

함수

함수 (function)

- 함수
 - 특정 기능을 하도록 설계된 단위 프로그램
 - 형식

```
# 정의
def function_name(arg1, arg2,...):
    """ doc string """
    ...
    return ...

# 호출
ret = function_name(arg_value1, arg_value2,...)
```

```
def sum(x,y):
    z=x+y
    return z
```

```
a=10
b=sum(a,20)
print(b)
```

30

```
def sum(x,y):
    z=x+y
    print(z)
```

```
a=10
sum(a,20)
```

30



함수 (function)

리스트 인자

```
def list_add(list):
    return sum(list)

def list_add2(list):
    sum=0
    for item in list:
        sum += item # sum = sum + item
    return sum

nums=[1,2,3]
print(list_add(nums))
print(list_add2(nums))
```

인자의 자료형

```
def add(x,y):
    z=x+y
    return z

print(add(1,2))
print(add("my", "string"))
print(add([1,2],[3,4]))
```

3
mystring
[1, 2, 3, 4]

디폴트(default) 인자

```
def add(x=5,y=20):
    z=x+y
    print(z)

add(10)
add(10,10)
add(y=10)
```

30
20
15

여러 값을 리턴하기

```
def swap(x,y):
    return (y,x)

print(swap(1,2))
```

(2, 1)

함수 (function)

- 가변형 인자 (variable argument)
 - 다수의 인자를 튜플이나 딕셔너리로 받는 것
 - 형식
 - 튜플 가변형 인자: *args
 - 딕셔너리 가변형 인자: **args

```
def scaled_add(c,*args):
    return c*sum(args)

def showme(*args):
    print(args)

print(scaled_add(2,1,2,3))
showme(1,2,3)
```

12
(1, 2, 3)

```
def super_add(*args,**kargs):
    c=kargs.get('scale')
    o=kargs.get('offset')
    return o+c*sum(args)

def showmemore(**kargs):
    print(kargs)
```

```
print(super_add(1,2,3,scale=2,offset=0))
showmemore(scale=2,offset=0)
```

12
{'scale': 2, 'offset': 0}



외부 함수 사용하기 (import)

```
import math  
math.sin(3.14)
```

```
from math import sin  
sin(3.14)  
math.cos(3.14)
```

```
from math import sin, cos  
sin(3.14)  
cos(3.14)
```

```
from math import * # 사용시 주의!  
sin(3.14)  
cos(3.14)
```

```
import math as m  
m.sin(3.14)  
m.cos(m.pi)
```

```
import module  
import module as m  
from module import f1, f2  
from module import *
```

수학 함수 (math library)

Command name	Description	Constant	Description
abs(<i>value</i>)	absolute value	e	2.7182818...
ceil(<i>value</i>)	rounds up	pi	3.1415926...
cos(<i>value</i>)	cosine, in radians		
floor(<i>value</i>)	rounds down		
log(<i>value</i>)	logarithm, base e		
log10(<i>value</i>)	logarithm, base 10		
max(<i>value1</i> , <i>value2</i>)	larger of two values		
min(<i>value1</i> , <i>value2</i>)	smaller of two values		
round(<i>value</i>)	nearest whole number		
sin(<i>value</i>)	sine, in radians		
sqrt(<i>value</i>)	square root		

모듈 (module)

- 모듈
 - 다른 파이썬 프로그램에서 import하여 쓸 수 있게 만들어진 파일
 - 함수, 변수, 클래스 등을 모아놓은 파일

```
# module calculator.py  
  
def add(a,b):  
    return a+b  
  
if __name__ == "__main__":  
    print(add(1,2))
```



```
# module do_calcs.py  
  
import calculator  
  
def main():  
    print(calculator.add(3,4))  
  
if __name__ == "__main__":  
    main()
```

모듈 (module)

- if __name__ == "__main__":
 - 현재 모듈이 main(구동 프로그램)인지 판단
 - 직접 이 파일을 실행시켰을 때는 참이 되어 if문 다음 문장들을 수행함
 - 다른 파일에서 이 모듈을 호출했을 때는 거짓이 되어서 if문 다음문장들을 수행하지 않음

모듈 (module)

```
# module calculator.py
```

```
def add(a,b):  
    return a+b
```

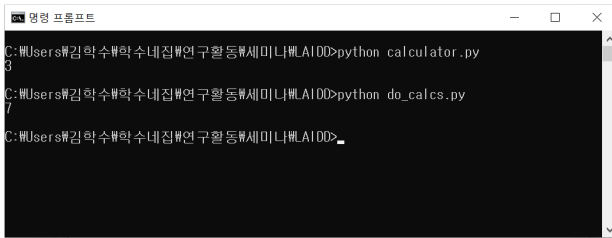
```
if __name__ == "__main__":  
    print(add(1,2))
```

```
# module do_calcs.py
```

```
import calculator
```

```
def main():  
    print(calculator.add(3,4))
```

```
if __name__ == "__main__":  
    main()
```



```
C:\Users\김기수>python calculator.py  
3  
C:\Users\김기수>python do_calcs.py  
7  
C:\Users\김기수>
```



Edited by Harksoo Kim

실습

- 입력 받은 문자열을 다음과 같은 방법으로 암호화하고 복호화하는 프로그램을 작성하시오.
 - 암호화
 - 공백(스페이스, 탭)으로 분리하고 역순으로 배치
 - 아스키 코드를 key만큼 증가
 - 복호화
 - 아스키 코드를 key 감소
 - 공백(스페이스, 탭)으로 분리하고 역순으로 배치
 - 예제: key가 1인 경우
 - "abc def" → 암호화 → "efg!bcd"
 - "efg!bcd" → 복호화 → "abc def"



Edited by Harksoo Kim

실습

?



Edited by Harksoo Kim

실습

- n 명의 이름과 전화번호를 입력 받아 전화번호 사전을 만들고, 이름과 전화번호로 검색하는 프로그램을 작성하시오.

?



Edited by Harksoo Kim

질의응답

Q & A

Homepage: <http://nlp.konkuk.ac.kr>
E-mail: nlpdrkim@konkuk.ac.kr



Edited by Harksoo Kim