

함수

함수

❖ 함수

- 특정 목적을 가진 코드의 집합이며 독립적으로 호출될 수 있는 것
- def로 선언
 - 인자와 리턴값이 있는 함수 선언

```
def 함수이름(인자1, 인자2, ...):  
    코드들  
    return 결과값
```

변수 = 함수이름(인자1, 인자2, ...)

- 리턴값은 있지만 인자가 없는 함수 선언

```
def 함수이름():  
    코드들  
    return 결과값
```

함수이름(인자1, 인자2, ...)

- 인자와 리턴값이 없는 함수 선언

```
def 함수이름():  
    코드들  
    return (또는 생략)
```

함수이름()

❖ 함수 선언 및 호출

```
def add_number(n1, n2):  
    ret = n1 + n2  
    return ret  
  
def add_txt(t1, t2):  
    print(t1 + t2)  
  
ans = add_number(10, 15)  
print(ans)  
  
text1 = '대한민국~'  
text2 = '만세!!!!'  
add_txt(text1, text2)
```

```
25  
대한민국~만세!!!!
```

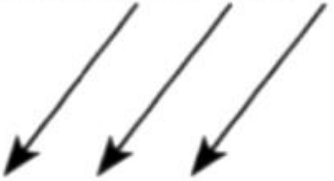
함수 인자

❖ 함수 인자

- 인자의 위치에 실제 값을 대입하여 함수를 호출하면 인자 순서에 대응하는 값을 함수 코드에 대입하여 실행

함수호출부분 **answer = func(10, 20, 30):**

함수선언부분 **def func(p1, p2, p3):**



❖ 함수 인자 전달 방식

- 기본 인자 이용: 기본 인자를 이용해 값을 전달
 - 디폴트값 지정
 - 인자 리스트의 마지막에 위치
- 키워드 인자 이용: 인자 이름을 이용해 값을 전달
- 가변 인자 이용: 가변 인자 튜플을 이용해 값을 전달
- 미정 키워드 인자 이용: 미정 키워드 인자 사전을 이용해 값을 전달

❖ 함수 인자 전달 방식

- 기본 인자 이용

```
def add_txt(t1, t2='파이썬')  
    print(t1 + ':' + t2)  
  
add_txt('베스트')
```

- 키워드 인자 이용


- 호출하는 측에서 지정

```
add_txt(t2='대한민국', t1='1등')
```

- 인자의 순서는 무시됨

함수호출부분 **answer = func(p2=10, p1=20, p3=30):**

함수선언부분 **def func(p1, p2, p3):**



❖ 함수 인자 전달 방식

- 가변 인자(튜플) 이용
 - 인자 앞에 *

```
def func1(*args):  
    print(args)
```

```
func1()  
func1(3, 5, 1, 5)
```

```
()  
(3, 5, 1, 5)
```

❖ 함수 인자 전달 방식

- 미정 키워드 인자(사전) 이용
 - 다른 인자와 같이 사용
 - 미정 키워드 인자 앞에 **

```
def func2(width, height, **kwargs):  
    print(kwargs)
```

```
func2(10, 20)  
func2(10, 20, depth=50, color='blue')
```

```
{}  
{'depth': 50, 'color': 'blue'}
```


지역변수(local)와 전역변수(global)

❖ 변수의 유효 범위에 따른 변수 구분

- 지역변수
 - 함수 내에서만 유효한 변수
 - 함수 내에서 선언된 변수
 - 함수 내에서 선언하지만 전역변수로 사용하고 싶은 경우 : `global` 키워드 사용
- 전역변수
 - 코드 전반에 걸쳐 유효한 변수
 - 함수 밖에서 선언된 변수

지역변수(local)와 전역변수(global)

❖ 지역변수, 전역변수

```
param = 10
strdata = '전역변수'

def func1():
    strdata = '지역변수'
    print(strdata)

def func2(param):
    param=1

def func3():
    # global param
    param = 50

func1()
print(strdata)

print(param)
func2(param)
print(param)

func3()
print(param)
```

```
지역변수
전역변수
10
10
10
```

함수 리턴값

❖ return

- 함수가 수행되고 그 결과를 리턴하고 함수 실행 종료
- return 만 사용하는 경우 결과 리턴하지 않고 함수 실행 종료

```
def reverse(x, y, z):  
    return z, y, x  
  
ret = reverse(1, 2, 3)  
print(ret)  
  
r1, r2, r3 = reverse('a', 'b', 'c')  
print(r1); print(r2); print(r3)
```

```
(3, 2, 1)  
c  
b  
a
```

Call by Value

❖ Call by Value

- 프리미티브 타입은 인자로 전달될 때 값이 복사됨
- 함수 내에서 값을 바꿔도 호출측에서 변경되지 않음

```
def swap(n1, n2):  
    t = n1  
    n1 = n2  
    n2 = t  
    print(n1)  
    print(n2)  
  
number1 = 10  
number2 = 20  
swap(number1, number2)  
  
print(number1)  
print(number2)
```

```
20    # n1  
10    # n2  
10    # number1  
20    # number2
```

Call by Reference

❖ Call by Reference

- 프리미티브 타입 이외의 타입은 참조 값을 전달
 - 요소의 값을 변경하면 호출한 쪽에서도 변경되어 있음

```
def swap2(listdata):  
    t = listdata[0]  
    listdata[0] = listdata[1]  
    listdata[1] = t  
    print(listdata)
```

```
numbers = [10, 20]  
print(numbers)  
swap2(numbers)  
print(numbers)
```

```
[10, 20]  
[20, 10]  
[20, 10]
```