사용자 정의 모듈

10주차_02_03

한 동 대 학 교
김경미 교수

학습목표

- 사용자가 만드는 모듈 정의하고 이해하기
- 다양한 모듈 소개

사용자가 만드는 모듈

- 자주 사용하는 함수들을 모아서 모듈로 사용 가능
- 관련 있는 함수들은 모아서, 몇 개의 사용자 정의 모듈을 만들면
  - 코딩 소요 시간 감소
  - 다른 사람과 같이 작업할 때, 공유해야 함

- 사용 방법
  - 사용하고 싶은 함수들이 있는 파일 이름을 import
  - import 한 모듈을 하나의 객체로 받음
  - 객체.함수이름()

사용자가 만드는 모듈 I


```
# filename; fibo.py
# Fibonacci numbers module

def fib(n) :
    if n == 0 :
        return 0
    elif n == 1 :
        return 1
    else :
        return fib(n-1) + fib(n-2)

def ifib(n) :
    a = 0
    b = 1
    for i in range(n) :
        a = b
        b = a + b
    return a
```

Fibo.py안에 함수 2개를 선언한다
이 함수들을 사용하고 싶은 곳에서,
import fibo
쓰면, 저장되어 있는 함수 2개 사용 가능하다

```
>>> import fibo
>>> f=fibo
>>> f.fib(11)
89
>>> f.ifib(11)
1024
>>>
```

사용자가 만드는 모듈 2


```
# filename; calculator.py
# add,subtract numbers module

def add(a,b) :
    result = a+b
    return result

def subtract(a,b) :
    result = a-b
    return result
```

```
>>> import calculator
>>> c = calculator
>>> c.add(5,3)
8
>>> c.subtract(8,2)
6
>>>
```

사용자가 만드는 모듈 3


```
# filename; List_index.py
# mid,end list module


def mid_list(a) :
    len_list = len(a)
    if len_list%2!=0:
        len_list = len_list-1
    mid = len_list//2
    return a[mid]


def end_list(a) :
    len_list = len(a)
    return a[len_list-1]
```

```
>>> import List_index
>>> l = List_index
>>> list1 = [1,3,5,7,9]
>>> l.mid_list(list1)
5
>>> l.end_list(list1)
9
```

연습문제 I

- 자주 사용하는 함수 3개를 저장하여 'freq.py'에 저장한다
- import freq 사용하여 저장 된 함수 3개를 사용해 본다

연습문제 I 코드


```
from datetime import date

def cal_birthday(month, day):
    today = date.today()
    birthday = date(today.year, month, day)
    due = birthday - today
    if due.days < 0 :
        next_birthday = date(today.year + 1, birthday.month, birthday.day)
        due = next_birthday - today

    print("생일까지 남은 날짜는: ", due.days)

import math
import cmath

def deter(a, b, c):
    return math.pow(b, 2) - 4*a*c
# continue to..
```

연습문제 I 코드


```
def roots_formula(a, b, c):
    if deter(a,b,c) >= 0:
        root01 = (-b + math.sqrt(deter(a,b,c)))/ (2*a)
        root02 = (-b - math.sqrt(deter(a,b,c)))/ (2*a)
    else:
        root01_real = -b/(2*a)
        root01_imag = (math.sqrt(math.fabs(deter(a,b,c))))/ (2*a)
        root02_real = -b/(2*a)
        root02_imag = (math.sqrt(math.fabs(deter(a,b,c))))/ (2*a)
        root01 = root01_real + root01_imag * 1j
        root02 = root02_real - root02_imag * 1j

    return [root01, root02]
```

```
>>> import freq
>>> f=freq
>>> print(f.roots_formula(1,2,3))
[(-1+1.4142135623730951j), (-1-1.4142135623730951j)]
>>> f.cal_birthday(11, 24)
생일까지 남은 날짜는:  147
>>>
```

연습문제 2

- ‘operators.py’에 덧셈, 뺄셈, 곱셈 과정과 결과를 함께 출력하는 함수 3개를 만든다
- Import 하여 사용해본다

연습문제 2 코드


```
#operators.py

def add(a, b):
    print(a, '+', b, '=', a+b)

def mul(a, b):
    print(a, '*', b, '=', a*b)

def min(a, b):
    print(a, '-', b, '=', a-b)
```

```
>>> import operators
>>> equation = operators
>>> equation.add(1,5)
1 + 5 = 6
>>> equation.min(90, 42)
90 - 42 = 48
>>> equation.mul(9, 9)
9 * 9 = 81
>>>
```

강의 요약

- 사용자가 만드는 모듈 정의하기
  - 자주 사용하는 함수들을 모아서 모듈로 정의
  - 관련 함수들을 모아 공동 작업자와 공유 가능

목표 달성 질문

- 사용자가 정의하는 모듈을 만들 때, 해당 파일의 확장자는 무엇으로 해야 하는가?

감사합니다