

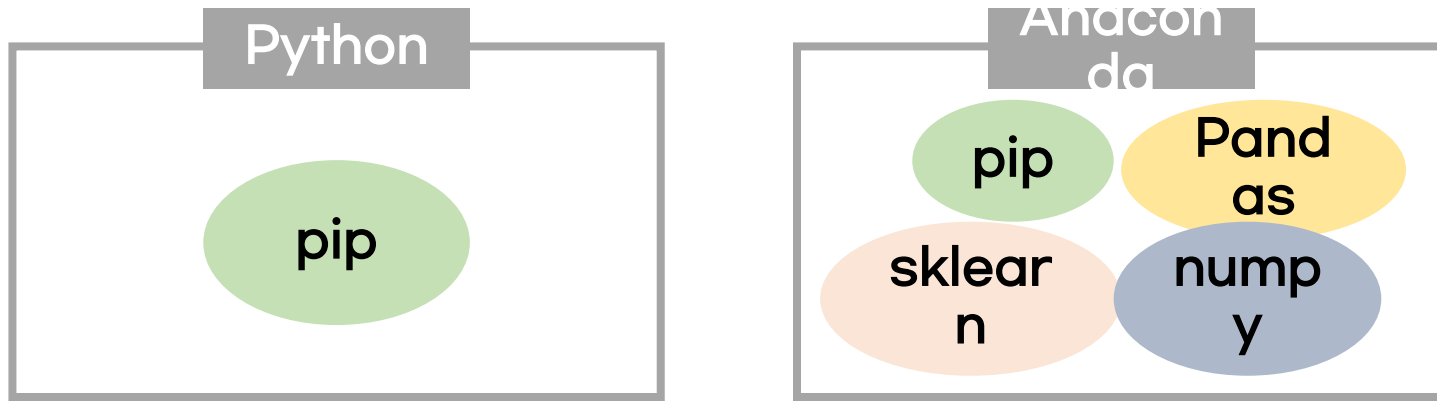
codingOn x posco

K-Digital Training

가상환경

Anaconda

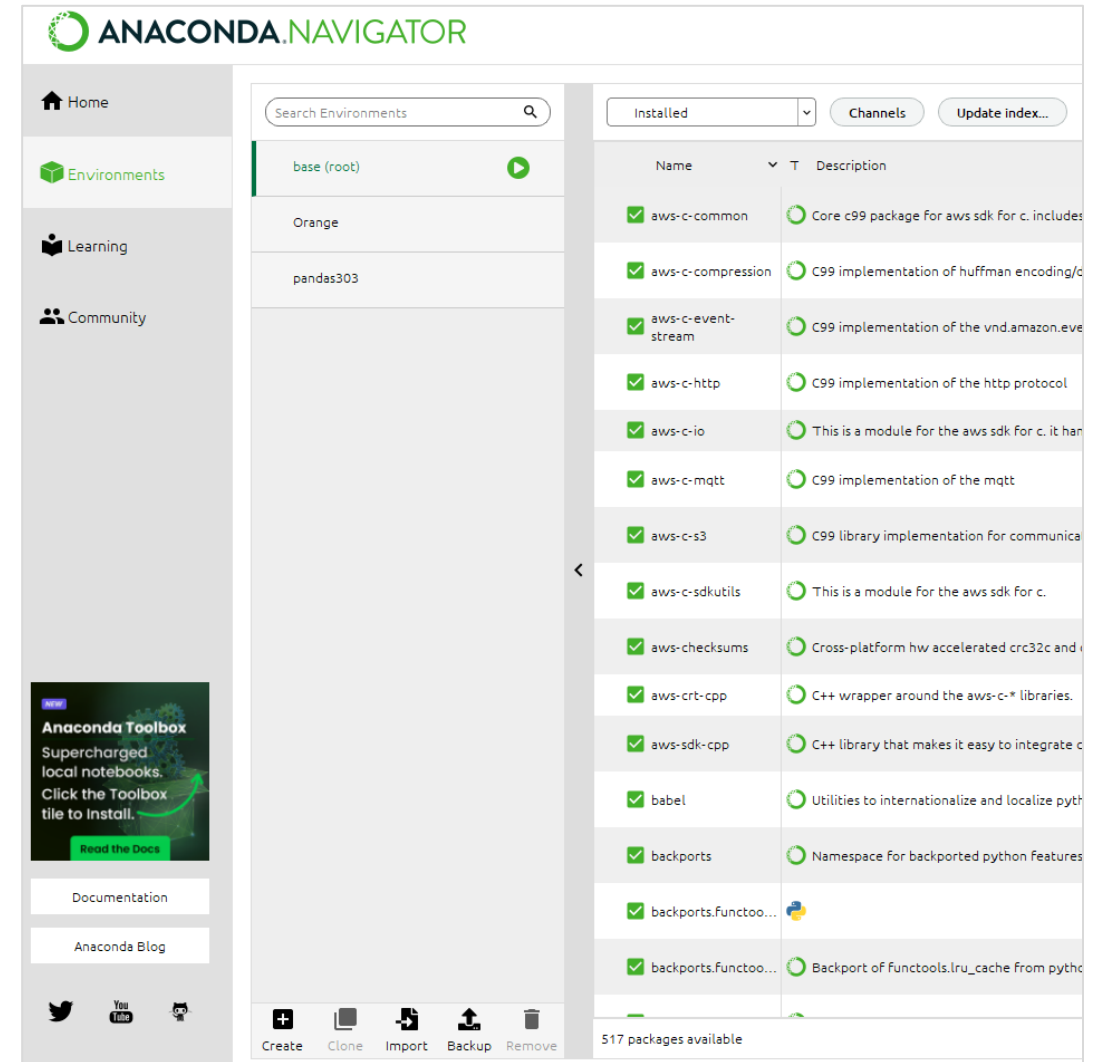
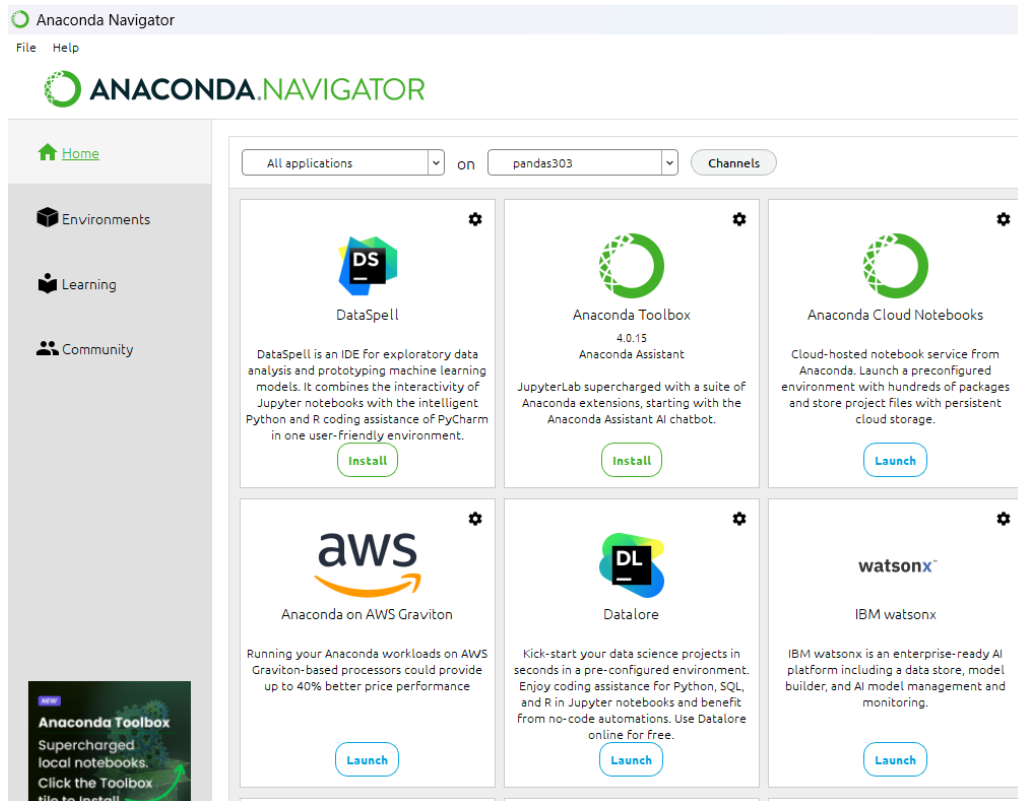
Python vs Anaconda



- Python은 기본적으로 패키지 관리 시스템인 pip만을 포함하고 있음
 - ➡ 필요한 툴, 패키지가 있다면 pip을 통해 수동으로 추가해야 함.
 - ➡ 패키지가 컴퓨터 자체에 설치됨
 - ➡ 프로젝트를 여러 번 진행하다 보면 필요한 패키지는 2~3개 정도면 되는데, 10개 15개의 패키지들이 설치되어 필요 이상으로 공간을 차지하기도 함.
- Anaconda는 데이터 분석, 머신 러닝 등에 사용하는 여러가지 패키지가 기본적으로 포함되어 있음

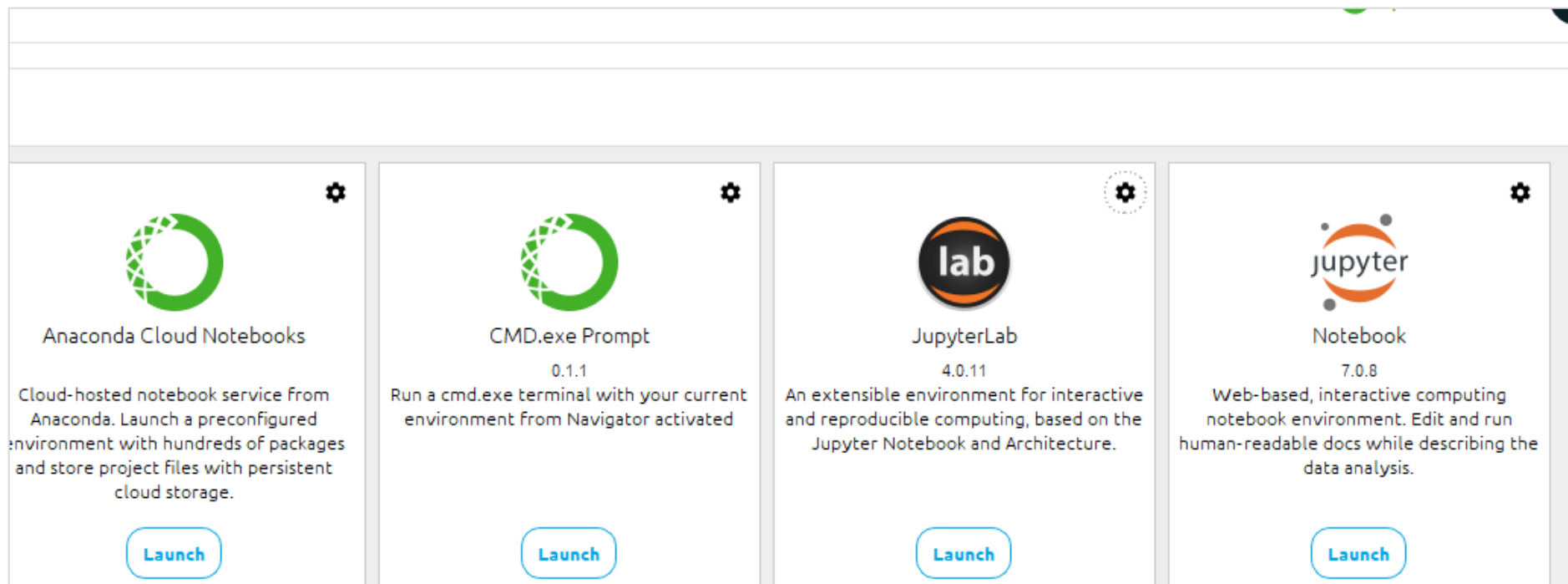
아나콘다 네비게이터 실행

- 아나콘다 라이브러리 설치 확인
Environments > base(root)



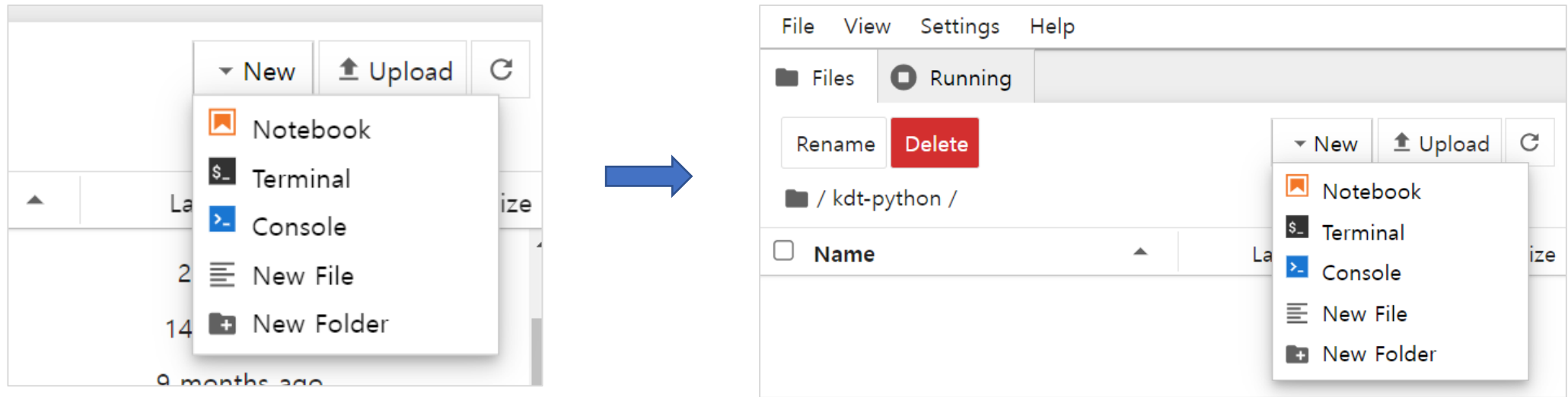
주피터 노트북 사용

▪ Jupyter Notebook > Launch



주피터 노트북 사용

- New > New Folder > 신규폴더 생성



- 신규폴더 > New > Notebook

주피터 노트북 사용

- 상단 > Untitled > basic.ipynb(파일)

```
# 변수
msg = "행운을 빌어요!!"
# print(msg)
msg

'행운을 빌어요!!'
```

- 단축키
 - 실행: ctrl + Enter
 - 실행후 다음 줄 생성: shift + Enter

주피터 노트북 사용

```
# 리스트  
cart = ['라면', '콩나물', '계란', '초코파이']  
cart  
cart[0]  
cart[-1]  
for i in cart:  
    print(i)
```

```
['라면', '콩나물', '계란', '초코파이']  
라면  
콩나물  
계란  
초코파이
```

주피터 노트북 사용

모듈 임포트하기

```
: import math
import random

v1 = math.ceil(12.56)
print(v1)
v2 = math.floor(12.56)
print(v2)

v3 = random.random()
print(v3)
v4 = random.randint(1, 10)
print(v4)

# 주사위 10번 던지기
for i in range(10):
    dice = random.randint(1, 6)
    print(dice)
```

주피터 노트북 사용

- basic2.ipynb

함수, 클래스 사용하기

```
def myabs(x):  
    if x < 0:  
        return -x  
    else:  
        return x
```

```
print(myabs(-2))  
print(myabs(2))
```

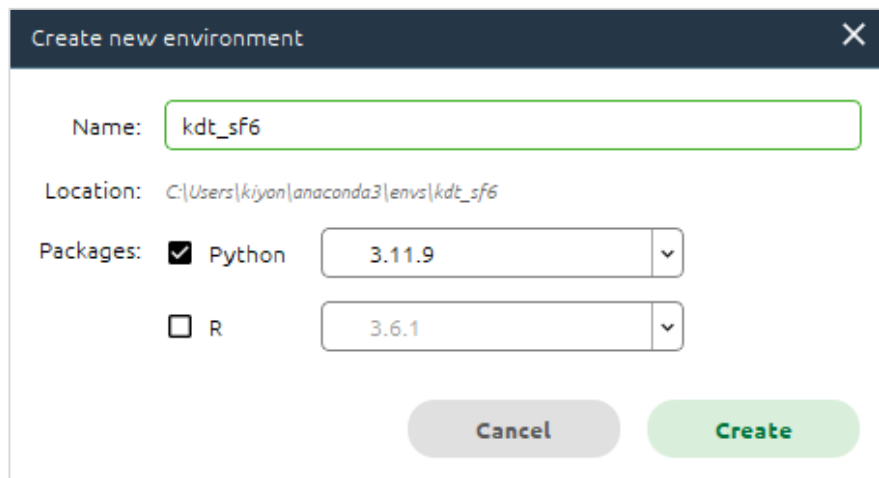
```
# 내장 abs()  
print(abs(-2))  
print(abs(2))
```

```
class Car:  
    def __init__(self, model_name, year):  
        self.model_name = model_name  
        self.year = year  
  
    def __str__(self):  
        return f'모델명: {self.model_name}, 연식: {self.year}'  
  
car1 = Car('K7', 2020)  
print(car1)  
car2 = Car('아이오닉5', 2023)  
print(car2)
```

아나콘다 가상환경

- 가상환경 생성

Environments > Create(하단 아이콘) > kdt_energy1(이름)

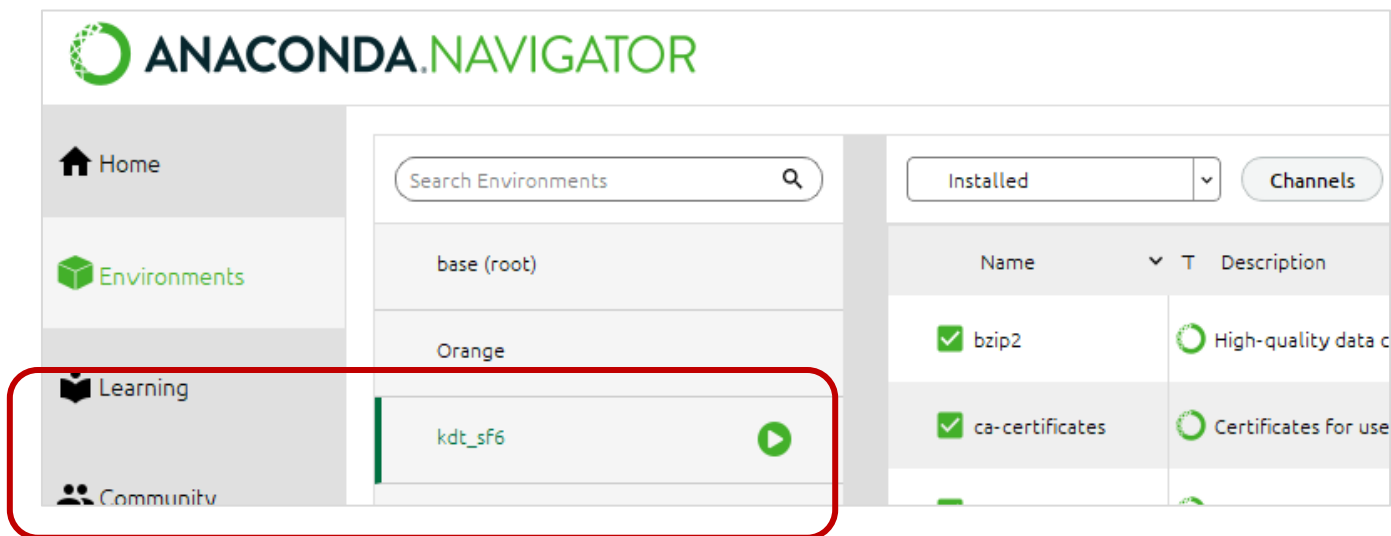


Create new environment

Name:

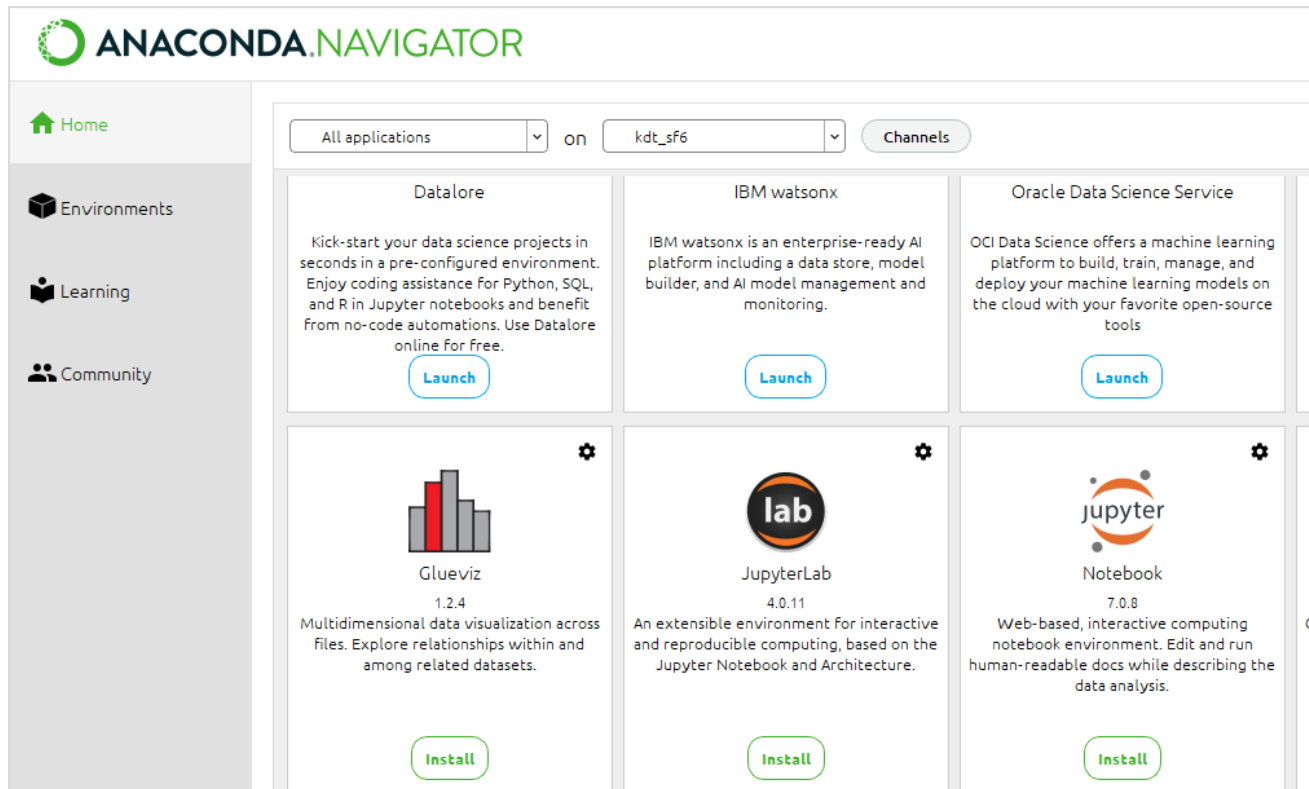
Location:

Packages: ☒ Python
☐ R



아나콘다 개발도구

- 개발도구(IDE) 실행
 - Home > 주피터 랩 > Install > Launch

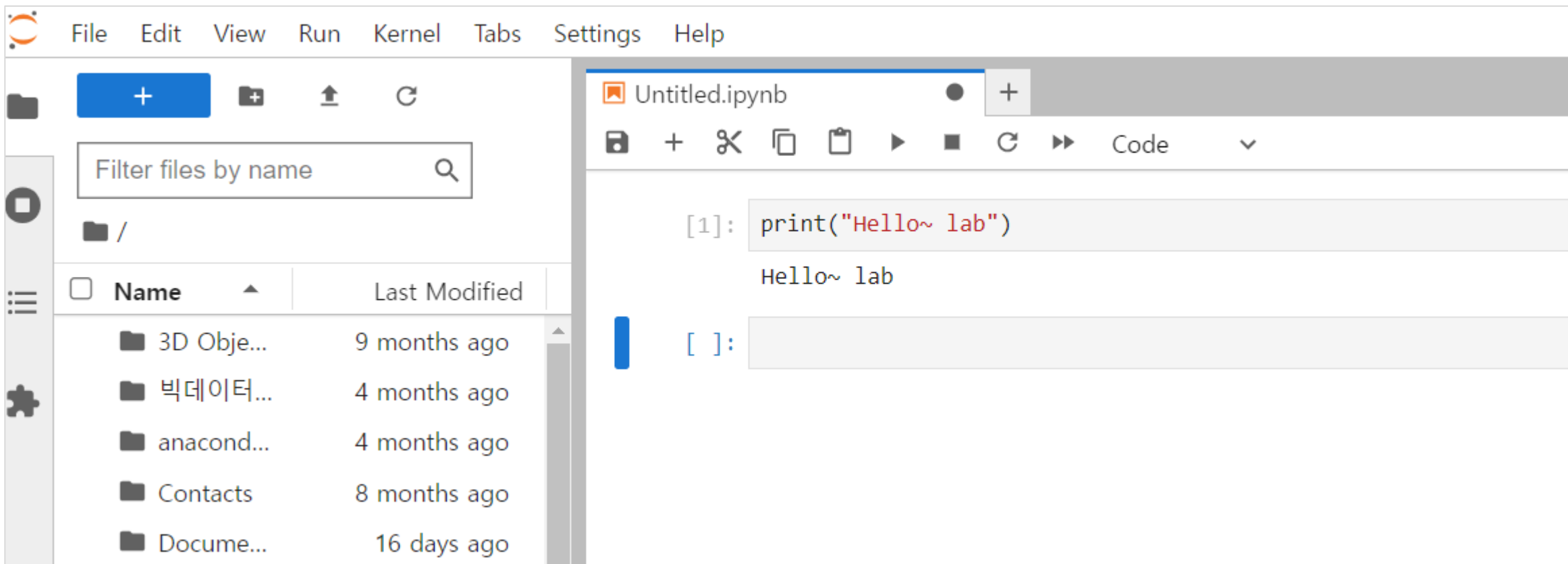


주피터 랩(Jupyter Lab)

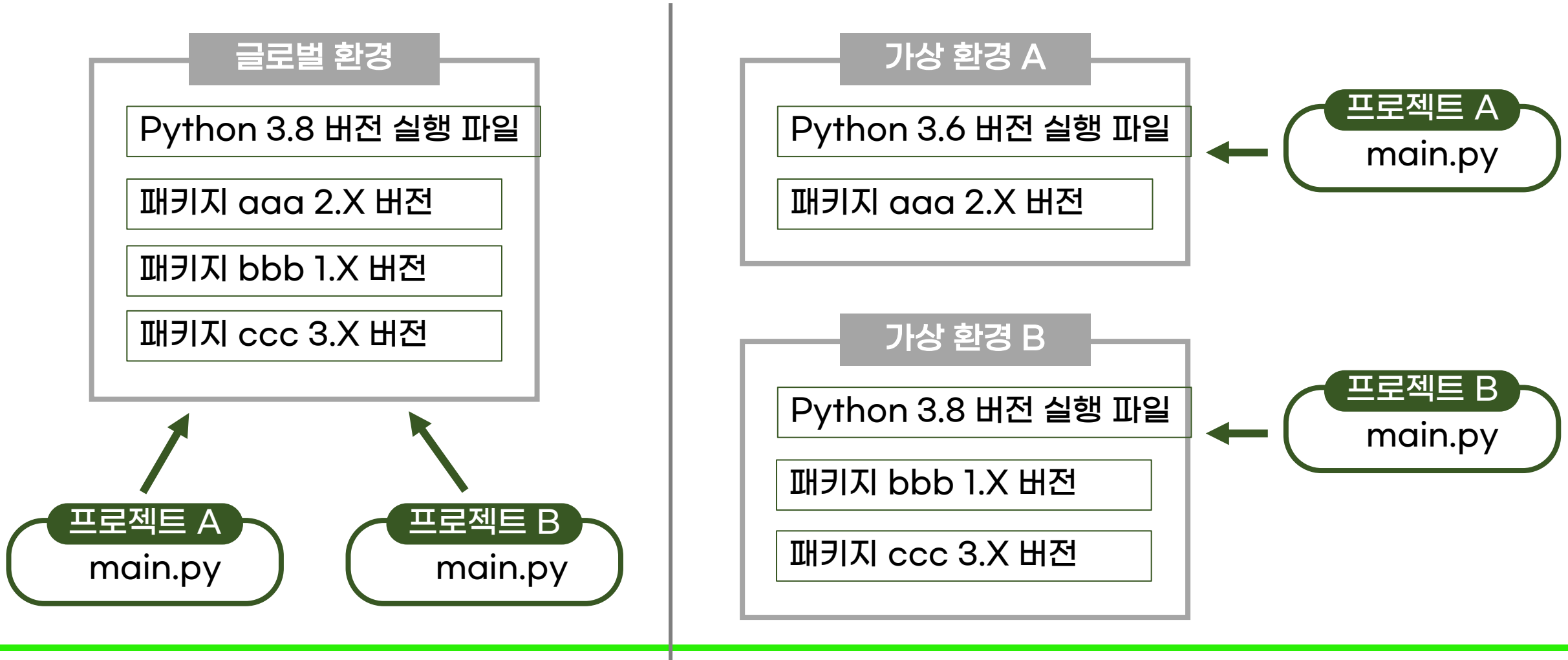
- ✓ Jupyter Lab은 2018년 출시되었고, Jupyter Notebook 보다 더 발전된 버전이다.
 - ✓ 주피터 랩은 대화형 컴퓨팅을 지원하여 코드를 실행하고 결과를 즉시 확인할 수 있다.
 - ✓ 다양한 플러그인과 확장 기능을 제공하여 사용자 정의 작업환경을 구성할 수 있다.
 - ✓ 데이터 분석 및 시각화 작업에 탁월하며 문서화 하여 저장하고 공유할 수 있다.
-

주피터 랩 실행

- Python3(ipykernel) 아이콘 클릭

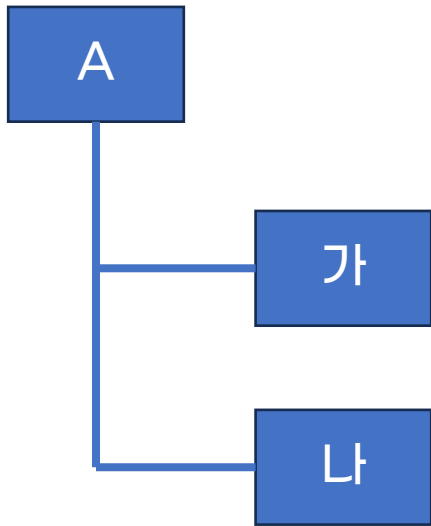


가상 환경 vs 글로벌 환경



가상 환경 vs 글로벌 환경

- 폴더구조



예) pip install 모듈

글로벌환경

- 가, 나 폴더에 모두 사용가능

가상환경

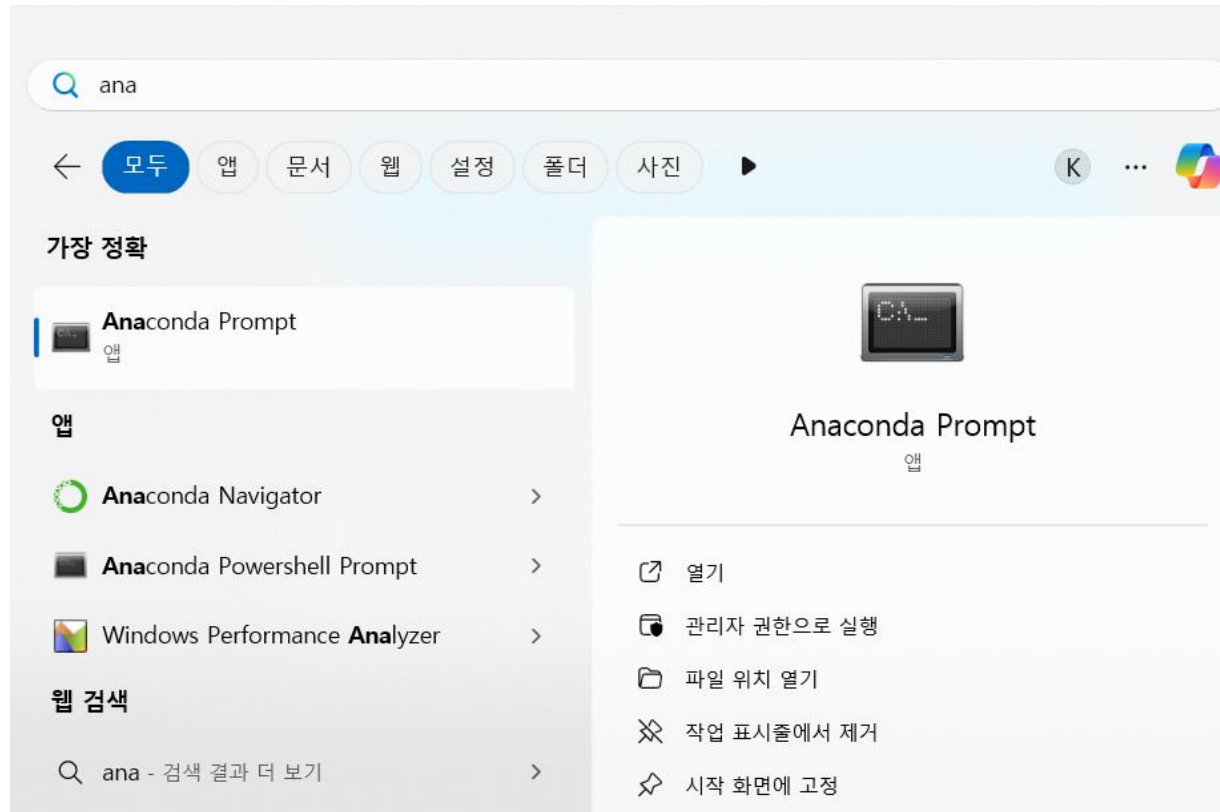
- 만약 가폴더에서만 가상환경을 생성하였다면
가폴더에서만 사용가능

가상환경이 필요한 이유?

- 프로젝트를 진행하다 보면 여러 library, package를 다운로드 하게 됨.
 - ➡ 각 library, package들끼리 **충돌**을 일으키는 경우 ↑
- 이런 경우에 가상환경을 이용한다면?
 - ➡ 프로젝트별로 **독립적인 작업 환경**에서 작업할 수 있다!

Anaconda 가상 환경 관리

- Anaconda Prompt 실행



Anaconda 가상 환경 관리

- 가상환경 조회 - conda env list

```
(base) C:\Users\shg02>conda env list
# conda environments:
#
base                * C:\Users\shg02\anaconda3
kdt_test            C:\Users\shg02\anaconda3\envs\kdt_test
```

Anaconda 가상 환경 관리

- 새로운 가상환경 생성

이미 navigator에서 생성했으므로 아래는 pass

가상환경 생성

```
> conda create -n [가상환경이름]  
또는 conda create --name [가상환경이름]
```

파이썬 버전을 지정하며 가상환경 생성하기

```
> conda create -n [가상환경이름] python=[0.0]  
또는 conda create --name [가상환경이름] python=[0.0]
```

Anaconda 가상 환경 관리

- 가상환경 활성화 - **생성한 폴더로 이동한 후**

1. `conda activate 폴더명`
2. `pip install ipykernel`
3. `python -m ipykernel install --user --name 이름`

2-주피터랩 또는 주피터 노트북과 같은 인터페이스에서 사용할 수 있는 커널

3-주피터랩 또는 주피터 노트북에서 작업할 때, 해당 가상환경을 선택할 수 있도록 추가

Jupyter lab 실행

- 주피터 랩 실행

- jupyter lab .(한 칸 띄고 점 찍음) -> 화면에서 생성한 이름 선택

```
(kdt_sf6) C:\Users\kiyon\KDT_SF6>jupyter lab .  
[W 2024-08-11 11:18:05.712 ServerApp] A `'_jupyter_server_extension_points'` function was found, a `'_jupyter_server_extension_paths'` function was found and will be used for now. This will be changed in future releases of Jupyter Server.  
[W 2024-08-11 11:18:05.731 ServerApp] A `'_jupyter_server_extension_points'` function was found, a `'_jupyter_server_extension_paths'` function was found and will be used for now. This will be changed in future releases of Jupyter Server.  
[I 2024-08-11 11:18:05.731 ServerApp] jupyter_lsp | extension was successfully linked.
```



Notebook



Python 3
(ipykernel)



kdt_test

Jupyter lab 실행

- 필수 라이브러리 설치 - 가상환경이 base(root)인 경우는 이미 설치됨
 - !pip install requests

```
[1]: !pip install requests
```

```
Requirement already satisfied: requests in c:\users\shg02\anaconda3\envs\kdt_test\lib\site-packages (2.32.3)  
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\shg02\anaconda3\envs\kdt_test\lib\site-packages (from requests) (3.3.2)  
Requirement already satisfied: idna<4,>=2.5 in c:\users\shg02\anaconda3\envs\kdt_test\lib\site-packages (from requests) (3.7)  
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\shg02\anaconda3\envs\kdt_test\lib\site-packages (from requests) (2.2.3)  
Requirement already satisfied: certifi>=2017.4.17 in c:\users\shg02\anaconda3\envs\kdt_test\lib\site-packages (from requests) (2024.8.30)
```


수고하셨습니다