함수

최석재
lingua@naver.com

파이썬 함수

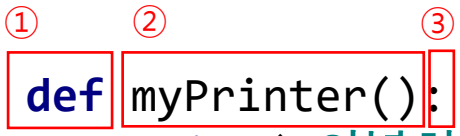
- 코드 작성 시 같은 내용을 재사용할 때가 있다
- 함수는 같은 내용을 반복해서 사용할 수 있게 해준다

```
def myPrinter():
    print("안녕하세요, 최석재님!")


myPrinter()          # 사용

print("*"*10)

myPrinter()          # 반복
```



① def 는 함수를 만들겠다고 선언하는 키워드
② 함수의 이름
③ 콜론

함수는 여러 줄

- 함수는 보통 두 줄 이상으로 작성되게 된다

```
def sayhello():
    print("안녕")
    print("처음 파이썬 함수를 만들었구나~~")
    print("좋은 하루!")


sayhello()
```

반환값이 있는 함수

- 함수가 어떤 결과를 내보내게 할 수 있다

```
def giveCoin():
    return 10

print("환영! {} 코인을 드립니다".format(giveCoin()))
```

매개변수도 있는 함수

- return 문은 매개변수와 함께 쓰는 경우가 많다
- 매개변수는 사용자 입력값으로, 함수의 결과를 다르게 할 수 있다

```
def giveCoin(level):
    coin = 10
    if level == 1:
        coin = 10
    elif level == 2:
        coin = 20
    elif level == 3:
        coin = 30
    return coin

print("환영! {} 코인을 드립니다".format(giveCoin(level=1)))
print("환영! {} 코인을 드립니다".format(giveCoin(level=2)))
print("환영! {} 코인을 드립니다".format(giveCoin(level=3)))
```

연습문제 1

- 절대값을 구하는 코드는 다음과 같이 작성할 수 있다

```
if number < 0:
    number = -number
```

- 위의 코드를 이용하여 절대값을 구하는 abs 함수를 구현하세요

```
3을 입력한 결과는 3
-3을 입력한 결과는 3
```

매개변수만 있는 함수

- return 문 없이 매개변수만 있는 함수는 있는 경우는 종종 있다

```
def sayhello(when):
    if when == "아침":
        print("Good Morning!")
    elif when == "점심":
        print("Good Afternoon~")
    elif when == "저녁":
        print("Good Night")


sayhello("아침")
sayhello("점심")
sayhello("저녁")
```

두 개 이상의 매개변수

- 매개변수는 둘 이상이 될 수 있다

```
def calTwo(num1, num2):
    result = num1 + num2
    print("당신이 입력한 함수는 {}, {}입니다.".format(num1, num2))
    print("두 함수의 합은 {}입니다.".format(result))


calTwo(3, 5)
```

다양한 리턴값

```
None
100
{10, 11, 12}
(10, 11, 12)
```

```
def returnNone(value):
    x = value

def returnValue(value):
    x = value * 10
    return x

def returnSet(value):
    x = {value, value+1, value+2}
    return x

def returnTuple(value):
    return value, value+1, value+2

print(returnNone(10))                # return 값이 없음
print(returnValue(10))               # return 값  있음
print(returnSet(10))                 # 여러 개의 return 값을 집합형으로 묶어서 보냄
print(returnTuple(10))               # 여러 개의 return 값을 튜플형으로 묶어서 보냄
```

매개변수의 기본값

- 매개변수에 기본값을 설정해 둘 수 있다

```
def report(message, who="Everyone"):
    print(message, who)


report("Good Morning")
report("Good Morning", "Mr. Choi")
```

```
Good Morning Everyone
Good Morning Mr. Choi
```

가변 인수

- 매개변수가 몇 개 들어올지 사전에 결정할 수 없을 경우, 가변 인수를 사용할 수 있다
- *표만 사용하면 되나, 가변 인수는 *arg, *args 를 사용하는 경우가 많다

```
def select_even(*args):
    result = []
    for num in args:
        if num % 2 != 1:
            result.append(num)
    return result

even = select_even(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
print(even)
```

```
[2, 4, 6, 8, 10]
```

가변 인수 사용시 주의사항

- 가변 인수는 사용을 조금 어렵게 느끼게 하므로 자주 사용하지는 않는다

- 가변 인수를 사용할 때는 한 함수에 하나만 사용할 수 있다
- 가변 인수 뒤에는 매개 변수가 올 수 없다

- def func(*args) # 가능
- def func(a, *args) # 가능
- def func(a, b, *args) # 가능
- def func(*args1, *args2) # 불가능
- def func(*args, c) # 불가능

연습문제 2

- 방금 작성한 가변 인수 함수를 일반 리스트 변수를 받는 함수로 수정하세요

함수 도움말

- 따옴표 세 개를 사용하여 함수가 어떤 기능을 하는지를 작성할 수 있다
- help() 함수로는 작성된 도움말을 볼 수 있다

```
def select_even(*args):
    """주어진 범위에서 홀수를 리스트로 만들어 주는 함수입니다.
    args: 홀수로 만들 숫자의 목록"""
    result = []
    for num in args:
        if num % 2 != 1:
            result.append(num)
    return result

even = select_even(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
print(even)

help(select_even)
```

내장함수 – abs()

- 파이썬이 가지고 있는 주요 내장함수를 살펴본다
- abs() 함수는 절대값(absolute value)을 취한다

```
num1 = 10
num2 = -10

print(abs(num1))
print(abs(num2))          # 모두 양수로 바뀌었다
```

내장함수 – bool()

- bool() 함수는 주어진 값이 참인지 거짓인지를 알려준다

```
found = False

if(found):
    print("찾았습니다")
else:
    print("못 찾았습니다")
```

- 부울형이 아닌 다른 자료형에도 False를 가질 수 있다
- 기본적으로 '없으면' False이다

```
num0 = 0
num1 = 1

list0 = []
list1 = [1, 2, 3]

str0 = ""
str1 = "파이썬"

print(bool(num0))       # False. 정수 0은 거짓으로 다룬다
print(bool(num1))       # True. 그 외의 값은 모두 참이다

print(bool(list0))      # False. 빈 리스트는 거짓이다
print(bool(list1))      # True. 항목이 하나라도 있으면 참이다

print(bool(str0))       # False. 빈 문자열은 거짓이다
print(bool(str1))       # True. 문자가 들어있으면 참이다
```

내장함수 – float()

- float() 함수는 주어진 값을 실수로 변환해 준다

```
num = 10

num_new = float(num)

print(num)                  # 원래 값. 10
print(num_new)              # 변환 값. 10.0
```

- 문자열도 숫자로 되어 있으면 실수로 변환해 준다

```
num2 = "20"

num2_new = float(num2)

print(num2)                 # 원래 값. 20
print(num2_new)             # 변환 값. 20.0
```

내장함수 – int()

- int() 함수는 주어진 값을 정수로 변환해 준다

```
# 정수는 그대로 정수로 출력한다
num = 10.0
num_new = int(num)

print(num)


# 실수는 정수로 변환한다
num = 10.0
num_new = int(num)

print(num)                  # 원래 값
print(num_new)              # 변환 값
```

- 그러나 문자열의 정수로 변환은 정수 형태만 변환할 수 있다
- 실수 형태의 문자열을 넣으면 오류가 발생한다

```
num2 = "20"                 # "20.0" 은 불가

num2_new = int(num2)

print(num2)                 # 원래 값
print(num2_new)             # 변환 값
```

연습문제 3

- 사용자로부터 정수를 입력받아 실수를 출력하는 프로그램을 작성하세요

내장함수 – len()

- len() 함수는 길이를 구하는 함수이다

```
mystr = 'python'
print(len(mystr))              # 문자의 수 6

mystr = '파이썬'
print(len(mystr))              # 문자의 수 3

mylist = [1, 2, 3, 4, 5, 6]
print(len(mylist))             # 원소의 수 6

mydict = {0:'바다', 1:'육지', 2:'하늘'}
print(len(mydict))             # 원소의 수 3
```

연습문제 4

- 대한민국 대통령의 순서와 이름을 출력하는 프로그램을 작성하시오

```
 1대 대통령 : 이승만
 2대 대통령 : 윤보선
 3대 대통령 : 박정희
 4대 대통령 : 최규하
 5대 대통령 : 전두환
 6대 대통령 : 노태우
 7대 대통령 : 김영삼
 8대 대통령 : 김대중
 9대 대통령 : 노무현
10대 대통령 : 이명박
11대 대통령 : 박근혜
12대 대통령 : 문재인
13대 대통령 : 윤석열
```

내장함수 – pow(), round(), sum()

```
print(pow(10, 2))        # 10의 2승
print(10**2)             # 동일한 결과

print(round(2.3))        # 반올림하여 정수를 도출
print(round(3.41))

mylist = [1, 2, 3, 4, 5]
print(sum(mylist))       # 모든 값을 더한다
```

```
100
100
2
3
15
```

내장함수 – max(), min()

- 최대값, 최소값을 찾는다

```
mynum = [56, 79, 82, 1, 22, 99]
print("maximum value:", max(mynum))      # 최대값을 찾는다
print("minimum value:", min(mynum))      # 최소값을 찾는다
```