

Abstract

-->Means 告诉计算机做什么而不是怎么去做

callable() 函数返回 true/false 是否能调用

文档字符串--docstring

所有的函数都返回值，如果你没告诉他们返回什么，就会返回None

参数

在函数内部重新关联参数（即给它赋值）时，函数外部的变量不受影响。

参数出存在局部作用域中

对序列执行切片操作时，返回的切片都是副本

dict中，key的排列顺序是不定的，每次打印都会有可能不同

类型：

位置参数、关键字参数、加*意思是无限数量参数

收集参数：

将参数收集到tuple或dict中

带星号的参数会返回一个tuple。

```
例如 def func(*a):  
    print (a)  
    func ('abcd')  
>> ('abcd',)  
    func (1,2,3)  
>> (1,2,3)
```

带星号的参数可以放其他位置，例如 def func(a, *b, c) 调用时赋予c需要用关键字

带星号的参数不会收集关键字参数，如果收集的话，要用两个星号

带两个星号的是返回一个dict

```
例如 def func(**a):  
    print(a)  
    func(a=1,b=2,c=3)  
>> {'a'=1,'b'=2, 'c'=3}
```

分配参数：

将 为dict或tuple的参数 分配到函数中去

例如

```
a=(1,2)  
def pr(x, y):  
    print(x, y)  
pr(*a)  
>> 1 2
```

作用域：

执行赋值语句时，例如a=1，实际上是创建了一个“看不见的dict”，将x=1放进去。可以用一个叫vars的内建函数看这个dict。

例如：

```
x=1  
b=vars()  
print(b['x'])
```

```
>>> 1
```

这个看不见的dict，就叫 **命名空间 (Namespace)** 或 **作用域 (scope)**

除了全局作用域，每个函数调用都会创建一个，在函数里用的变量叫**局部变量**，是在内部作用域中执行的，不会影响到**外部作用域 (全局变量)**。（慎用全局变量！容易出bug！）

函数**globals**返回全局变量的dict，**locals**返回局部变量的dict，如果在函数内一个变量与全局变量同名，在函数内全局的会被遮盖，要访问就要用 例如 `globals()['parameter_name']` 或者在函数里定义变量时, `global parameter_name`，然后使用时会调用全局变量。

python函数可以嵌套，作用域也会被嵌套

递归一般比循环慢，但是可读性高。

bisect module 提供了标准的二分查找实现

每日小技巧：

1.

```
a=[1,2]
```

```
print(a) >> [1,2]
```

```
print(*a) >> 1 2
```