

# 모듈(module)과 패키지(package)

이번 장에서는 파이썬에서 모듈의 import 방법, 모듈의 함수 사용, 패키지 (외부모듈)의 설치에 대하여 공부합니다.



1. 모듈과 import
2. import 문법
3. 모듈의 함수 호출하기
4. sys 모듈
5. time 모듈
6. random 모듈
7. statistics 모듈
8. turtle 모듈
9. 패키지
10. 패키지 설치하기
11. 패키지 설치와 사용하기 예



# 1. 모듈과 import

## □ 모듈 (module)

- 모듈은 파이썬 코드를 작성해 놓은 스크립트 파일
- 모듈 안에 함수, 변수, 클래스 등이 정의되어 있음
- 파이썬 설치 디렉토리를 보면, 확장자가 py인 파일이 수십여개 있으며, 여기에 파이썬에서 제공하는 표준 모듈들이 있음
- 표준 모듈의 집합을 '표준 라이브러리(standard library)' 라고 부름  
( <https://docs.python.org/3/library/index.html> )

## □ import 명령

- 모듈을 사용하려면 import 명령으로 모듈 이름을 지정함
- 현재 파일에 import된 모듈의 코드가 읽혀짐





## 2. import 문법

### □ import 사용법

- `import` 모듈명
- `import` 모듈명 `as` 별명
- `from` 모듈명 `import` 함수명 목록 [또는 \*]
- `from` 모듈명 `import` 함수명 `as` 별명

```
import math                # math 모듈을 import
import math as m           # math 모듈을 import하되 m 으로 지정

from math import sqrt      # math 모듈의 sqrt 함수만 import
from math import sqrt,pow  # math 모듈의 sqrt, pow 함수를 import
from math import *         # math 모듈의 모든 함수를 import (비권장)

from math import sqrt as sq # math 모듈의 sqrt 함수를 import하되
                             # sq로 지정
```



### 3. 모듈의 함수 사용하기

#### ❑ math 모듈의 함수 호출 예

```
import math
```

```
print(math.sqrt(4))
```

```
import math as m
```

```
print(m.sqrt(4))
```





### 3. 모듈의 함수 사용하기

#### ❑ math 모듈의 함수 호출 예

```
from math import sqrt
```

```
print(sqrt(4))
```

```
from math import sqrt, pow
```

```
print(sqrt(4), pow(2,3))
```

```
from math import sqrt as sq, pow as pp
```

```
print(sq(4), pp(2,3))
```



### 3. 모듈의 함수 사용하기

#### ❑ 모듈에 정의되어 있는 함수, 변수 알아보기

- 파이썬 내장함수 dir()

#### ❑ 모듈에 정의되어 있는 함수 도움말 보기

- 파이썬 내장함수 help()

```
import math
```

```
print(dir(math))          # math 모듈에 정의된 함수, 변수 등을 출력
```

```
help(math.pow)           # math.pow 함수의 도움말 출력
```



## 4. sys 모듈

### □ sys 모듈

- 컴퓨터 시스템 정보, 파이썬이 실행되는 환경 정보 등을 조회하고 관리

```
import sys

print("version = ", sys.version)           # 파이썬 버전
Print(" platform = ", sys.platform)        # OS (윈도우의 경우 win32 출력)
print("path = ", sys.path)                 # 모듈 검색을 위한 폴더 값
```







### □ time 모듈

- 날짜와 시간 관련 기능 제공

```
import time

t = time.time()                # 1970년 이후 지나온 시간을 초로 출력
                                # 소수점은 나노 초(nano second) 까지

print(t)
print(time.ctime(t))           # 년도, 월, 일, 요일, 시간으로 변환 출력
print(time.localtime(t))       # struct_time 형식으로 변환 출력

start = time.time()
for a in range(100) :
    print(a, end=', ')
print()
end = time.time()
print(end - start)
```



## 6. random 모듈

### □ random 모듈

- 난수 생성, 무작위 선택 등 제공

```
import random

for i in range(5) :
    print(random.random())      # 난수 생성

for i in range(5) :
    print(random.randint(1, 10)) # 1 ~ 10 중의 정수 난수를 생성

for i in range(5) :
    print(random.uniform(1, 10)) # 1 ~ 10 중의 실수 난수를 생성

food = ["짜장면", "짬뽕", "탕수육", "군만두"]
print(random.choice(food))      # food 리스트에서 하나를 무작위 선택
print(random.sample(food, 2))   # food 리스트에서 2개를 선택
random.shuffle(food)            # food 리스트의 요소를 무작위 섞음
print(food)
```



### □ statistics 모듈

- 평균, 분산 등 통계 함수 제공

```
import statistics

score = [30, 40, 50, 60, 70, 90]
print(statistics.mean(score))
print(statistics.median(score))
print(statistics.stdev(score))
print(statistics.variance(score))
```





### □ turtle 모듈

- 거북이를 이동시켜 도형을 그리는 방식으로 간단한 그래픽 기능 제공

```
import turtle as t

t.shape("turtle")           # 이동체의 모양 지정 (turtle, circle 등)

t.right(60)                 # 오른쪽으로 60도 방향 설정
t.forward(100)              # 100 픽셀의 거리를 이동
t.right(120)                # 오른쪽으로 120도 방향 설정
t.forward(100)
t.right(120)
t.forward(100)
t.done()
```



## 8. turtle 모듈

### ❑ turtle 예

```
import turtle as t

t.shape("turtle")

for a in range(5) :
    t.forward(150)
    t.right(144)
t.done()
```



## 8. turtle 모듈

### ❑ turtle 예

```
import turtle as t

t.shape("turtle")

t.pensize(3)
t.color("blue")
t.bgcolor("green")
t.fillcolor("yellow")
t.begin_fill()
t.circle(100)
t.end_fill()
t.done()
```



## 9. 패키지

### □ 패키지 (package)

- 모듈의 모음

### □ 외부 모듈(third-party module)

- 파이썬이 제공하는 것이 아니고, 제3자가 제작하여 배포하는 모듈
- 보통, 패키지로 묶어서 제공
- 파이썬 설치할 때 같이 설치되는 pip 프로그램으로 외부 패키지를 설치할 수 있음

### □ 파이썬 패키지 인덱스 (pyPI)

- 파이썬 패키지들의 저장소(repository)
- <https://pypi.org>



## 10. 패키지 설치하기 : pip 프로그램

### 1. cmd를 실행한 후, pip 명령어 입력

`pip list` : 설치된 패키지 목록을 출력함

이때 pip의 upgrade가 필요하다는 경고(warning)이 나오면, upgrade해야함

`python -m pip install --upgrade pip`

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\MJU>pip list
Package            Version
-----
astroid             2.4.2
colorama            0.4.3
isort               4.3.21
lazy-object-proxy   1.4.3
mccabe              0.6.1
pip                 20.1.1
pylint              2.5.3
setuptools          47.1.0
six                 1.15.0
toml                0.10.1
wrapit              1.12.1
WARNING: You are using pip version 20.1.1; however, version 20.2.3 is available.
You should consider upgrading via the 'c:\users\mju\appdata\local\programs\python\python38-32\python.exe -m pip install
--upgrade pip' command.
```





## 10. 패키지 설치하기 : pip 프로그램

### 2. pip의 upgrade

```
python -m pip install --upgrade pip
```

```
C:\Users\MJU>python -m pip install --upgrade pip
Collecting pip
  Downloading pip-20.2.3-py2.py3-none-any.whl (1.5 MB)
    |████████████████████████████████████████| 1.5 MB 1.1 MB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.1.1
    Uninstalling pip-20.1.1:
      Successfully uninstalled pip-20.1.1
Successfully installed pip-20.2.3
```



## 10. 패키지 설치하기 : pip 프로그램

- ❑ 패키지 설치하기  
pip install [패키지이름]

예) pip install numpy

: 패키지 numpy를 설치함

- ❑ 패키지 삭제하기  
pip uninstall [패키지이름]

- ❑ 설치된 패키지 목록 보기  
pip freeze  
pip list

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18363.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\PC1>pip install numpy
Collecting numpy
  Downloading https://files.pythonhosted.org/packages/df/cb/7af5de0a4068ce314e13074a4f86afada7fa86a095489bbc80f5dabd4156/numpy-1.19.1-cp37-cp37m-win_amd64.whl (12.9MB)
    100% |#####| 12.9MB 2.7MB/s
Installing collected packages: numpy
Successfully installed numpy-1.19.1
You are using pip version 19.0.3, however version 20.2b1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```



## 11. 패키지 설치와 사용하기 예

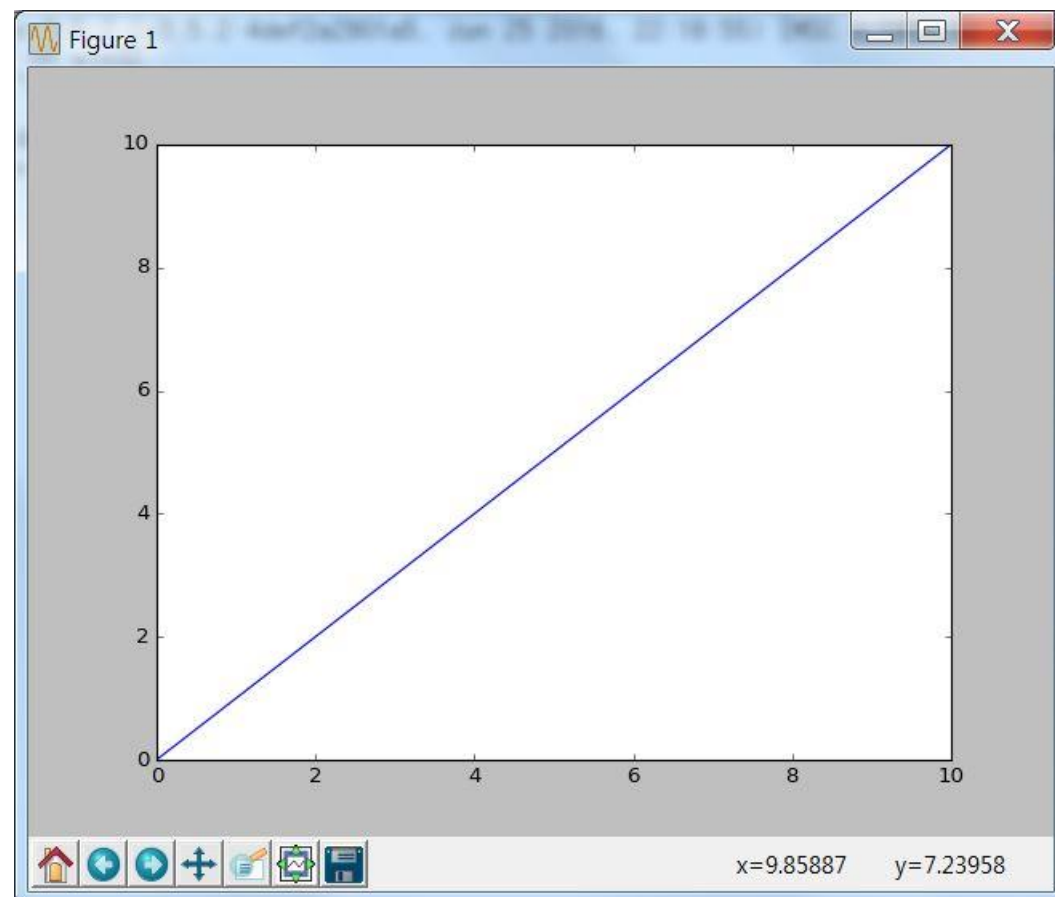
- ◆ 패키지 matplotlib 설치와 사용  
pip install matplotlib

### 코드 예

```
import matplotlib.pyplot as plt

# 0~10까지 x, y에 대입
x=y=[i for i in range(0, 11)]

plt.plot(x, y)
plt.show()
```





## 11. 패키지 설치와 사용하기 예

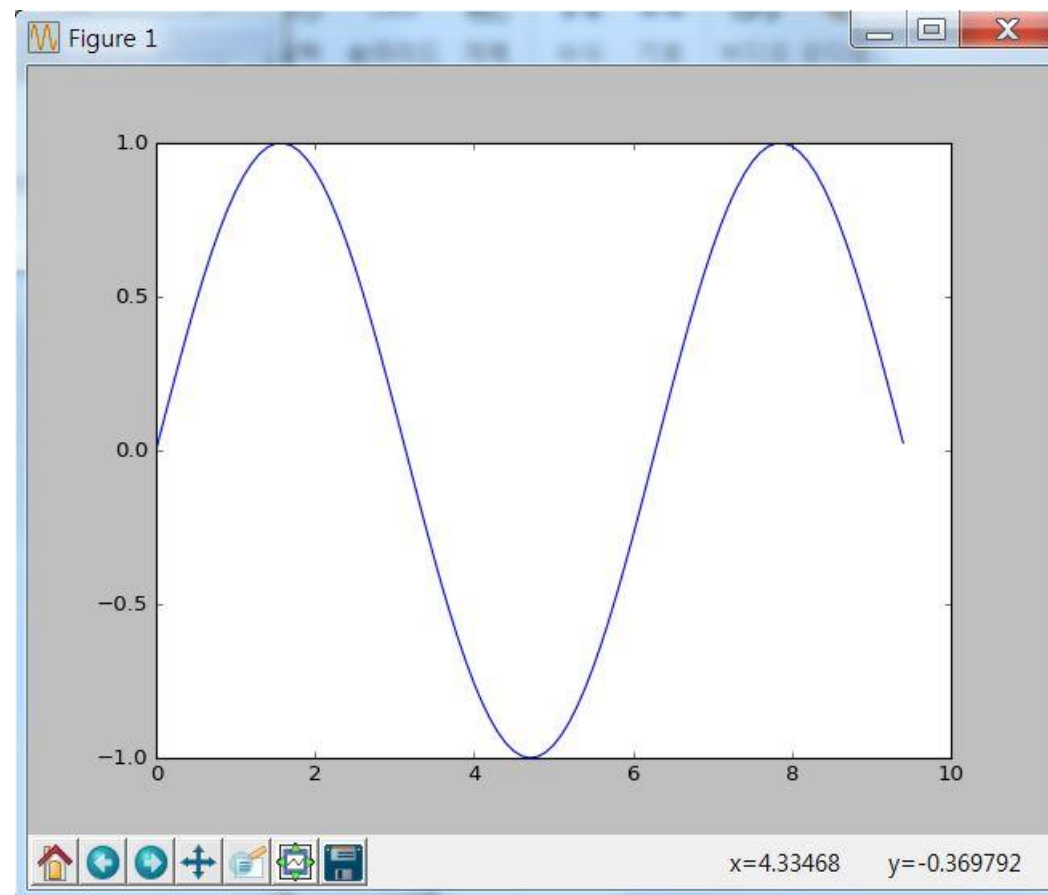
### ◆ 패키지 numpy 설치와 사용 pip install numpy

#### 코드 예

```
import numpy as np
import matplotlib.pyplot as plt

# 사인곡선에 해당하는 x와 y 좌표를 계산
x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)

# matplotlib를 사용해 점들을 표시
plt.plot(x, y)
plt.show()
```





### ◆ 패키지 BeautifulSoup 설치와 사용 pip install beautifulsoup4

```
from urllib import request
import bs4

target = request.urlopen("http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp?stnId=108")
soup = bs4.BeautifulSoup(target, "html.parser")

for city in soup.select("location") :
    name = city.select_one("city").string
    wf = city.select_one("wf").string
    tmn = city.select_one("tmn").string
    tmx = city.select_one("tmx").string
    print(name, ': ', wf, '(', tmn, '~', tmx, ')')
```