

Python语言基础与应用

计算和控制流 / 代码组织：函数的参数

陈斌 北京大学 gischen@pku.edu.cn



代码组织：函数的参数

- › 形式参数与实际参数
- › 定义函数的参数
- › 固定参数/可变参数
- › 调用函数的参数
- › 位置参数/关键字参数

形式参数与实际参数

› 参数：传入到函数的值

当调用含参数的函数时，这些参数的值会被复制给函数中的对应参数

› 形式参数(parameter)

函数创建和定义过程中，函数名后面括号里的参数

› 实际参数(argument)

函数在调用过程中传入的参数

形式参数与实际参数

- › 形式参数只是代表一个位置、一个变量名
- › 实际参数是一个具体内容，赋值到变量的值

```
>>> def add(a, b):  
        return a + b  
  
>>> add(1, 2)  
3
```

定义函数的参数

- › 定义函数时，参数可以有两种
- › 一种是在参数表中写明参数名key的参数，固定了顺序和数量的**固定参数**
- › 一种是定义时还不知道会有多少参数传入的**可变参数**

```
def func(key1, key2, key3...):
```

```
def func(key1, key2=value2...):
```

```
def func(*args): #不带key的多个参数
```

```
def func(**kwargs): #key=val形式的多个参数
```

固定参数

```
16 def func_test(key1, key2, key3=23):  
17     print("k1=%s,k2=%s,k3=%s" % (key1, key2, key3))  
18  
19  
20     print("====func_test")  
21     # 没有传入key3, 用了缺省值  
22     func_test('v1', 'v2')  
23     # 传入了key3  
24     func_test('ab', 'cd', 768)  
25     # 使用参数名称就可以不管顺序  
26     func_test(key2='KK', key1='K')
```

```
====func_test  
k1=v1,k2=v2,k3=23  
k1=ab,k2=cd,k3=768  
k1=K,k2=KK,k3=23
```

可变参数

```
29 # 可以随意传入0个或多个无名参数
30 def func_test2(*args):
31     for arg, i in zip(args, range(len(args))):
32         print("arg%d=%s" % (i, arg))
33
34
35 print("====func_test2")
36 func_test2(12, 34, 'abcd', True)
```

```
====func_test2
arg0=12
arg1=34
arg2=abcd
arg3=True
```

可变参数

```
39 # 可以随意传入0个或多个带名参数
40 def func_test3(**kwargs):
41     for key, val in kwargs.items():
42         print("%s=%s" % (key, val))
43
44
45 print("====func_test3")
46 func_test3(myname="Tom", sep="comma", age=23)
```

```
====func_test3
sep=comma
age=23
myname=Tom
```


调用函数的参数

› 调用函数的时候，可以传进两种参数

› 一种是没有名字的位置参数

`func(arg1, arg2, arg3...)`

会按照前后顺序对应到函数参数传入

› 一种是带key的关键字参数

`func(key1=arg1, key2=arg2...)`

由于指定了key，可以不按照顺序对应

› 如果混用，所有位置参数必须在前，关键字参数必须在后

位置参数 / 关键字参数

```
16 def func_test(key1, key2, key3=23):
17     print("k1=%s,k2=%s,k3=%s" % (key1, key2, key3))
18
19
20 print("====func_test")
21 # 没有传入key3, 用了缺省值
22 func_test('v1', 'v2')
23 # 传入了key3
24 func_test('ab', 'cd', 768)
25 # 使用参数名称就可以不管顺序
26 func_test(key2='KK', key1='K')
```

```
====func_test
k1=v1,k2=v2,k3=23
k1=ab,k2=cd,k3=768
k1=K,k2=KK,k3=23
```