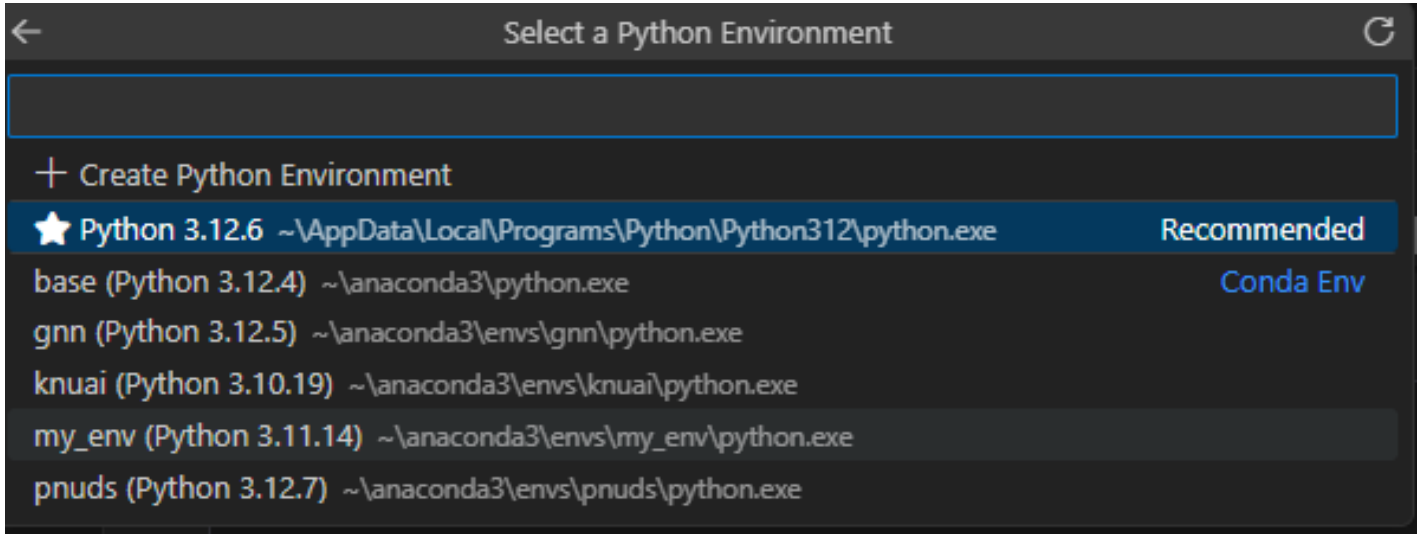
```
# To activate this environment, use
#
#     $ conda activate my_env
#
# To deactivate an active environment, use
#
#     $ conda deactivate


C:\Users\minsu\OneDrive\바탕 화면\26-1 BC>conda activate my_env

(my_env) C:\Users\minsu\OneDrive\바탕 화면\26-1 BC>
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Virtual Environment

▪ Anaconda 활용 가상환경 설치

- Anaconda Prompt or Anaconda Navigator 활용 or VS Code 이용
- **프롬프트 / 터미널 이용 시: conda create –n [가상환경 이름] [python=[파이썬 버전], 선택]**

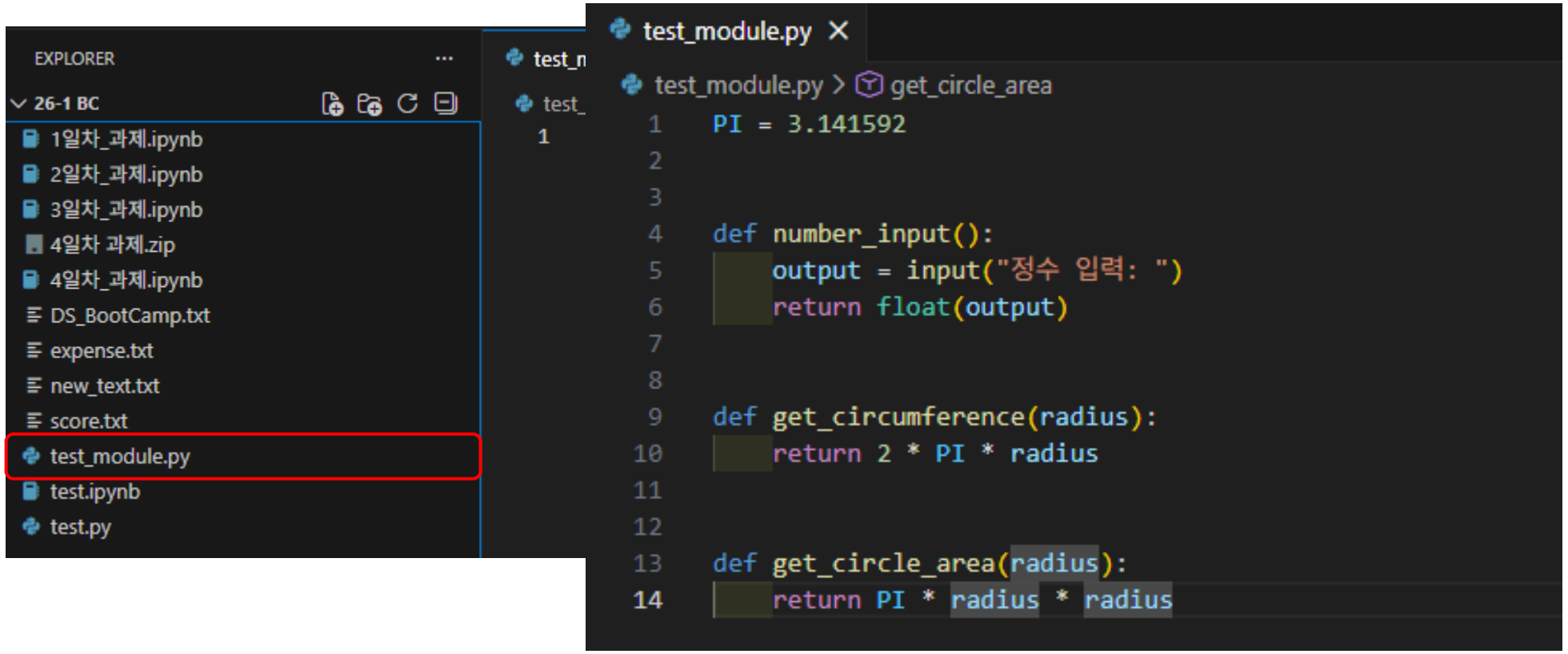


윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Package

▪ Module

- 사용자 정의 모듈 만들기 & 불러오기



윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Package

▪ Module

- 사용자 정의 모듈 만들기 & 불러오기

test_module.py | test.py

test.py > ...

```
1  import test_module as test
2
3  radius = test.number_input()
4  print(test.get_circumference(radius))
5  print(test.get_circle_area(radius))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER SPELL CHECKER Python

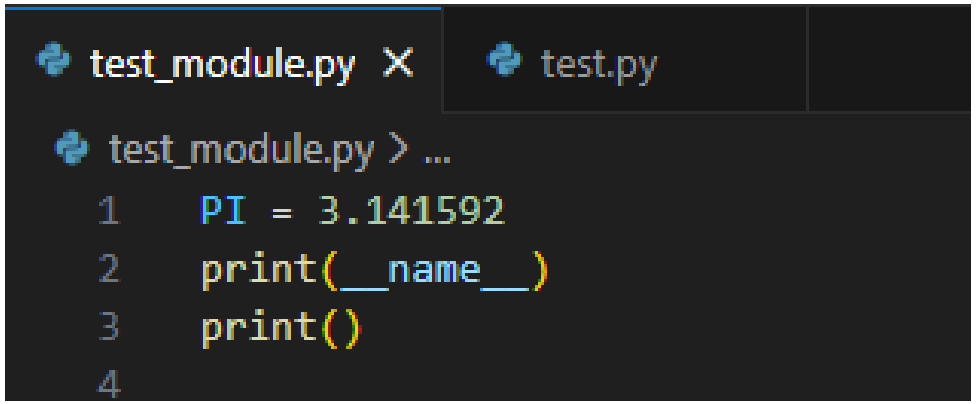
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 기능 및 개선 사항에 대 한 최신 PowerShell을 설치 하세요! https://aka.ms/PSWindows

개인 및 시스템 프로필을 로드하는 데 3295ms가 걸렸습니다.
(base) PS C:\Users\minsu\OneDrive\바탕 화면\26-1 BC> & C:/Users/minsu/AppData/Local/Programs/Python/Python312/python
.exe "c:/Users/minsu/OneDrive/바탕 화면/26-1 BC/test.py"
정수 입력: 5
31.41592
78.5398
(base) PS C:\Users\minsu\OneDrive\바탕 화면\26-1 BC>
```

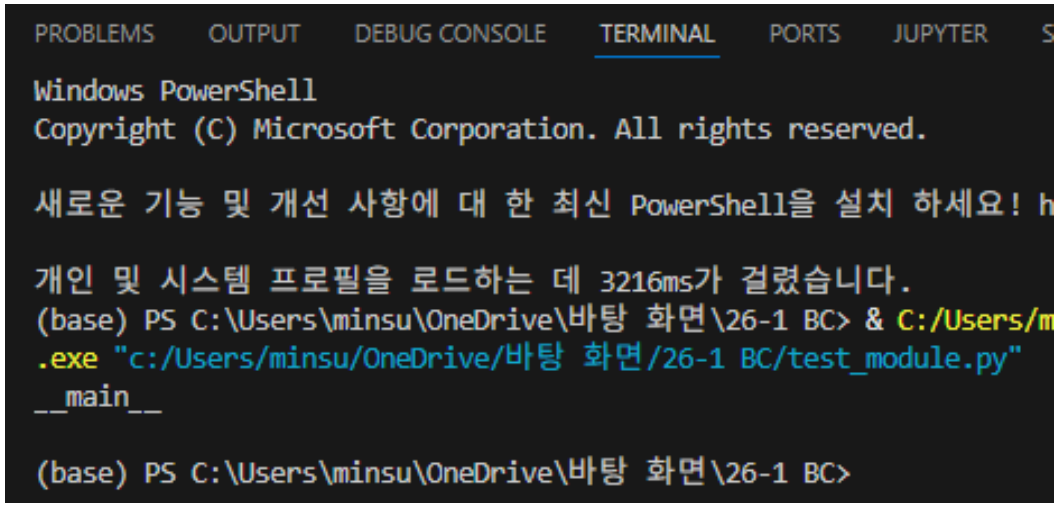
윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Package

▪ Module

- '_ _name_ _ == "_ _main_ _"' = 모듈 코드를 단독적으로 실행시키고 싶을 때 사용
- '_ _name_ _': 모듈 이름 저장한 변수 (자동 생성)
- .py를 작성 후 그 파일을 실행하면 '_ _name_ _ = "_ _main_ _"' 이다.





윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Package

▪ Module

- '_ _name_ _ == "_ _main_ _"' = 모듈 코드를 단독적으로 실행시키고 싶을 때 사용
- '_ _name_ _': 모듈 이름 저장한 변수 (자동 생성)
- .py를 작성 후 그 파일을 실행하면 '_ _name_ _ = "_ _main_ _"' 이다.

test_module.py

```python
PI = 3.141592
print(f"test_module.py 실행")
print(f"test_module.py의 __name__: {__name__}")
```

test.py

```python
import test_module

print(f"test.py 실행")
print(f"test.py의 __name__: {__name__}")
```

```
(base) PS C:\Users\minsu\OneDrive\바탕 화면\26-1 BC> & C:/Use
.exe "c:/Users/minsu/OneDrive/바탕 화면/26-1 BC/test.py"
test_module.py 실행
test_module.py의 __name__: test_module
test.py 실행
test.py의 __name__: __main__
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Package

▪ Module

- '_ _name_ _ == "_ _main_ _"' = 모듈 코드를 단독적으로 실행시키고 싶을 때 사용
- '_ _name_ _': 모듈 이름 저장한 변수 (자동 생성)
- .py를 작성 후 그 파일을 실행하면 '_ _name_ _ = "_ _main_ _"' 이다.

test_module.py

```python
PI = 3.141592


def number_input():
    output = input("정수 입력: ")
    return float(output)


def get_circumference(radius):
    return 2 * PI * radius


def get_circle_area(radius):
    return PI * radius * radius


if __name__ == "__main__":
    print(f"get_circumference(10): {get_circumference(10)})")
```

test.py

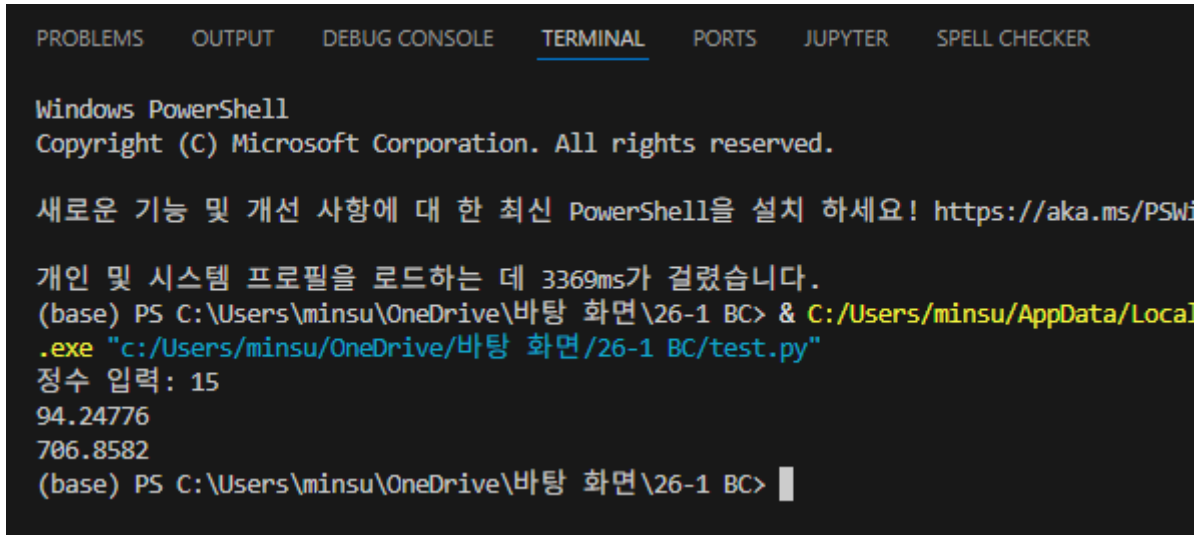
```python
import test_module as test

radius = test.number_input()
print(test.get_circumference(radius))
print(test.get_circle_area(radius))
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

Package

▪ Module

- '_ _name_ _ == "_ _main_ _"' = 모듈 코드를 단독적으로 실행시키고 싶을 때 사용
- '_ _name_ _': 모듈 이름 저장한 변수 (자동 생성)
- .py를 작성 후 그 파일을 실행하면 '_ _name_ _ = "_ _main_ _"' 이다.

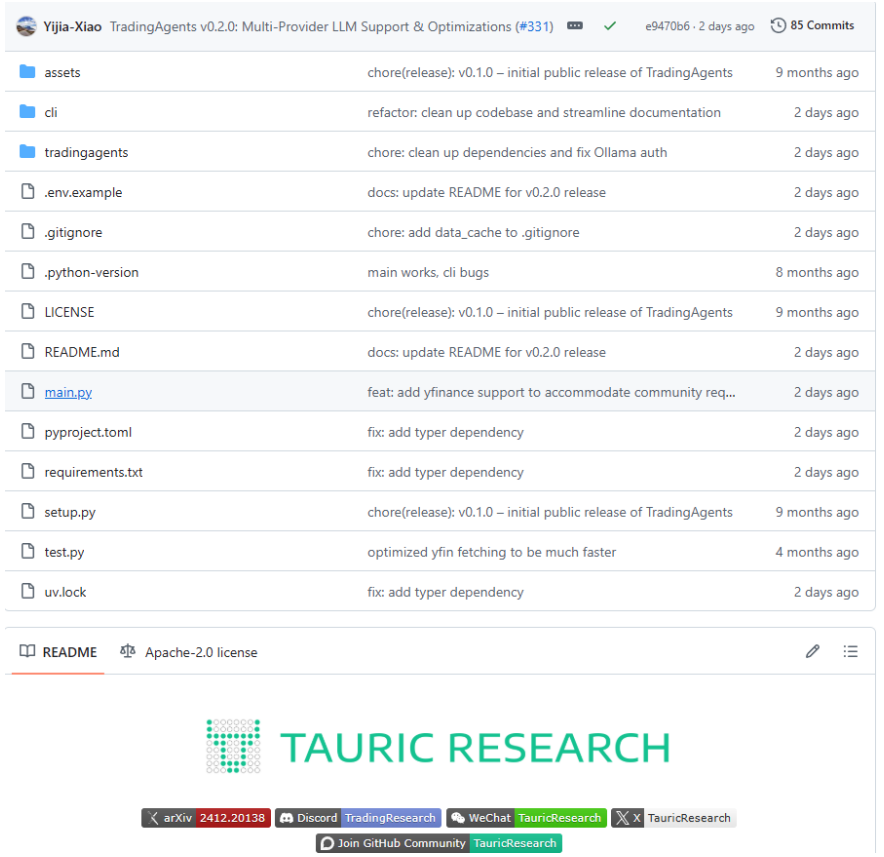


test.py 실행

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://dojang.io/mod/page/view.php?id=2470#google_vignette

▪ Module 실제 활용 예시

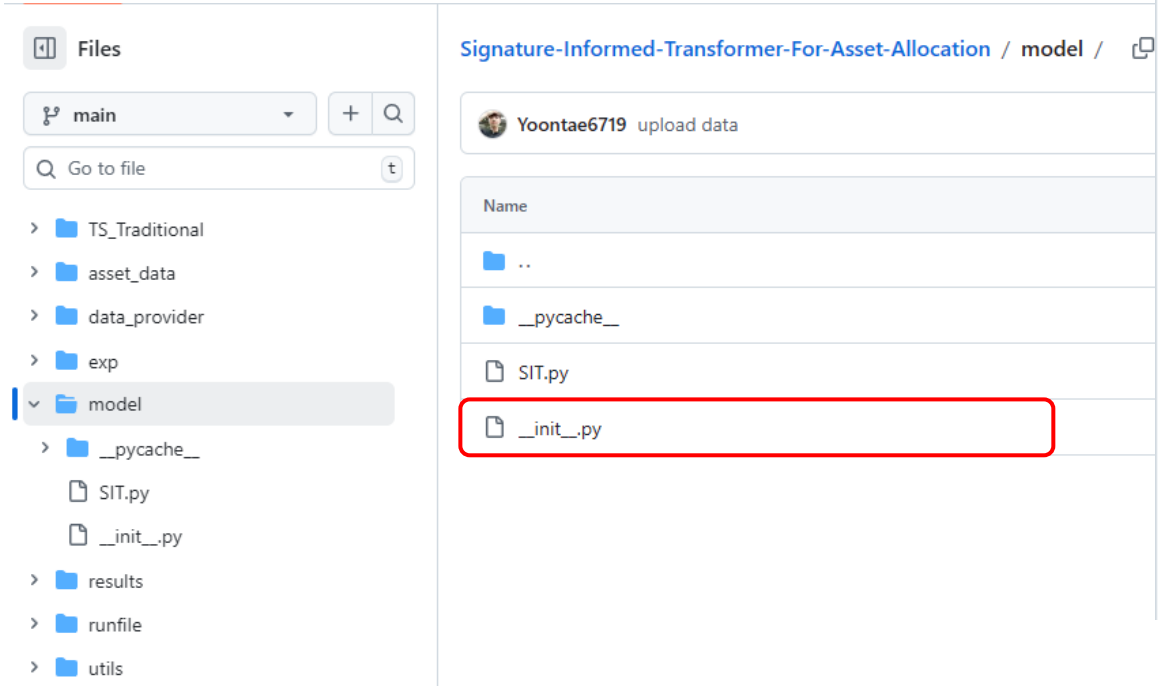
- https://github.com/tauricresearch/tradingagents

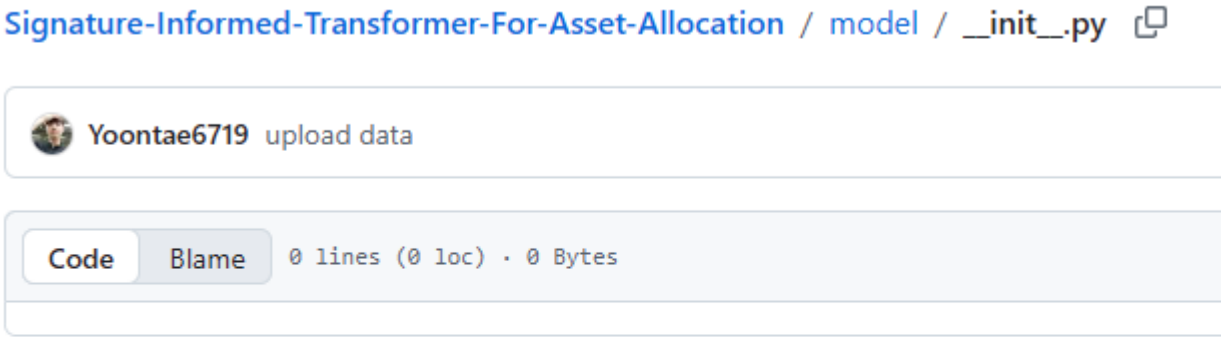


Package

▪ 패키지 (Package)

- 변수, 함수 ☞ 모듈 ☞ 패키지 (numpy, pandas, …)
- _ _init.py_ _: Module이나 Package와 관련된 초기화 파일 (없어도 됨)





윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 400-457쪽
https://github.com/Yoontae6719/Signature-Informed-Transformer-For-Asset-Allocation

Q&A