

# CS101 – 매개 변수와 반환값을 가진 함수

## Lecture 8

School of Computing  
KAIST

학습 목표:

- 매개 변수와 반환값을 가진 함수를 이해하고 사용할 수 있다.

함수(Function)라는 이름은 수학에서 비롯되었습니다.

수학에서의 함수의 정의는 한 집합의 임의의 한 원소를, 다른 집합의 오직 한 원소에 대응시키는 대응 관계입니다.

$$f : \mathbb{R} \rightarrow \mathbb{R}$$

$$x \rightarrow \pi \times \frac{x}{180.0}$$

이 예시에서  $x$ 는 함수의 인자(Argument)이고,  $f(x)$ 는 함수의 결과입니다.

Python에서의 함수 역시 인자를 전달받아 결과를 반환합니다.

```
def to_radians(deg):
    return (deg / 180.0) * math.pi
```

```
>>> a = to_radians(90)
```

```
>>> print(a)
```

```
1.5707963267948966
```

# 유용한 함수

Python은 많은 내장 함수를 제공하고 있습니다.

**형 변환 함수**는 객체의 형태를 다른 형태로 바꿔주는 함수입니다.

```
>>> int("32")
```

```
32
```

```
>>> int(17.3)
```

```
17
```

```
>>> float(17)
```

```
17.0
```

```
>>> float("3.1415")
```

```
3.1415
```

```
>>> str(17) + " " + str(3.1415)
```

```
'17 3.1415'
```

```
>>> complex(17)
```

```
(17 + 0j)
```

수학 함수를 사용하려면, Python에서 `math` 모듈을 사용하고 싶다고 선언해야 합니다.

```
import math
degrees = 45
radians = degrees / 360.0 * 2 * math.pi
print(math.sin(radians))
print(math.sqrt(2) / 2)
```

함수들을 자주 사용한다면 함수에 더 짧은 이름을 지어줄 수 있습니다.

```
import math
sin = math.sin
pi = math.pi
radians = degrees / 360.0 * 2 * pi
print(sin(radians))
```

# 매개 변수를 사용한 함수 정의

함수는 인자들을 가리키는 변수들을 이용하여 정의됩니다.  
이 이름들은 매개 변수 (Parameter)라고 부릅니다.

```
def compute_interest(amount, rate, years):
```

함수 내부에서 매개 변수는 다른 변수와 동일하게 사용됩니다.

```
    value = amount * (1 + rate/100.0) ** years
```

함수의 결과 계산이 끝나면 결과값을 반환 (Return)해야 합니다.  
함수는 함수가 반환되는 시점에 종료되며, 함수의 결과는 반환값을 통해 전달됩니다.

```
    return value
```

이제 우리는 같은 함수를 다른 인자 값들을 이용하여 부를 수 있습니다.

```
>>> s1 = compute_interest(200, 7, 1)
>>> s2 = compute_interest(500, 1, 20)
```

## 여러 반환문을 가진 함수

다음 함수는 절대값을 계산하는 함수입니다 (내장 함수 `abs`와 동일한 기능):

```
def absolute(x):  
    if x < 0:  
        return -x  
    else:  
        return x
```

같은 함수를 다음처럼 쓸 수도 있습니다.

```
def absolute(x):  
    if x < 0:  
        return -x  
    return x
```

하지만, 다음처럼 쓰면 안 됩니다.

```
def absolute(x):  
    if x < 0:  
        return -x  
    if x > 0 :  
        return x
```

# 논리값을 반환하는 함수

조건을 검사해서 **True**나 **False**를 반환하는 함수는 조건함수(Predicate)라고 부릅니다.

```
# is integer a divisible by b?
```

```
def is_divisible(a, b):
```

```
    if a % b == 0:
```

```
        return True
```

```
    else:
```

```
        return False
```

조건 함수는 **if** 문이나 **while** 문에서 바로 사용할 수 있습니다.

```
if is_divisible(x, y):
```

```
    print('x is divisible by y')
```

위 함수는 다음처럼 간단하게 정의할 수도 있습니다.

```
def is_divisible(a, b):
```

```
    return a % b == 0
```

## 결과값이 없는 함수

지금까지 사용한 함수들 중에는 반환문이 없는 함수들도 많이 있습니다.

```
def turn_right():  
    for i in range(3):  
        hubo.turn_left()
```

반환문이 없는 함수는 자동으로 **None**을 반환합니다.

```
>>> s = turn_right()  
>>> print(s)
```


None



## 함수 호출

함수가 호출되면, 호출될 때의 함수 인자 (argument) 들은 함수의 매개 변수 (parameter)로 대입됩니다.

```
def print_twice(text):  
    print(text)  
    print(text)
```



함수 호출에 사용하는 인자의 수는 함수의 매개 변수의 수와 동일해야 합니다.

```
>>> print_twice("I love CS101")
```

```
I love CS101
```

```
I love CS101
```

```
>>> print_twice(math.pi)
```

```
3.14159265359
```

```
3.14159265359
```

## 여러 값 반환하기

함수는 하나의 값만 반환할 수 있습니다.

하지만 함수는 튜플을 반환할 수도 있습니다.

Python은 함수가 여러 값을 반환할 때, 자동으로 이 값들을 튜플로 만들어 반환합니다.

```
def student():  
    name = "Hong, Gildong"  
    id = 20101234  
    return name, id
```

튜플로 된 반환값은 다음처럼 바로 풀어낼 수 있습니다.

```
>>> name, id = student()
```

```
>>> print(name)
```

```
"Hong, Gildong"
```

```
>>> print(id)
```

```
20101234
```

**input** 함수는 메시지를 출력하고, 키보드를 통한 사용자의 문자열 입력을 기다립니다. 사용자가 Enter 키를 입력하면, 사용자가 입력한 전체 문자열이 반환됩니다.

```
name = input("What is your name? ")  
print("Welcome to CS101, " + name)
```

숫자 입력이 필요하다면, 문자열을 숫자로 변환해야 합니다.

```
raw_n = input("Enter a positive integer> ")  
n = int(raw_n)  
for i in range(n):  
    print("*" * i)
```

# 정리 및 연습

본 강의 학습 목표:

- 매개 변수와 반환값을 가진 함수를 이해하고 사용할 수 있다.

다음 강의 학습 목표:

- 함수를 사용하여 휴보 로봇이 비퍼들을 쥘 수 있는 프로그램을 모듈러하게 작성할 수 있다.
- 함수를 사용하여 색의 밝기 측정 및 디지털 사진을 흑백 모드로 변환할 수 있다.