4. PYTHON 입력과 출력

KOREA DIGITAL MEDIA HIGH SCHOOL 2020 PYTHON LECTURE

04-1 함수

- 함수(function)의 정의

```
def 함수 이름(매개변수):
    수행할 문장1
    수행할 문장2
    ...
```

- Def는 함수를 만들 때 사용하는 예약어.
- 함수 이름은 사용자가 임의로 작성.
- 함수 이름 뒤 괄호안의 매개변수는 함수에 입력으로 전달되는 값을 받는 변수.

```
def add(a,b): #a,b는 매개변수
    return a+b
```

```
a=3
b=4
c=add(a,b)
print(c)
print(add(5,6)) #5,6은 인수
```

04-1 함수

- 함수의 매개변수와 인수
  - 매개변수(parameter) : 함수에 입력으로 전달된 값을 받는 변수
  - 인수(argument) : 함수를 호출할 때 전달하는 입력값


```
def 함수 이름(매개변수):
    수행할 문장
    ...
    return 결괏값
```

04-1 함수

- 함수의 종류
  - 입력값이 없는 함수

```
def say():
    return 'Hello'
```

  - 결과값이 없는 함수

```
def add(a,b):
    print("%d, %d의 합은 %d입니다." %(a,b,a+b))
```

  - 입력값도 결과값도 없는 함수

```
def say():
    print('Hi')
```

04-1 함수

- 함수의 호출
  - 매개변수 지정하여 호출하기

```
def add(a,b): #a,b는 매개변수
    return a+b
```

```
result=add(a=3,b=7)
print(result)
```

  - 여러 개의 입력 값을 받는 함수 만들기1

```
def add_many(*args):
    result = 0
    for i in args:
        result = result + i
    return result
```

04-1 함수

- 함수의 호출
  - 여러 개의 입력 값을 받는 함수 만들기2

```
def add_mul(choice, *args):
    if choice=="add":
        result = 0
        for i in args:
            result=result+i
    elif choice=="mul":
        result = 1
        for i in args:
            result = result * i
    return result

result = add_mul('add',1,2,3,4,5)
print(result)

result = add_mul('mul',1,2,3,4,5)
print(result)
```

04-1 함수

- 키워드 파라미터(키워드 매개변수)

```
def print_kwargs(**kwargs):
    print(kwargs)

print_kwargs(a=1)

print_kwargs(name='foo', age=3)
```

매개변수 이름 앞에 **을 붙이면 결과값이 딕셔너리가 저장된다.

04-1 함수

- 함수의 특징
  - 함수의 결과값은 언제나 하나이다.

```
def add_and_mul(a,b):
    return a+b, a*b


result=add_and_mul(5,7)
print(result)
```

결과값은 2개 변수는 1개. 오류는 발생하지 않지만 튜플값 하나로 반환한다.
두 개의 값으로 반환하고 싶다면 다음과 같이 분할하여 선언한다.

```
def add_and_mul(a,b):
    return a+b, a*b


result1, result2=add_and_mul(5,7)
print(result1,result2)
```

04-1 함수

- 함수의 특징
  - 매개변수에 초기값 미리 설정하기

```
def say_myself(name,old,man=True):
    print("name %s" %name)
    print("age %d" %old)
    if man:
        print("Male")
    else:
        print("Female")
say_myself("Jeff", 19)
say_myself("Jeff", 19, True)
```

- 매개변수에 초기값을 미리 설정할 수 있다. 다만 매개변수의 위치는 반드시 지켜야 한다.

04-1 함수

- 함수와 변수(function and variable)
  - 함수 안에서 선언한 변수의 효력 범위

```
#vartest
v1=1
def vartest(v1,v2):
    v1=v1+1
    v2=v2+1

vartest(1,1)
print("v1=",v1)
print("v2=",v2)
```

- 함수안에서 새로 만든 변수는 "**함수만의 변수**"이다.
- 함수 안에서 사용하는 매개변수는 함수 밖의 변수 이름과는 전혀 상관없다.
- 그러므로 변수 v2는 **오류**가 발생한다.

04-1 함수

- 함수 안에서 함수 밖의 변수를 변경하는 방법
  - return 사용하기

```
#vartest_return.py
v3=1
def vartest(v3):
    v3=v3+1
    return v3

v3=vartest(v3)
print(v3)
```

04-1 함수

- 함수 안에서 함수 밖의 변수를 변경하는 방법
  - global 명령어 사용하기

```
#vartest_golbal
v4=1
def vartest():
    global v4
    v4=v4+1

vartest()
print(v4)
```

04-1 함수

- 또 다른 사용자 함수 lambda
  - 함수를 한 줄로 간결하게 만들 때 사용
  - return 없이도 결과값을 돌려준다.

```
lambda 매개변수1, 매개변수2, ... : 매개변수를 사용한 표현식
```

```
#lambda
add=lambda a, b: a+b
result=add(3,4)
print(result)
```

04-2 사용자 입력과 출력

- input()의 사용함수를 한 줄로 간결하게 만들 때 사용
  - 입력값은 모두 문자열로 취급.
- print()
  - 큰따옴표로 둘러싸인 문자열은 +연산과 동일하다.
  - 문자열 띄어쓰기는 콤마(,)로 한다.
  - 이어쓰기 : 한줄에 결과값 출력하기 end=' '

04-3 파일 읽고 쓰기

- 파일 생성하기

```
파일 객체 = open(파일 이름, 파일 열기 모드)
```

- 파일열기모드

파일 열기 모드	설명
r	읽기 모드 - 파일을 읽기만 할 때 사용
w	쓰기 모드 - 파일에 내용을 쓸 때 사용
a	추가 모드 - 파일의 마지막에 새로운 내용을 추가할 때 사용

04-3 파일 읽고 쓰기

- 파일을 쓰기 모드로 열어 출력값 적기

```
#writedata
f=open("./test.txt",'w')
for i in range(1,11):
    data = "%d번째 줄₩n" %i
    f.write(data)
f.close()
```

04-3 파일 읽고 쓰기

- 파일을 읽는 여러가지 방법-readline

```
#readline_all
f=open("./test.txt","r")
while True:
    line=f.readline() #한줄씩 읽어 들인다.
    if not line: break
    print(line)
f.close()
```

04-3 파일 읽고 쓰기

- 파일을 읽는 여러가지 방법-readlines(), read()

```
#readlines
f=open("./test.txt","r")
lines=f.readlines() #각각의 줄을 요소로 갖는 리스트로 돌려준다.
for line in lines:
    print(line)
f.close()
```

```
#read
f=open("./test.txt","r")
data=f.read() #파일전체의 내용을 문자열로 돌려준다.
print(data)
f.close()
```

04-3 파일 읽고 쓰기

- 파일에 새로운 내용 추가하기
  - 쓰기모드('w')로 파일을 열 때 존재하는 파일을 열면 내용이 모두 사라진다. 새로운 값을 추가하려면 추가모드('a')로 열어야 한다.

```
#adddata
f=open("./test.txt","a")
for i in range(11,20):
    data="%d 추가줄₩n" %i
    f.write(data)
f.close()
```

04-3 파일 읽고 쓰기

- with문과 함께 사용하기
  - with블록이 종료되면 열린 파일 객체가 자동으로 닫힌다.

```
#with
with open("./test.txt",'w') as f:
    f.write("Python is Life.")
```

Thank You

KOREA DIGITAL MEDIA HIGH SCHOOL 2020 Python LECTURE

PNG FILE