

# 인공지능프로그래밍

게임콘텐츠학과 박경수

<https://github.com/ggorr/Machine-Learning/tree/master/Python>

# 8장. 함수

- ▶ 8.1 함수와 매개변수
- ▶ 8.2 지역 변수
- ▶ 8.3 global 문
- ▶ 8.4 기본 인수값
- ▶ 8.5 키워드 인수
- ▶ 8.6 VarArgs 매개 변수
- ▶ 8.7 return 문
- ▶ 8.8 DocString

# 8.1 함수와 매개 변수

- ▶ 함수는 def 로 정의

```
def print_max(a, b):  
    if a > b:  
        print(a, 'is maximum')  
    elif a == b:  
        print(a, 'is equal to', b)  
    else:  
        print(b, 'is maximum')  
  
print_max(3, 4)  
  
x = 5  
y = 7  
print_max(x, y)
```

```
4 is maximum  
7 is maximum
```

## 8.2 지역 변수

- ▶ 함수에서 정의된 변수는 함수 안에서만 사용 가능

```
x = 50
def func(x):
    print('x is', x)
    x = 2
    print('x is changed to', x)
```

```
func(x)
print('x is still', x)
```

```
x is 50
x is changed to 2
x is still 50
```

## 8.3 global

- ▶ 외부에서 정의된 변수를 사용

```
x = 50

def func(): # x를 제거
    global x
    print('x is', x)
    x = 2
    print('x is changed to', x)

func() # x를 제거
print('x is still', x)
```

```
x is 50
x is changed to 2
x is still 2
```

## 8.4 기본 인수값

### ▶ 기본 인수값

```
def say(message, times=1):  
    print(message * times)  
  
say('hello') # times = 1  
say('world', 3) # times = 3
```

```
hello  
worldworldworld
```

- ▶ 상수이어야 한다
- ▶ 기본 인수값이 없는 매개 변수보다 뒤에 와야 한다
  - ▶ `def say(times=1, message):` # 에러
- ▶ 인수를 대입하면 기본 인수값은 무시된다
  - ▶ `say('world', 3)` # `times=1`이 무시됨

## 8.5 키워드 인수

- ▶ 순서에 상관없이 매개 변수 이름으로 값을 준다

```
def func(a, b=5, c=10):  
    print('a is', a, 'and b is', b, 'and c is', c)
```

```
func(3, 7) # a=3, b=7, c=10
```

```
func(25, c=24) # a=25, b=5, c=24
```

```
func(c=50, a=100) # a=100, b=5, c=50
```

## 8.6. VarArgs 매개 변수

```
def f(x, *y):  
    print('x:', x)  
    print('y: ', end='')  
    for z in y:  
        print(z, end=' ')  
    print()  
    print('-----')  
  
f(1)  
f(1, 2)  
f(1, 2, 3)
```

```
x: 1  
y:  
-----  
x: 1  
y: 2  
-----  
x: 1  
y: 2 3  
-----
```



```
def g(x, *y, **z):
    print('x:', x)

    print('y: ', end='')
    for yi in y:
        print(yi, end=' ')
    print()

    print('z:')
    for key in z:
        print(f'    {key}: {z[key]}')
    print('-----')

g(1)
g(1, 2, 3)
g(1, 2, 3, a=5, b=99)
```

```
x: 1
y:
z:
-----
x: 1
y: 2 3
z:
-----
x: 1
y: 2 3
z:
    a: 5
    b: 99
-----
```

## 8.7 return 문

```
def add(x, y):  
    return x + y  
  
def maximum(x, y):  
    if x > y:  
        return x  
    elif x == y:  
        return 'The numbers are equal'  
    else:  
        return y  
  
print(add(5, 6))  
print(maximum(2, 3))
```

## 8.8 DocString

- ▶ '''와 ''' 사이에 함수에 대한 설명을 적는다

```
def print_max(x, y):  
    '''Prints the maximum of two numbers.  
  
    The two values must be integers.'''
```

- ▶ 연습문제 8.1 함수 `operate()`를 작성하시오. 이 함수는 다음과 같은 값을 반환한다

```
operate(5, 2) : 5 + 2를 return  
operate(5, 2, op='+') : 5 + 2를 return  
operate(5, 2, op='*') : 5 * 2를 return
```

- ▶ 연습문제 8.2 인수를 모두 더하여 반환(`return`)하는 함수 `add()`를 작성하시오

```
add(1) : 1을 return  
add(1, 2) : 1 + 2를 return  
add(1, 2, 3) : 1 + 2 + 3을 return
```

- ▶ 연습문제 8.3 `n`이 소수(质数)인지 판별하는 함수 `is_prime()`을 작성하시오

```
is_prime(3) : True를 반환  
is_prime(21) : False를 반환
```