모듈, 패키지, 라이브러리

Update – 2018.10

Contents

모듈 개요

- 모듈이란?

  함수, 변수, 클래스의 집합. 함수, 변수, 클래스들을 모아놓은 파일

  여러 변수, 함수, 클래스(다수의 변수+함수)를 저장해둔 파이썬 소스코드 파일.

  *.py 확장자를 이용하여 직접 만들 수 있다.

  다른 사람들에 의해서 만들어진 파이썬 라이브러리들을 모듈이라고 한다.

  모듈을 불러오는 방법 : import 명령을 통해서 모듈을 불러와 사용할 수 있다.

- 모듈의 종류
  - 표준 모듈 : 파이썬이 기본적으로 제공해주는 모듈
  - 외부 모듈 : 파이썬이 기본적으로 제공해주지 않는 모듈
  - 사용자정의 모듈 : 직접 만들어 사용하는 모듈

모듈 개요

- 파이썬이 모듈을 찾는 과정(초기화 과정)

  내장모듈인 경우 파이썬은 이미 어디에 모듈이 있는 지를 알고 있다
  내장 모듈이 아닌 경우(직접 만들거나, 남이 만들어 놓은 라이브러리들)에는 sys.path 변수에 정의된 디렉토리들을 검색한다.
  이 디렉토리에서 해당 모듈을 찾게되면 모듈 내부에 있는 명령(함수, 클래스)들을 읽어오게된다.
  초기화 과정은 모듈을 불러올 때 이루어진다.

모듈 import

모듈을 사용하기 위해서는 다음과 같은 문법을 이용하여 모듈을 임포트 해야한다.

- import <모듈이름>
- import <모듈이름> as <모듈변수>
- from <모듈> import <모듈이름>,<모듈이름>
- from 모듈이름 import 모듈함수1, 모듈함수2, ...
  - from ~ import ~ 문을 이용하면 모듈이름을 붙이지 않고 바로 해당 모듈 함수 사용 가능
  - (콤마)를 사용하지 않고 여러 함수를 불러올 수 있는 방법
  - from 모듈이름 import *

math 모듈

- 수학 함수를 사용할 수 있는 math 모듈

```
import math
import math as m
from math import pi, sin


print(math.pi)
print(m.pi)
print(math.sin(10))
```

```
3.141592653589793
3.141592653589793
-0.5440211108893698
```

random 모듈

- 수학 함수를 사용할 수 있는 math 모듈

```
import random
# random() - 0~1 사이의 실수
print(random.random())
# uniform(n1,n2) - n1~n2 사이의 실수
print(random.uniform(10,20))
```

```
0.06362769542842617
19.709011791182284
```

```
# randrange(n) - 0~n 까지의 정수 추출
print(random.randrange(10))
# randrange(n1,n2,2) - n1~n2 사이의 짝수 정수 추출
print(random.randrange(0,101,2))
```

```
6
42
```

random 모듈

- 수학 함수를 사용할 수 있는 math 모듈

```
# choice([item...] - 리스트의 아이템 중 추출
print(random.choice(['a','b','c','d']))
# shuffle(리스트변수) - 리스트를 재배열
list_a = [1,2,3,4,5,6,7]
random.shuffle(list_a)
print(list_a)
```

```
b
[1, 6, 7, 2, 3, 4, 5]
```

sys 모듈

- 사용하는 os 버전 사용 프로그램 정보 표시

```
import sys
print(sys.argv)
print(sys.version)
```

파이썬 인터프리터와 인터프리터가 실행중인 환경(자신의 시스템)에 관련된 기능들을 갖고 있다.

```
['C:₩₩Anaconda3₩₩lib₩₩site-packages₩₩ipykernel_launcher.py', '-f', 'C:₩₩Users₩₩queen₩₩AppData₩₩Roaming₩₩jupyter₩₩runtime₩₩kernel-f38f27cf-3a98-4ae7-955d-205fe791a41b.json']
3.6.4 |Anaconda custom (64-bit)| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]
```

sys 모듈

- sys.path : 파이썬 인터프리터, lib 등 패스 확인

```
>>>import sys
>>> sys.path
['', 'C:₩₩Anaconda3₩₩python36.zip',
'C:₩₩Anaconda3₩₩DLLs', 'C:₩₩Anaconda3₩₩lib',
'C:₩₩Anaconda3', 'C:₩₩Anaconda3₩₩lib₩₩site-
packages', 'C:₩₩Anaconda3₩₩lib₩₩site-
packages₩₩win32', 'C:₩₩Anaconda3₩₩lib₩₩site-
packages₩₩win32₩₩lib',
'C:₩₩Anaconda3₩₩lib₩₩site-packages₩₩Pythonwin']
```

sys 모듈

- 터미널창에서 입력한 문자열 출력하기

터미널창에서
입력한 문자열 출력하기

```
import sys
print("명령줄 인수 체크하기 !!!")
#sys.argv변수는 문자열 리스트이다.
#명령줄 인수들을 담고 있는 리스트이다.
for n in sys.argv:
    print (n)
print ("₩n₩n파이썬의 sys.path 디렉토리", sys.path)
```

sys 모듈

- 터미널창에서 입력한 문자열 출력하기

```
C:\__StudyPython\workspace\python_itdong_basic> python quiz_module_sys.py who are you?
명령줄 인수 체크하기!!!
quiz_module_sys.py
who
are
you?
파이썬의 sys.path 디렉토리 ['C:\\__StudyPython\\workspace\\python_itdong_basic',
'C:\\Anaconda3\\python36.zip', 'C:\\Anaconda3\\DLLs', 'C:\\Anaconda3\\lib', 'C:\\Anaconda3',
'C:\\Anaconda3\\lib\\site-packages', 'C:\\Anaconda3\\lib\\site-packages\\win32',
'C:\\Anaconda3\\lib\\site-packages\\win32\\lib', 'C:\\Anaconda3\\lib\\site-packages\\Pythonwin']
```

os 모듈

- 사용하는 os 버전 사용 프로그램 정보 표시

```
import os

print(os.name)

# Curent Working Directory

print(os.getcwd())

print(os.listdir())
```

```
nt
C:₩__StudyPython₩workspace₩python_Basic
['.ipynb_checkpoints', 'data', 'IF 조건문 퀴즈 - bmi.ipynb', 'json.ipynb', 'result.txt', '딕셔너리 자료형.ipynb', '람다함수.ipynb', '마크다운.ipynb'
```

urllib 모듈

- URL 페이지의 소스 표시

```
from urllib import request


target = request.urlopen('https://google.com')
output = target.read()
print(output))
```

datetime 모듈

- 년도, 월, 일, 시, 분, 초와 관련된 모듈

```
from datetime import datetime
now = datetime.now()
print(now.year, ' 년')
print(now.month, ' 월')
print(now.day, ' 일')
```

사용자정의 모듈

- 사용자 정의 모듈 파일 만들고 사용하기
  - 모듈 함수 이름 사용은 모듈이름.함수(인자)

```
# mod1.py
def sum(a, b):
    return a + b
```

```
# testMod.py
import mod1
print(mod1.sum(3,4))
```

사용자정의 모듈

- 2개의 함수가 정의된 모듈 파일을 제작하고 모듈을 임포트하여 함수를 호출한다.

```
# quiz_module1.py

def show_info(name):
    print("이름 :",name)

def math1(i, j):
    if (i+j) == 0:  show_info("제로")
    else:
        return i+j
```

```
# quiz_module2.py
import quiz_module1 as q;

q.show_info("황진희")
print(q.math1(200, 50))
```

```
이름 : 황진희
250
```

사용자정의 모듈

- 모듈의 name속성 ("__name__")
  - 모든 모듈은 이름을 갖고 있다.
  - 모듈의 name 속성을 이용하면 외부로부터 불러들여 졌을 때 곧바로 실행되지 않게 할 수 있다.
  - 파이썬의 모듈은 name 속성을 가지고 있는데 name이 'main'일 경우에는 사용자가 직접 실행한것임을 의미한다. 따라서, 적절하게 name속성을 활용하여 요구에 맞게 실행될 수 있도록 한다.

사용자정의 모듈

```
//python quiz_module__name__.py
def sum(i, j):
    return i+j

def differ(i, j):
    return i-j

if __name__ =="__main__":
    print(sum(1, 2))
    print(differ(1,2))
```

```
// python quiz_module__sub.py
import quiz_module__name__ as q;

print(q.differ(10,5))
print(q.sum(10,5))
```

메일 파일로 실행

모듈로 호출 후 실행

```
>>> python quiz_module__name__.py
3
-1
>>> python quiz_module__sub.py
5
15
```

사용자정의 모듈

- 클래스, 변수, 함수를 포함하는 모듈

```
# quiz_module_Class.py
PI = 3.14192

class Math:
    def aaa(self, r):
        return PI * (r**2)

def sum(i, j):
return i+j

if __name__ =="__main__":
    print(PI)
    bb = Math()
    print(bb.aaa(10))
    print(sum(PI, 10))
```

```
# quiz_module_Class2.py
import quiz_module_Class as q;

print(q.PI)
bbb = q.Math()
bbb.aaa(5)
```

```
>>> python  quiz_module_Class.py
3.14192
314.192
13.141919999999999
>>> python  quiz_module_Class2.py
3.14192
```

pyc 파일에 대한 이해

*.py는 파이썬 텍스트 소스파일이다.
*.pyc는 컴파일된 파이썬 바이너리파일이다.
*.pyo는 최적화된(Optimized) 컴파일된 파이썬 바이너리파일이다

- 파이썬에서는 모듈을 불러올 때 좀 더 빠르게 처리할 수 있도록 하기 위해서 .pyc 확장자를 가지는 바이트 코드를 활용한다.
- 파이썬에서 컴파일 할때 만들어놓는 파일 이다.
- 모듈을 불러올 때는 .pyc파일을 이용하여 불러오게된다. 불러올 때 선행작업을 수행하지 않고 더빨리 모듈을 불러올 수 있다.
- 바이트 코드는 플랫폼에 구애받지 않고 사용가능하다.
- 참고로, 파이썬이 저장 디렉토리에 쓰기 권한이 없을 경우에는 생성되지 않는다.

패키지

- 파이썬에서 폴더는 모듈을 담는 역할을 한다.
- __init.py__ 파일 포함

  : 3.3 이전 버전에서는 __init__.py 파일이 없으면 패키지로 인식이 되지 않음

  하위 버전과 호환을 위해서는 각 패키지 폴더에 __init__.py이 저장되어 있어야 한다.

  : 저장 폴더가 패키지의 일부임을 알려준다.

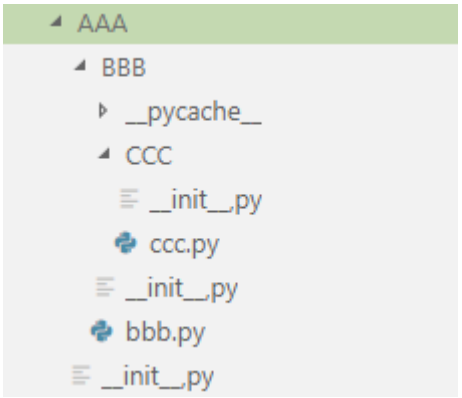
From 패키지1.패키지2.모듈명 import 함수명

from AAA.BBB.CCC.ccc import test_ccc

패키지

- 패키지 테스트. 다음과 같은 폴더 구조를 생성하고 __init__.py 파일을 생성한다.
- BBB, CCC 각 패키지 폴더안에 함수를 생성하고 임포트하여 테스트 한다.

quiz_package.py



AAA > BBB(bbb.py) > CCC(cccc.py)

quiz_package.py

```
# bbb.py
def test_bbb():
print("bbb")
```

```
# ccc.py
def test_ccc():
print("ccc")
```

```
# quiz_package.py
from AAA.BBB.bbb import test_bbb
test_bbb()

from AAA.BBB.CCC.ccc import test_ccc
test_ccc()
```

```
bbb
ccc
```

라이브러리

- 라이브러리(library) : 전세계의 파이썬 개발자(사용자)들에 의해서 이미 만들어진 프로그램
- 파이썬 라이브러리는 파이썬 설치시 자동으로 컴퓨터에 설치
- lib 폴더에 저장
- sys모듈, pickle모듈,io모듈, os모듈, glob모듈, time모듈, calendar모듈, random 모듈, 쓰레드(thread)모듈