

函数参数详解

一、缺省参数

调用函数时，缺省参数的值如果没有传入，则取默认值。

下例会打印默认的age，如果age没有被传入：

```
def printinfo(name, age=35):  
    # 打印任何传入的字符串  
    print("name: %s" % name)  
    print("age %d" % age)  
  
# 调用printinfo函数  
printinfo(name="miki") # 在函数执行过程中 age取默认值35  
printinfo(age=9 ,name="miki")
```

以上实例输出结果：

```
name: miki  
age: 35  
name: miki  
age: 9
```

总结：

- 在形参中默认有值的参数，称之为缺省参数
- 注意：带有默认值的参数一定要位于参数列表的最后面

```
>>> def printinfo(name, age=35, sex):  
...     print name  
...  
File "<stdin>", line 1  
SyntaxError: non-default argument follows default argument
```

二、不定长参数

有时可能需要一个函数能处理比当初声明时更多的参数，这些参数叫做不定长参数，声明时不会命名。

基本语法如下：

```
def functionname([formal_args,] *args, **kwargs):
    """函数_文档字符串"""
    function_suite
    return [expression]
```

注意：

- 加了星号 (*) 的变量args会存放所有未命名的变量参数，args为元组
- 而加**的变量kwargs会存放命名参数，即形如key=value的参数，kwargs为字典。

```
def test(a, b, *args, **kwargs):
    "函数在声明时，需要两个参数"
    print('a={},b={},args={},kwargs={}'.format(a,b,args,kwargs))

test(2, 3, '你好', 'hi', 'how do you do', name="zhangsan", age=18)
# a=2,b=3,args=('你好', 'hi', 'how do you do'),kwargs={'name': 'zhangsan', 'age': 18}

b = ('hi', '大家好', '今天天气真好')
d = {'name': "zhangsan", "age": 19}

# 注意，在传入参数时的星号问题。
test(10, 20, *b, **d)
# a=10,b=20,args=('hi', '大家好', '今天天气真好'),kwargs={'name': 'zhangsan', 'age': 19}

# 如果在传值时，不使用星号，会把后面的参数当做 args
test(10,20,b,d)
# a=10,b=20,args=((('hi', '大家好', '今天天气真好'), {'name': 'zhangsan', 'age': 19})),kwargs={}

# Python 人工智能学院
```

三、缺省参数在*args后面

```
def sum_nums_3(a, *args, b=22, c=33, **kwargs):
    print(a)
    print(b)
    print(c)
    print(args)
    print(kwargs)

sum_nums_3(100, 200, 300, 400, 500, 600, 700, b=1, c=2, mm=800, nn=900)
```

说明：

- 如果很多个值都是不定长参数，那么这种情况下，可以将缺省参数放到 *args* 的后面， 但如果有的话， **kwargs* 必须是最后的

千锋Python人工智能学院