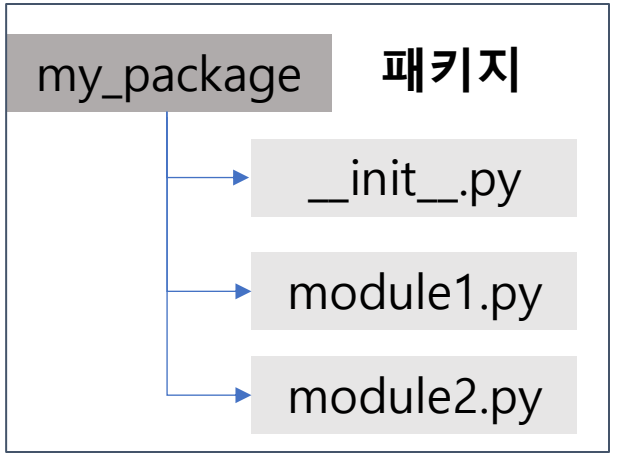
파이썬 라이브러리

Library

Library > Package > module

Library?

Library, Module, Package



- **Class** : 객체(Object)를 생성하는 틀(설계도)
- **Function** : : 특정 작업을 수행하는 코드 블록
- **Module** : method(function) + variable + class
  일반적으로 파일 1개로 구성 ex.) moduel1.py
- **Package** : 특정 기능과 관련된 **여러 모듈**이 모인 폴더(디렉토리)
- **Library** : 특정 기능을 수행하기 위해 미리 작성된 코드 모음 (**패키지 모음**)

Library > Package > module

```
1 import numpy as np
2 import pandas as pd
```

Library, Module, Package

- **Function, class : 기능 구현**

```python
def greet(name):
    return f"Hello, {name}!"

# 함수 호출
print(greet("Alice"))  # 출력: Hello, Alice!
```

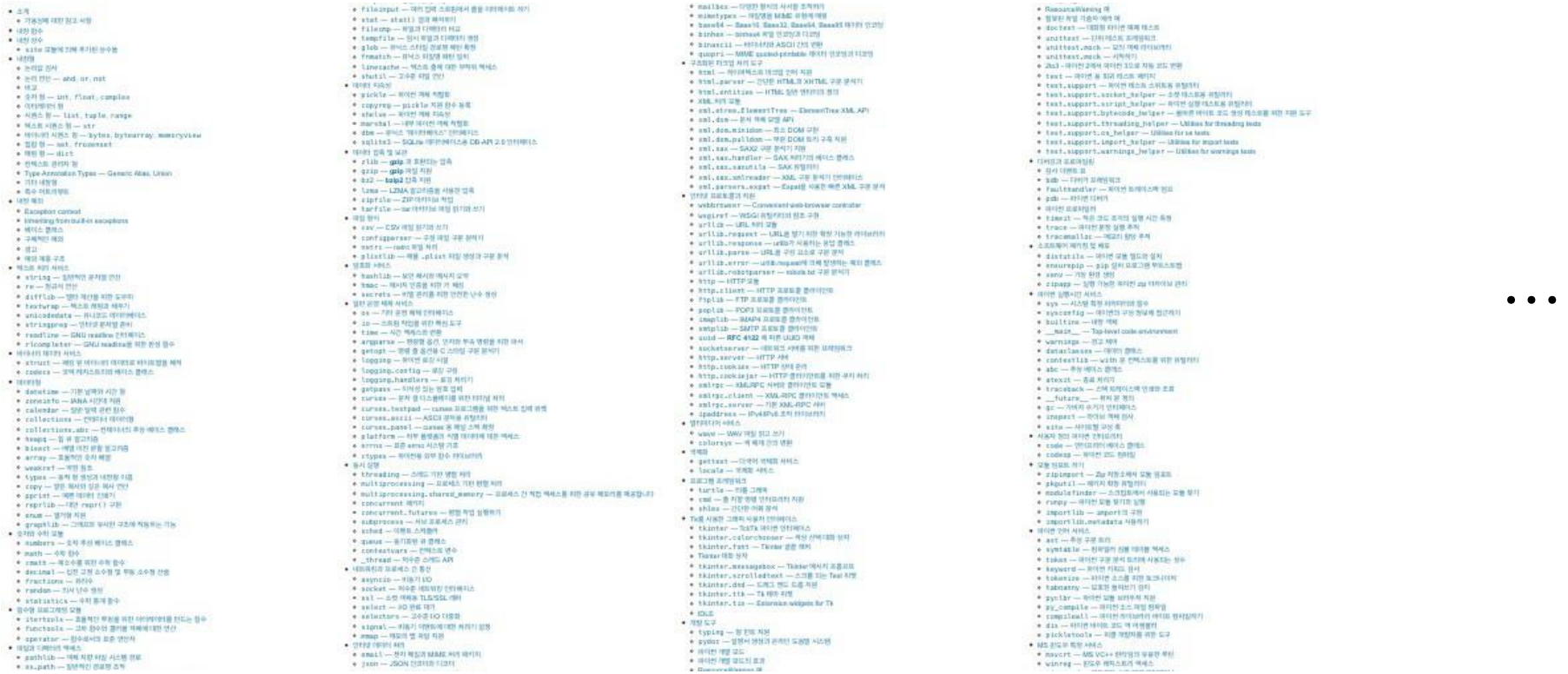
```python
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def greet(self):
        return f"Hello, my name is {self.name} and I am {self.age} years old."

# 클래스 인스턴스 생성
person1 = Person("Alice", 30)

# 메서드 호출
print(person1.greet())  # 출력: Hello, my name is Alice and I am 30 years old.
```

Standard Library

Python 과 함께 배포되는 표준 라이브러리

Standard Library

- 파이썬과 함께 배포되는 표준 라이브러리
- Python 공식 문서 설명 : https://docs.python.org/ko/3/library/index.html



...

Standard Library

1. 운영 체제 인터페이스 – os 모듈

- 운영 체제와 상호작용할 수 있는 다양한 기능 제공
- 파일 시스템 작업(파일 및 디렉터리 생성, 삭제, 이동 등)과 환경 변수 작업 등 수행

```
import os

# 현재 작업 디렉터리 출력
current_directory = os.getcwd()
print("Current Directory:", current_directory)

# 새로운 디렉터리 생성
os.mkdir("new_folder")
print("Created 'new_folder'")

# 디렉터리 존재 여부 확인
print("Does 'new_folder' exist?", os.path.exists("new_folder"))

# 디렉터리 이름 변경
os.rename("new_folder", "renamed_folder")
print("Renamed 'new_folder' to 'renamed_folder'")

# 디렉터리 삭제
os.rmdir("renamed_folder")
print("Deleted 'renamed_folder'")
```

Standard Library

2. 파일 탐색 - glob

- 파일 경로와 일치하는 모든 파일 이름을 찾는 데 사용
- 파일 이름의 패턴 매칭 수행, 와일드카드(*, ?, [])를 사용하여 특정 패턴에 맞는 파일 서치

```
import glob

# 현재 디렉터리에서 .py 파일을 모두 찾기
python_files = glob.glob("*.py")
print("Python files in the current directory:", python_files)
```

Standard Library

3. 수학 모듈 - math

- 수학적인 계산에 필요한 다양한 함수와 상수 제공
- math 모듈을 사용하여 기본적인 산술 연산부터 고급 수학 함수(삼각 함수, 로그 함수 등)까지 다양한 수학적 작업 수행

```
import math

# 수학 상수
print("PI:", math.pi)
print("Euler's number (e):", math.e)

# 기본 수학 함수
print("Square root of 16:", math.sqrt(16))
print("3 raised to the power of 4:", math.pow(3, 4))
print("Factorial of 5:", math.factorial(5))
print("GCD of 60 and 48:", math.gcd(60, 48))

# 삼각 함수
angle_rad = math.pi / 4  # 45도(라디안)
print("sin(45 degrees):", math.sin(angle_rad))
print("cos(45 degrees):", math.cos(angle_rad))

# 로그 함수
print("Natural log of 10:", math.log(10))
print("Log base 10 of 100:", math.log10(100))

# 올림/내림 함수
print("Ceiling of 4.3:", math.ceil(4.3))
print("Floor of 4.7:", math.floor(4.7))
```

Standard Library

4. 인터넷 액세스 – request

- **HTTP 요청을 보낼 수 있게 해주는 라이브러리**
- **GET, POST 등의 HTTP 메서드를 사용하여 웹 서버와 상호작용**

```
# !pip install requests

import requests

# GET 요청 사용, 웹페이지 내용 가져오기
response = requests.get("https://api.github.com")

# 상태 코드 출력
print("Status Code:", response.status_code)

# 응답 내용 출력 (텍스트 형태)
print("Response Text:", response.text)

# JSON 데이터로 변환하여 출력
data = response.json()
print("JSON Response:", data)
```

Standard Library

5. 날짜와 시간 – datetime

- 날짜와 시간을 처리하는 데 사용되는 표준 라이브러리
- 현재 날짜와 시간, 특정 날짜 및 시간 계산, 날짜 형식 변환 등 수행

```
# 날짜 계산 (100일 후의 날짜)
future_date = today + datetime.timedelta(days=100)
print("Date 100 days from today:", future_date)

# 문자열을 날짜로 변환
date_str = "2024-08-25"
date_obj = datetime.datetime.strptime(date_str, "%Y-%m-%d")
print("Converted Date from String:", date_obj)

# 날짜를 문자열로 변환
date_to_str = now.strftime("%Y-%m-%d %H:%M:%S")
print("Formatted Date to String:", date_to_str)
```

Standard Library

5. 날짜와 시간 – datetime

- 날짜와 시간을 처리하는 데 사용되는 표준 라이브러리
- 현재 날짜와 시간, 특정 날짜 및 시간 계산, 날짜 형식 변환 등 수행

```
import datetime

# 현재 날짜와 시간 가져오기
now = datetime.datetime.now()
print("Current Date and Time:", now)

# 특정 날짜와 시간 생성
specific_date = datetime.datetime(2023, 12, 25, 10, 30, 0)
print("Specific Date and Time:", specific_date)

# 날짜 부분만 가져오기
today = datetime.date.today()
print("Today's Date:", today)
```

Standard Library

- 실습

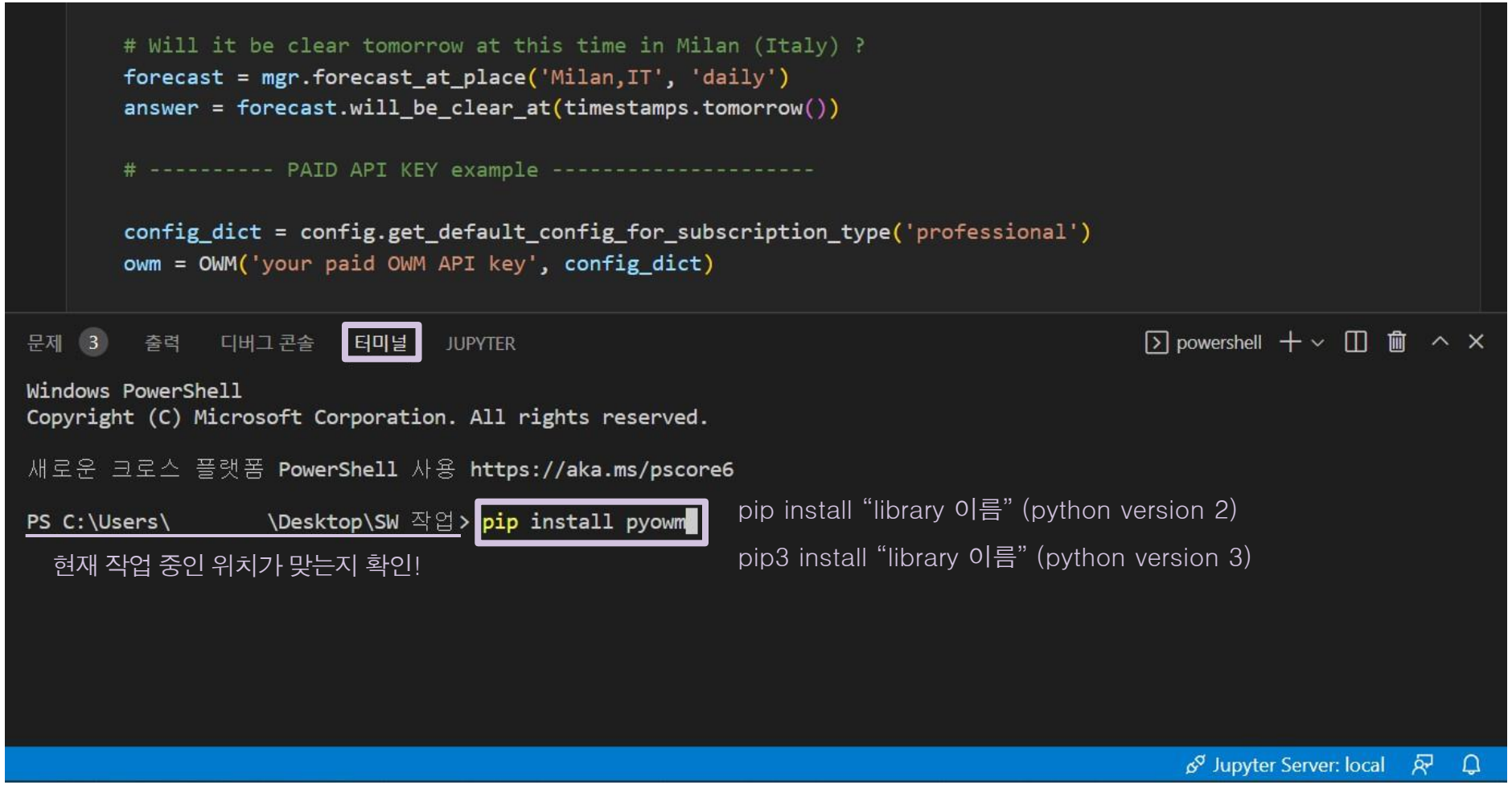
  10.1.3.StandardLibrary.ipynb

기타 유용한 Library

- 데이터 처리 : Pandas, Numpy, NetworkX
- 고속화 : Numba, Cython
- 딥러닝 : Pytorch, Tensorflow, Keras, Caffe, Theano, ScikitLearn, Scipy, Xgboost
- 이미지 및 영상 : Pillow, OpenCV, MoviePy
- 시각화 : Matplotlib, Plotly, Seaborn, Bokeh, ggplot2
- 크롤링 : Beautifulsoup, Selenium, Scrapy
- 백엔드, 웹 등 : Django, Requests, Bottle
- 게임 : PyGame
- Gui 등 : Tkinter, PyQt, Pyinstaller

How to install library

1. Visual Studio code



How to install library

2. colab

```
1  !pip install numpy
```

!로 시작 : command line interface (CLI)

Standard Library

- 파이썬 표준 라이브러리
  https://docs.python.org/ko/3/library/index.html

THANK YOU