파이썬 - HW4

임베디드스쿨1기

Lv1과정

2020. 08. 18

박하

1. Review - python module

1) import - 외부 모듈 가져옴

```
import math
print(math.pow(2,10))
print(math.log(100))
print(math.pi)

print(dir(math)) #해당 모듈 내에서 사용할 수 있는 것을 표시
```

```
1024.0
4.605170185988092
3.141592653589793
['__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh',
'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum',
'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'na
n', 'pi', 'pow', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
```

2) 사용자 모듈 만들기

```
import ex35_mod

print(dir(ex35_mod))

setA = [1, 3, 7, 10]
setB = [2, 3, 4, 9]

print(ex35_mod.union(setA, setB))
print(ex35_mod.intersect(setA, setB, [1,2,3]))
```

1. Review - python module

2) 사용자 모듈 만들기

```
import sys
print(sys.path)
from functools import *

def intersect(*ar):
    return reduce(_intersectSC, ar)

def _intersectSC(listX, listY):
    setList = []
    for x in listX:
        if x in listY:
            setList.append(x)
    return setList

def difference(*ar):
    setList = []
    intersectSet = intersect(*ar)
    unionSet = union(*ar)
    for x in unionSet:
        if not x in intersectSet:
            setList.append(x)
    return setList

def union(*ar):
    setList = []
    for item in ar:
        for x in item:
            if not x in setList:
                setList.append(x)
    return setList

def loadMathMod():
    print("import math")
    import math
    print(dir(math))

loadMathMod()

from ex35_mod import union

print(union([1,2,3],[3],[3,4]))
```

```
import math
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin',
[1, 2, 3, 4]
['WRAPPER_ASSIGNMENTS', 'WRAPPER_UPDATES', '__builtins__', '__cached__', '__doc__', '__fi
[1, 3, 7, 10, 2, 4, 9]
[3]
```

1. Review - python module

3) reduce (python3에서는 내장함수에서 빠짐 ?)
   - from functools import reduce 선언

```
from functools import reduce
jlist = [1,2,3,4,5]
reduce(lambda x, y: x+y, jlist)
```

```
15
```

4) __name__==’__main__’으로 직접 호출

2. Preview - python exception

1) try / except / finally

```
def divide(a, b):
    return a / b

try:
    c = divide(5, 0)
except ZeroDivisionError:
    print("Exception Occured!") #5/0은 0이므로 false조건이니, except가 실행됨
finally:
    print("항상 finally는 실행된다.") #finally는 무조건 실행
```

```
Exception Occured!
항상 finally는 실행된다.
```

2) Exception: ZeroDivisionError, TypeError // +else

```
def divide(a, b):
    return a / b

try:
    e = divide(5, 2)
except ZeroDivisionError:
    print("Exception Occured!") #5/0은 0이므로 false조건이니, except가 실행됨
except TypeError:
    print("숫자로 설정하시오!")
except:
    print("ZeroDIvisionError, TypeError외의 에러")
else:
    print("Result: {0}".format(e)) #except에서 걸리지 않으면 else 사용
finally:
    print("항상 finally는 실행된다.") #finally는 무조건 실행
```

```
Result: 2.5
항상 finally는 실행된다.
```

2. Preview - python exception

3) Exception: ArithmeticError (모든 산술 관련 에러 처리)

```
: def divide(a, b):
      return a / b

  try:
      c = divide(5, 'string')
  except TypeError as e:
      print("Error", e.args[0])  #args[0] : 0번째 인자

  try:
      d = divide(5,0)
  except ArithmeticError: #모든 산술 관련 에러 처리
      print("There are Arithmetic Error")
```

```
Error unsupported operand type(s) for /: 'int' and 'str'
There are Arithmetic Error
```

4) 파일 열고 닫기, IOError 처리(파일 I/O처리)

```
FilePath = './prepare.txt'

try:
    f = open(FilePath,'r')
    try:
        data = f.read(128)
        print(data)
    finally:
        f.close() #예외와 관련없이 무조건 close()해줌

except IOError: #File I/O  OR 기타 I/O 에러 처리
    print("Fail to open {0} file".format(FilePath))
```

```
prepare
```

2. Preview - python exception

5) raise

- 직접 예외 발생시키기 – 예외('에러메시지')

```
def RaiseErrorFunc():
    raise NameError

try:
    RaiseErrorFunc()
except:
    print("NameError Catched!")

def PropagateError():
    try:
        RaiseErrorFunc()
    except:
        print("Before Error Propagation")
        raise

PropagateError()
```

```
NameError Catched!
Before Error Propagation
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-30-b4333b8d935f> in <module>
     14         raise
     15
---> 16 PropagateError()

<ipython-input-30-b4333b8d935f> in PropagateError()
      9 def PropagateError():
     10     try:
---> 11         RaiseErrorFunc()
     12     except:
     13         print("Before Error Propagation")
```

Q1. 결과와 매칭이 안됨

2. Preview - python exception

5) Exception – NegativeDivisionError 사용자 예외 발생

```
class NegativeDivisionError(Exception): #3. b가 -일때 들어옴
    def __init__(self, value):
        self.value = value

def PositiveDivide(a, b):
    if(b < 0):
        raise NegativeDivisionError(b) #2. 사용자 예외 직접 처리
    return a / b

try:
    ret = PositiveDivide(10, -3) #1. 10/-3은 '-'값이 나오므로, NegativeDivisionError 예외로 이동
    print(' 10 / 3 = {0} '.format(ret))
except NegativeDivisionError as e:
    print('Error - Second argument of PositiveDivide is ', e.value) #4. 에러 메세지 처리
except ZeroDivisionError as e:
    print('Error - ', e.args[0])
except:
    print(e.args)
```

```
Error - Second argument of PositiveDivide is  -3
```

2. Preview - python exception

5) Exception – assert 조건, 작업 동작

```
def testAssert(x):
    assert type(x) == int, "Input value have to be integer"
    return x * 10

ret = testAssert("a")
print(ret)
```

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-40-d5bc58e442a3> in <module>
      3     return x * 10
      4
----> 5 ret = testAssert("a")
      6 print(ret)

<ipython-input-40-d5bc58e442a3> in testAssert(x)
      1 def testAssert(x):
----> 2     assert type(x) == int, "Input value have to be integer"
      3     return x * 10
      4
      5 ret = testAssert("a")

AssertionError: Input value have to be integer
```

Q2. 결과와 매칭이 안됨

3. 문제은행 1 - .py

```
'''
문제은행 1. 컴퓨터가 주사위를 굴리고 사용자 입력으로 맞추면 승리 틀리면 실패

컨셉. while문을 사용하여 컴퓨터 주사위를 랜덤으로 돌린 숫자와 사용자 입력을 받은 숫자를 비교한다. -> compare 함수에서 적용
     숫자가 같으면 "you win"/ 다르면 "you lose"를 프린팅한다.
     만약 게임을 그만하고 싶으면, 9를 입력하여 goto문으로 while문을 빠져나온다.
'''

import random as rd
from goto import with_goto

@with_goto
def compare():
    while 1:
        rand = rd.randint(1, 6)

        print("주사위 숫자를 입력하시오 (그만하려면 9를 누르시오)", end=' ')
        my_num = int(input())
        print('rand_num:{0} / my_num: {1} '.format(rand, my_num))

        if my_num == 9:
            goto .err_handler

        if my_num == rand:
            print("you win")
        else:
            print("you lose")

    label.err_handler

compare()
```

```
주사위 숫자를 입력하시오 (그만하려면 9를 누르시오) 3
rand_num:3 / my_num: 3
you win
주사위 숫자를 입력하시오 (그만하려면 9를 누르시오) 2
rand_num:6 / my_num: 2
you lose
주사위 숫자를 입력하시오 (그만하려면 9를 누르시오) 4
rand_num:3 / my_num: 4
you lose
주사위 숫자를 입력하시오 (그만하려면 9를 누르시오) 9
rand_num:4 / my_num: 9
```

3. 문제은행 2 - .py

```
'''
카지노의 3 슬롯 머신을 만들어보자.
컨셉. 10만원 현금넣고, 틀릴때마다 만원씩 감소, 맞으면 100배.
'''
import random as rd
from goto import with_goto

ALLOT = 100
user_money = 100000
m_money = 10000
p_money = 50000

@with_goto
def machine_game(user_money):

    money = user_money
    while 1:
        if money < 0:
            goto.err_handler

        rand_1 = rd.randint(1, 10)
        rand_2 = rd.randint(1, 10)
        rand_3 = rd.randint(1, 10)

        print("현재 잔액은 {0}입니다.".format(money))
        print("슬롯머신: {0} {1} {2}".format(rand_1, rand_2, rand_3))

        if (rand_1 == rand_2 == rand_3):
            money = money * ALLOT
            print("you win~")
            goto.err_handler
        else:

            money -= m_money
            print("You lose!")

    label.err_handler
    print("게임이 종료 되었습니다.")

print("본인의 현재 잔액은 {0}이고, 한번할때마다 {1}씩 감소, 맞추면 {2}씩 증가.".format(user_money, m_money,p_money))
machine_game(user_money)
```

```
본인의 현재 잔액은 100000이고, 한번할때마다 10000씩 감소, 맞추면 50000씩 증가.
현재 잔액은 100000입니다.
슬롯머신: 4 6 2
You lose!
현재 잔액은 90000입니다.
슬롯머신: 6 6 1
You lose!
현재 잔액은 80000입니다.
슬롯머신: 9 1 2
You lose!
현재 잔액은 70000입니다.
슬롯머신: 4 5 3
You lose!
```

3. 문제은행 3 - .py

```
0~100에 해당하는 난수를 100~200 개 사이로 생성한다.
전체 평균, 중앙값, 분산과 표준편차를 구하시오.

'''
import random as rd
import math
import numpy as np

def cal_mean(rand_val):
    cnt = len(rand_val)
    mean_ret = sum(rand_val)/cnt
    return mean_ret

def cal_median(rand_val):
    sort = sorted(rand_val)
    idx = len(sort) - 1
    median_idx = idx // 2
    median_ret = rand_val[median_idx]
    return median_ret

def cal_dispersion(rand_val,mean):
    disper = list(map(lambda x:pow(x-mean,2), rand_val))
    disper_ret = sum(disper)/len(rand_val)
    return disper_ret

def cal_standard(dispersion):
    std = math.sqrt(dispersion)
    return std

rand_val = []
for i in range(0, 200):
    rand_val.append(rd.randint(0,100))

print(rand_val)
mean = cal_mean(rand_val)
median = cal_median(rand_val)
dispersion = cal_dispersion(rand_val, mean)
standard = cal_standard(dispersion)

print("평균: {0} / 중위값: {1} / 분산: {2} / 표준편차: {3} ".format(mean, median, dispersion, standard))
print("[numpy] 평균: {0} / 중위값: {1} / 분산: {2} / 표준편차: {3} ".format(np.mean(rand_val), np.median(rand_val), np.var(rand_val), np.std(
```

```
[69, 49, 87, 34, 15, 83, 60, 4, 56, 67, 84, 69, 18, 24, 39, 70, 6, 20, 70, 34, 44, 39, 83, 85, 32
2, 62, 23, 32, 2, 56, 61, 79, 34, 3, 75, 47, 3, 51, 50, 97, 8, 7, 65, 42, 56, 10, 69, 37, 3, 69,
0, 72, 97, 74, 26, 96, 86, 29, 45, 25, 88, 45, 11, 36, 77, 30, 38, 42, 26, 23, 85, 47, 49, 45, 82
15, 91, 87, 47, 11, 99, 15, 71, 42, 16, 74, 33, 28, 0, 58, 11, 39, 35, 40, 28, 34, 21, 77, 96, 87
83, 90, 48, 66, 76, 36, 21, 91, 7, 11, 53, 30, 39, 43, 50, 53, 43, 44, 6, 89, 66, 70, 8, 29, 7, 0
1, 84, 79, 94, 14, 43, 75, 10, 27, 0, 23, 2, 59, 84, 42, 79, 86, 30, 25, 37, 68, 21, 44, 75, 14]
평균: 46.625 / 중위값: 74 / 분산: 810.174375 / 표준편차: 28.463562233142923
[numpy] 평균: 46.625 / 중위값: 43.5 / 분산: 810.174375 / 표준편차: 28.463562233142923
```

3. 문제은행 4 - .py

```python
from math import factorial, exp

rand_num = rd.randint(0,10)

def cal_pois(n, jlamb):
    result = (jlamb ** n) * exp(-jlamb) / factorial(n)
    return result


poisson = [cal_pois(n, 10) for n in range(40)]
print(poisson)
```

```
[4.5399929762484854e-05, 0.0004539992976248485, 0.002269996488124243, 0.007566654960414142, 0.018916637401035354, 0.03783327480207071,
0.06305545800345118, 0.09007922571921599, 0.11259903214901998, 0.1251100357211333, 0.1251100357211333, 0.11373639611012118, 0.0947803300
9176766, 0.07290794622443666, 0.05207710444602619, 0.03471806963068413, 0.021698793519177577, 0.012763996187751515, 0.00709110899319528
6, 0.003732162627997519 2, 0.0018660813139987594, 0.0008886101495232189, 0.0004039137043287358 4, 0.0001756146540559720 8, 7.31727725233217
2e-05, 2.9269109009328688e-05, 1.125734961897257e-05, 4.169388747767619e-06, 1.4890674099170067e-06, 5.134715206610368e-07, 1.7115717355
367894e-07, 5.521199146892869e-08, 1.7253747334040217e-08, 5.22840828304249e-09, 1.5377671420713203e-09, 4.393620405918058e-10, 1.220450
112755016e-10, 3.298513818256801e-11, 8.680299521728422e-12, 2.2257178260842107e-12]
```

3. 문제은행 5 - .py

```
'''
임의의 난수를 컴퓨터가 생성한다.
사용자의 입력을 받아 스무고개 게임을 진행하시오.
입력한 값과 난수의 대소 비교를 진행해서 사용자에게 알려준다.
맞추면 승리. 20번의 기회를 모두 소비하면 패배!
'''

import random as rd
from goto import with_goto

chance = 20
rand_num = rd.randint(0,100)

@with_goto
def compare(rand_num):
    for cnt in range(0,chance):
        if cnt == -1:
            goto .err_handler

        my_num = int(input())

        if rand_num > my_num:
            print("random 값이 더 큽니다.")
        elif rand_num < my_num:
            print("random 값이 더 작습니다.")
        else:
            print("정답입니다.")
            goto .err_handler
        print('기회는 {0}번 남았습니다.'.format(chance - (cnt+1)))
        cnt+=1


    label.err_handler
    print("\n기회가 끝났습니다.")
print('답: {0} \n'.format(rand_num))
print("값을 입력하십시오 ", end=' ')

compare(rand_num)
```

```
답: 92

값을 입력하십시오  30
random 값이 더 큽니다.
기회는 19번 남았습니다.
42
random 값이 더 큽니다.
기회는 18번 남았습니다.
60
random 값이 더 큽니다.
기회는 17번 남았습니다.
92
정답입니다.
```

3. 문제은행6 - .py

3. 문제은행 7 - .py

```
'''
1,3,4,7,11,18,29,47,76... 형태로 숫자가 진행된다.
23번째 숫자는 무엇일까?
'''

# fst, snd = 1, 3
# fibo = []
# for i in range(23):
#     fibo.append(fst)
#     fst, snd = snd, fst+snd
#
# print("23번째 숫자: {0}".format(fibo[22])) #23번째 index값


def fibo(idx):
        if idx == 1:
            return 1
        elif idx == 2:
            return 3
        else:
            jlist = fibo(idx-1)+fibo(idx-2)
            return jlist

print(fibo(23))
```

```
'''
예를들어,

fibo(5)라고 하면
fibo(4)+fibo(3)
(fibo(3)+fibo(2))+fibo(3)
(((fibo(2)+fibo(1))+fibo(2))+fibo(3)    ##fibo(2) = 3, fibo(1) = 1  이것을 대입한다.


((3+1)+3)+fibo(3)                       ##다시 fibo(3)은 함수호출로 (fibo(2)+fibo(1))가 된다.
(7)+(fibo(2)+fibo(1))                   ##fibo(2) = 3, fibo(1) = 1  이것을 대입한다.
7+(3+1)
=11                                     ##5번째의 결과값
'''
```

```
64079
```

3. 문제은행 8 - .py

```
'''
1,1,2,3,5,8,13,21,34,55 형태로 숫자가 진행된다.
1~57번째까지의 수들로 홀수들의 합을 하고 짝수들의 합을 구한다.
홀수들의 합 - 짝수들의 합의 결과를 출력하시오
'''

CNT = 57
jlist = []

def cal_sum(jlist):
    odd = []
    even = []
    for i in range(CNT):
        if jlist[i] % 2 == 0:
            even.append(jlist[i])
        else:
            odd.append(jlist[i])

    even_num = sum(even)
    odd_num = sum(odd)
    print("홀수 합: {0}, 짝수 합: {1}".format(odd_num, even_num))

fst, snd = 1, 1
fibo = []
for i in range(CNT):
    fibo.append(fst)
    fst, snd = snd, fst+snd


jlist = fibo
cal_sum(jlist)
```

```
홀수 합: 478361013020, 짝수 합: 478361013020
```

3. 문제은행1 - .C

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>


#define RAND rand() % 6 + 1

int main(void)
{

        int my_num, rand_num;
        while (1)
        {
                srand(time(NULL));

                printf("주사위를 입력하시오(그만하려면 9를 누르시오): ");
                scanf("%d", &my_num);
                printf("rand_num: %d / my_num: %d\n", RAND, my_num);

                if (my_num == 9)
                        goto STEP_1;
                if (my_num == RAND)
                        printf("You Win\n");
                else
                        printf("You lose");

        }
STEP_1:
        printf("끝났습니다.\n");
}
```

```
주사위를 입력하시오(그만하려면 9를 누르시오): 4
rand_num: 6 / my_num: 4
You lose주사위를 입력하시오(그만하려면 9를 누르시오): 2
rand_num: 6 / my_num: 2
You lose주사위를 입력하시오(그만하려면 9를 누르시오): 3
rand_num: 3 / my_num: 3
You lose주사위를 입력하시오(그만하려면 9를 누르시오): 9
rand_num: 4 / my_num: 9
끝났습니다.
```