

# Python 函数参数

任何编程语言函数都是非常重要的一部分，而在进行函数调用时，了解函数的参数传递方式是非常有必要的。Python中支持哪些传参方式呢？

Python中的传参方式是比较灵活的，主要包括以下六种：

- 按照位置传参
- 按照关键字传参
- 默认参数
- 元组传参
- 字典传参
- 综合传参

## 一、按照位置传参

顾名思义，实参与形参数目一致，并且是按照位置一一对应的。

```
def func(a, b):  
    return a+b*2
```

当调用函数func(1,2)时，则按照a=1，b=2的方式形参与实参对应，返回结果为5。

## 二、按照关键字传参

这种方式在调用函数时既传实参，也传形参。因而，不需要根据位置对应了，位置可以颠倒：

```
def func(a, b):  
    return a+b*2
```

当调用函数func(b=1,a=2)时，则按照b=1，a=2的方式形参与实参对应，匹配的方式是关键字而非位置，返回结果为4。

## 三、默认参数

默认参数必须放在前两类参数之后，在调用函数的时候可以传参，也可以不传。不传则采用默认的参数值。

```
def func(a, b, c = 3):  
    return a+b*2+c
```

其中，c为默认参数。当通过func(1,2)调用函数时，c的取值为默认值3，函数的执行结果为8；当通过func(1,2,5)调用函数时，c的取值为实参值5，函数的执行结果为10。

## 四、元组传参

传入的参数以元组的形式呈现，长度不限，可以通过元组的访问方式依次访问各个入参：

```
def func(*args):  
    for eacharg in args:  
        print ('tuple arg:', eacharg)
```

调用方式如下：func('I', 'Love', 'PythonSomething')

调用结果为：

```
tuple arg: I  
tuple arg: Love  
tuple arg: PythonSomething
```

## 五、字典传参

传入的参数以字典的形式呈现，长度不限，可以通过字典的访问方式依次访问各个入参：

```
def func(**kwargs):  
    for eachkwarg in kwargs.keys():  
        print('Dict Arg', eachkwarg, ':', kwargs[eachkwarg])
```

调用方式如下：

func(English\_name = 'PythonSomething', Chinese\_name= 'Python那些事')

调用结果为：

```
Dict Arg English_name : PythonSomething  
Dict Arg Chinese_name : Python那些事
```