

# 데이터처리프로그래밍

## Module



강원대학교 교육혁신원 송혜정

<hjsong@kangwon.ac.kr>



# Module

## ✓ 학습목표

- Module 을 이해한다.

## ✓ 학습내용

- Module import
- Python 표준 모듈 라이브러리
  - os : Operating system functions
  - math: Mathematical functions
  - random : random number functions
  - datetime : 기본 날짜와 시간 형



# 강의에 앞서서..

- 본 강의자료는 아래의 자료들을 참고하여 만들어 졌음을 알립니다
  1. 데이터과학을 위한 파이썬 프로그래밍, 최성철, 한빛아카데미,2019
  2. Python (<https://docs.python.org>)
  3. 실용 파이썬 프로그래밍(Practical Python Programming)  
(<https://wikidocs.net/84360>)
  4. 점프 투 파이썬 (<https://wikidocs.net/book/1>)

# Module

- 모듈

- 변수, 함수, 클래스를 정의하여 묶어서 만든 코드 라이브러리
- import명령문을 사용하여 모듈을 사용

import 모듈명                      #모듈명으로 포함된 변수, 함수, 클래스 접근

Import 모듈명 as 별명          #별명으로 모듈에 포함된 변수, 함수, 클래스 접근

from 모듈 import 변수, 함수, 클래스    #변수, 함수, 클래스명으로 직접 접근

- Python 표준 모듈 라이브러리

(<https://docs.python.org/3/library/>)

- os : Operating system functions, os.getcwd(), os.chdir()
- math: Mathematical functions , math.sqrt(), math.pi
- random : random numb functions, random.random()
- datetime : 기본 날짜와 시간 형 ,

class datetime.date : date.today(), date.year

class datetime.time , class datetime.datetime

# Module

- Module import

```
#math 모듈의 log함수만 가져와 직접접근
from math import log
print(log(10))      #log함수 직접접근
print(sqrt(10))
```

2.302585092994046

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-2-930a704bf6ed> in <module>
      2 from math import log
      3 print(log(10))  #log함수 직접 사용 가능
----> 4 print(sqrt(10))
```

**NameError**: name 'sqrt' is not defined

# Module

- Module import

```
#math 모듈을 m이란 이름으로 가져오기
import math as m
print(m.log(10))
print(m.sqrt(10))
print(math.exp(10))
```

2.302585092994046  
3.1622776601683795

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-1-4eade9fa7cff> in <module>
      3 print(m.log(10))
      4 print(m.sqrt(10))
----> 5 print(math.exp(10))
```

**NameError:** name 'math' is not defined

```
#math 모듈의 함수, 클래스, 변수를 math로 접근
import math
print(math.log(10))
print(math.sqrt(10))
```

2.302585092994046  
3.1622776601683795

# Module

- os module
  - 운영체제 (operating system) 관련 모듈

```
#운영체제 동작 관련 모듈
import os
dir(os) #os 모듈에 포함된 변수, 함수, 클래스확인

'_exit',
'_fspath',
'_get_exports_list',
'_putenv',
'_unsetenv',
'_wrap_close',
'abc',
'abort',
'access',
'add_dll_directory',
'altsep',
'chdir',
'chmod',
```

# Module

- os module

```
os.getcwd()      # Return the current working directory
```

```
'C:\\Users\\hjsong\\Documents'
```

```
os.chdir('C:/Users/hjsong/Documents/python/python2021')  # Change current working directory
os.getcwd()      # Return the current working directory
```

```
'C:\\Users\\hjsong\\Documents\\python\\python2021'
```

```
os.listdir()
```

```
'Untitled4.ipynb',
'[10]Python_datastructure.ipynb',
'[11-1]Python_comprehension.ipynb',
'[11-2]Python_collection.ipynb',
'[12]Python_00P.ipynb',
'[13-1]Python_Inheritance.ipynb',
'[13-2]module.ipynb',
'[13-2]Python_module.ipynb',
'[2-1]JupyterNotebook_Markdown.ipynb',
'[2-2]Python_변수_자료형.ipynb',
```



# Module

- os module

```
import os
os.system('mkdir today') # 시스템 명령 처리 (디렉토리 생성명령, mkdir )
os.chdir('./today')
os.getcwd() # 현재 작업 디렉토리 확인
```

'C:\\Users\\hj song\\Documents\\python\\python2021\\today'

```
os.chdir('../') #이전 디렉토리로 이동
os.getcwd() # 현재 작업 디렉토리 확인
```

'C:\\Users\\hj song\\Documents\\python\\python2021'

```
os.listdir()
```

```
['.ipynb_checkpoints',
'Report10_상속_모듈.ipynb',
'Report2나이계산.ipynb',
'Report3_list.ipynb',
'Report4_if.ipynb',
'Report5_loop.ipynb',
'Report6_function.ipynb',
'Report7_단어장.ipynb',
'Report8_단어카운팅.ipynb',
'Report9.ipynb',
'today',
```

# Module

- os module

```
os.rmdir('today')  #디렉토리 삭제
```

```
os.listdir()
```

```
['.ipynb_checkpoints',
'Report10_삼속_모듈.ipynb',
'Report2나이계산.ipynb',
'Report3_list.ipynb',
'Report4_if.ipynb',
'Report5_loop.ipynb',
'Report6_function.ipynb',
'Report7_단어장.ipynb',
'Report8_단어카운팅.ipynb',
'Report9.ipynb',
'Untitled.ipynb',
'Untitled2.ipynb',
'Untitled3.ipynb',
'Untitled4.ipynb',
```

# Module

- math module
  - 수학함수, 수학 상수 정의 모듈  
 수학함수 : `ceil()`, `floor()`, `log()`, `sin()`, `cos()`, `sqrt()`  
 수학 상수 : `pi`, `e`

```
#올림(ceil()), 내림(floor()) 함수
import math
x = 35.6
x1 = round(x)      #반올림
x2 = math.ceil(x)  #무조건 올림
x3 = math.floor(x) #무조건 내림
print(x1)
print(x2)
print(x3)
```

```
36
36
35
```

```
#cos(), log()
print(math.cos(math.pi / 4))
print(math.log(1024, 2))
```

```
0.7071067811865476
10.0
```

# Module

- random module
  - 임의의 수를 생성 및 선택하는 함수를 포함

```
import random
l = ['apple', 'cherry', 'banana']
print(l)
random.shuffle(l)  #리스트의 순서를 랜덤으로 섞기
print(l)
c = random.choice(l) #리스트에서 임의로 1개 선택
print(c)
s = random.sample(range(100), 10) # 중복없이 샘플링, 0-99사이에 10개 랜덤 추출
print(s)
print("random.random()")
for i in range(5):
    r = random.random() # random float, 0~1사이의 랜덤 실수값
    print(r, end=" ")
print("\nrandom.randrange(6)")
for i in range(5):
    r = random.randrange(6) #range(6): 0~5범위의 랜덤 정수값
    print(r, end=" ")
print("\nrandom.randrange(1, 7)")
for i in range(5):
    r = random.randrange(1, 7) # range(1,7): 1~6 범위의 랜덤 정수값
    print(r, end=" ")
```

```
['apple', 'cherry', 'banana']
['banana', 'cherry', 'apple']
apple
[63, 67, 31, 6, 60, 8, 62, 24, 82, 27]
random.random()
0.33451776838297365 0.7821068575893978 0.43383207055810635 0.1253773514749631 0.7776232530127621
random.randrange(6)
4 2 2 2 2
random.randrange(1, 7)
1 6 3 6 3
```

# Module

## 예제1. 주사위를 100번 던져 각 값이 나올 확률 계산

```
from collections import Counter
t = 100
n = [ random.randrange(1, 7) for i in range(t)] #1-6사이의 랜덤 정수를 100번 추출하여 리스트로 생성
c = Counter(n) #리스트의 값을 카운팅
sc = sorted(c.items()) #정렬된 카운팅 결과 확인
print(sc)
p={}
for k, v in c.items(): #리스트의 카운팅값으로 각 키의 확률 계산
    p[k] = v / t #각개수를 전체개수로 나누어 확률 계산
sp = sorted(p.items()) #정렬된 확률 확인
print (sp)
```

```
[(1, 14), (2, 17), (3, 11), (4, 18), (5, 22), (6, 18)]
```

```
[(1, 0.14), (2, 0.17), (3, 0.11), (4, 0.18), (5, 0.22), (6, 0.18)]
```

# Module

## 예제2. 5개의 반지름을 랜덤 값으로 얻어서 면적 계산

```
import math
import random

def circleArea(r):
    a = math.pi * r ** 2
    return a

for i in range(5):
    r = random.random() * 100
    a = circleArea(r)
    out = 'radius={0:.2F} area={1:.2f}'
    print(out.format(r, a))
```

```
radius=98.64 area=30566.78
radius=48.32 area=7335.84
radius=21.18 area=1409.71
radius=40.01 area=5029.31
radius=17.74 area=988.29
```