

Python 函数

函数是“一系列命令的集合”，我们可以通过调用函数来自动执行某一系列命令。虽然经常性地出现于文章中的`print()`是被录入在Python的标准库中的函数，但是，程序员亦可创建自己的函数。

如果想要定义函数，则需要以“`def 函数名():`”的格式为开头编写代码。在这之下的一个模块就是一个函数的范围。Python的模块就如前文中提到的，是根据缩进的等级来进行区分的。

同时，对于函数也需要设定参数，函数可以根据参数的值来执行各种指令。在Python中，可以通过使用列表或者双精度浮点型变量来灵活指定参数。同时，也可以将函数运行的结果作为返回值返回。函数可以多次调用。

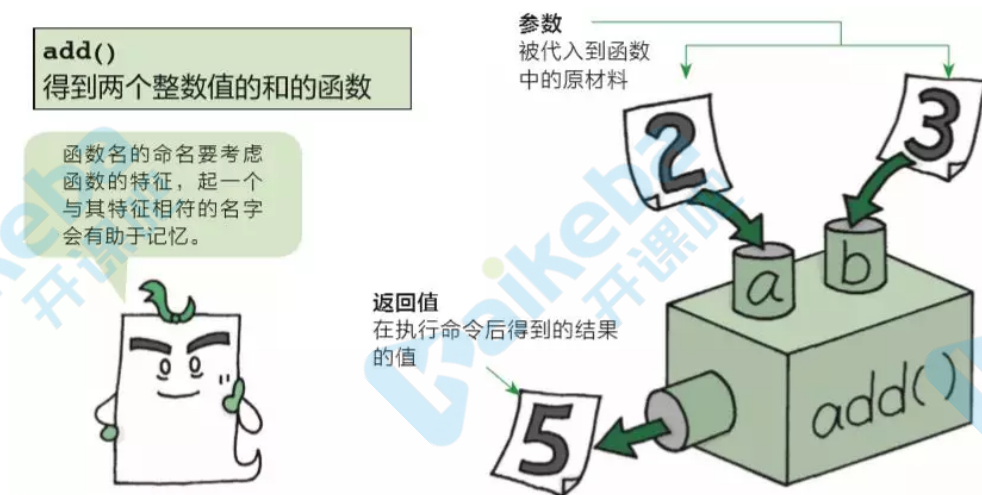
所以，如果设计出出色的函数，那么在编写复杂的程序时可以将行文简洁地记述出来。

一、函数的定义

1、所谓函数？

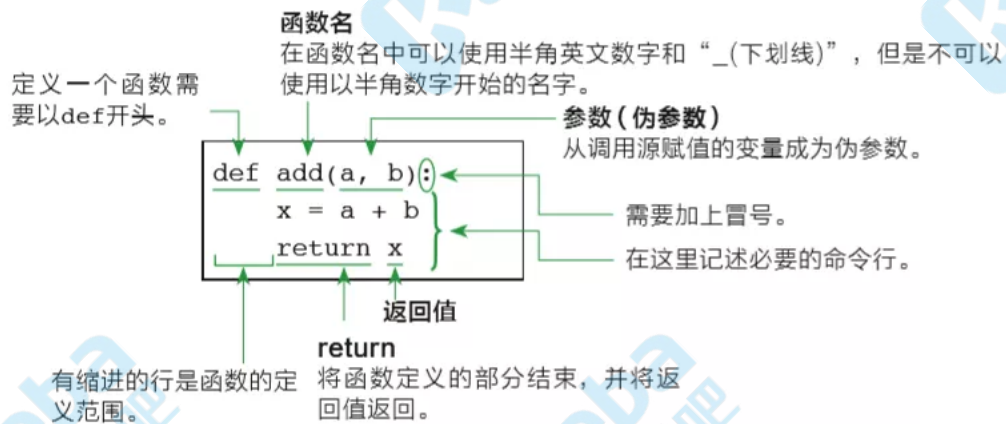
函数就是对程序给出的值按照既定的处理指令执行命令并将结果返回的黑箱。

其中函数所处理的值被称为参数、函数所返回的结果的值被称为返回值。

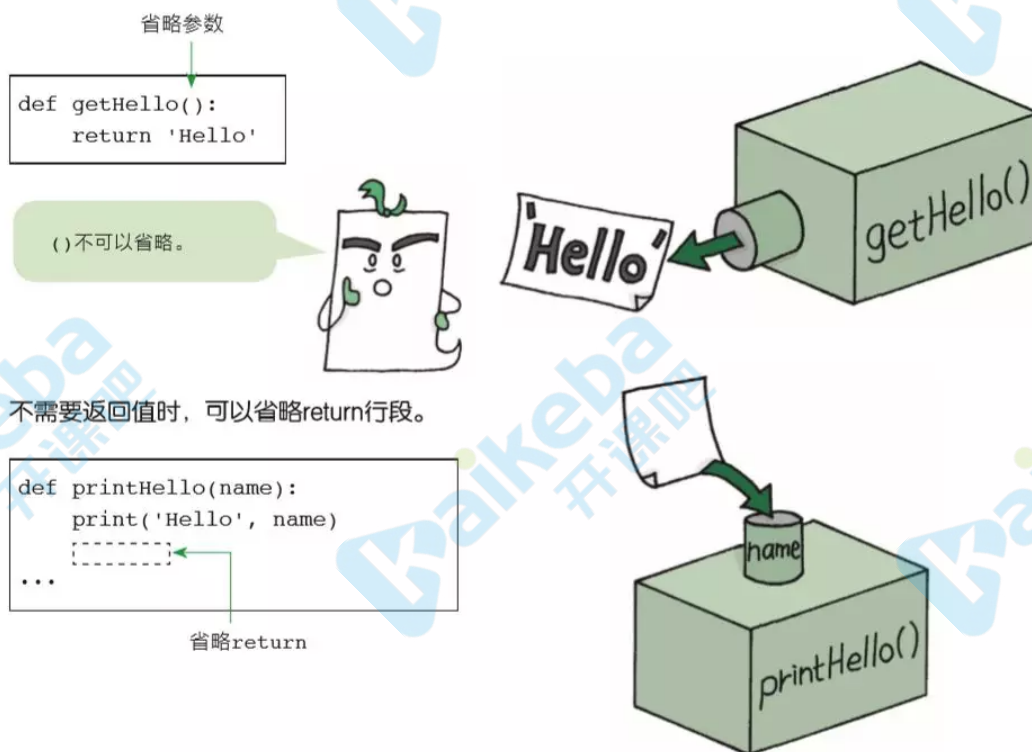


2、函数的定义

将上文中的函数按Python的方式记述，则会变为如下格式。像这样将函数的功能记述出来的方式被称为定义函数。



参数在不必要的时候可以省略



如果要做一个什么命令都不执行的函数，则用pass。

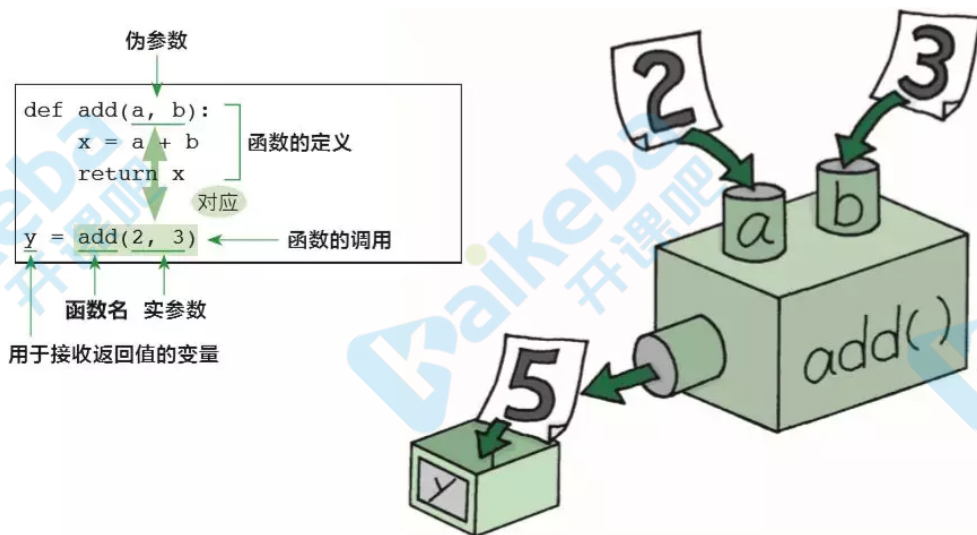
```
def nowork():  
    pass
```

在函数中存在pass命令，则整个函数的内容都将变得无效。

二、函数的调用

1、调用的基本形式（位置参数）

函数可以用如下的方式调用。在调用时使用的参数被称为实参，从原则上说要引入和定义时所指定的参数个数一样的数量



如果是不具备参数的函数，则是以下格式。



2、使用了关键字的参数指定（关键参数）

可以通过使用伪参数的名字来指定其对应的实参。

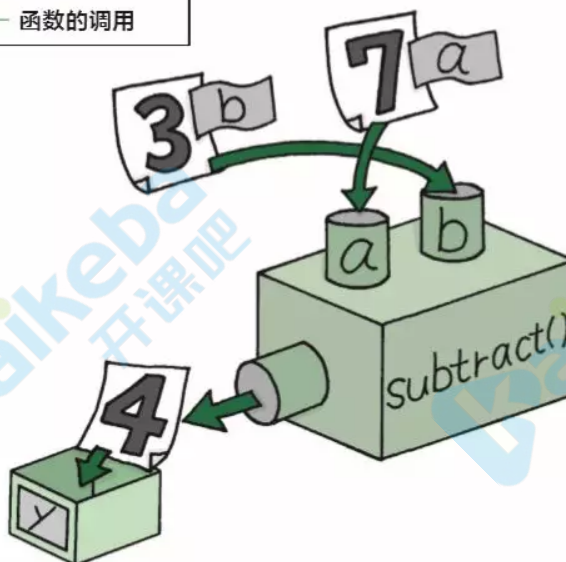
```
def subtract(a, b):
    x = a - b
    return x

y = subtract(b = 3, a = 7)
```

函数的定义

函数的调用

实参的顺序不需要与伪参数一致。



```
def calc(calctype, a, b, c):
    if calctype == '和':
        x = a + b + c
        s = '{}+{}+{}={}'.format(a, b, c, x)
    elif calctype == '积':
        x = a * b * c
        s = '{}*{}*{}={}'.format(a, b, c, x)
    else:
        s = '???'
    return calctype + ':' + s

print( calc('和', a=5, b=8, c=3) )
print( calc('积', a=5, b=8, c=3) )
print( calc('差', a=5, b=8, c=3) )
```

运行结果

最初的参数用位置，其余的参数用关键字来指定。

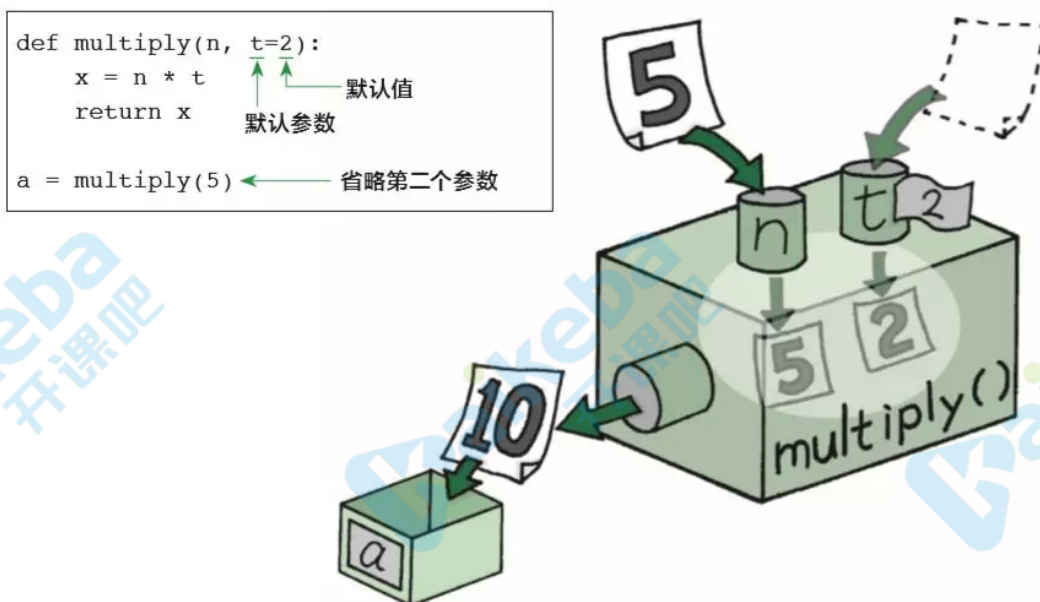


和 : 5+8+3=16
积 : 5*8*3=120
差 : ???

三、函数的使用技巧

1、默认函数

可以给参数规定一个默认值的函数，指定了默认值的函数在被调用时，参数可以省略。



默认参数可以从右侧的参数开始设置多个，但是不可以将中间的某个参数作为默认参数。



```
def func(a, b=1, c=0)  
    :
```



```
def func(a, b=0, c)  
    :
```

2、函数对象

在函数名的末尾不添加(), 只写名称的格式所表示的是函数本身。我们将其称之为函数对象，可以像值一样将其代入到变量中去。

```
def printHello(name):  
    print('Hello', name)
```

函数的定义

```
func = printHello  
func('Shiori')
```

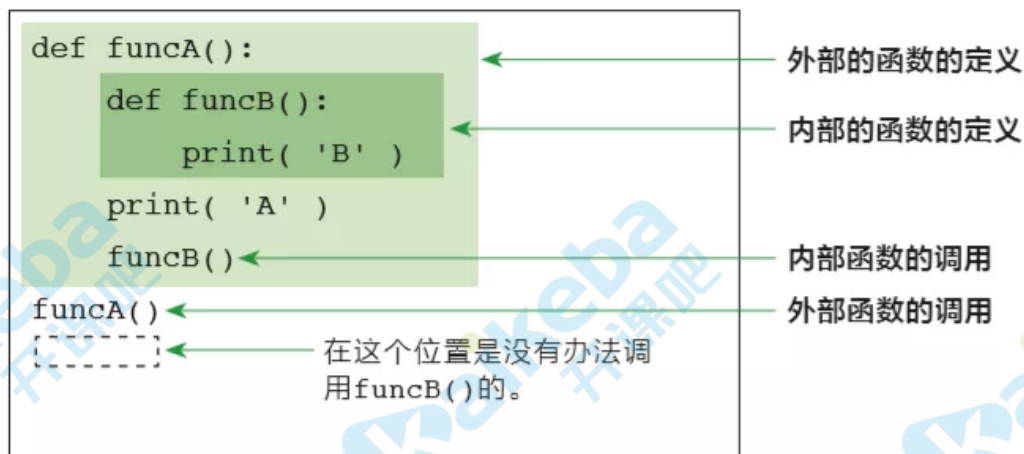
将函数printHello代入到func
func可以作为函数使用

运行结果

Hello Shiori

3、函数的嵌套

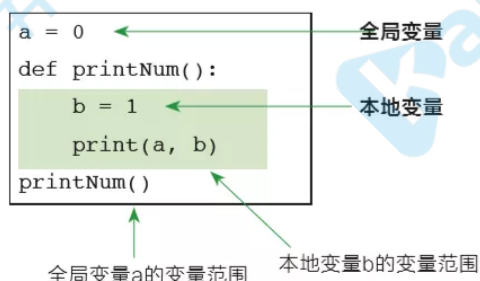
我们可以定义函数中的函数。内部的函数被称为本地函数，只可以在特定的函数中被使用的函数都可以被认为是本地函数。



四、变量范围

1、本地变量和全局变量

在函数内部使用的变量为本地变量，本地变量只在函数内部有效。与之相对，在函数外部使用的变量为全局变量，全局变量可以在任意函数中调用。



本地变量的变量范围在函数内部。全局变量的变量范围是整个文件内部。



运行结果

0 1

在函数被用于嵌套中时，只在其每一单次的函数调用内有效。

```

a = 0
def funcA():
    b = 1
    def funcB():
        c = 2
        print(a, b, c)
    funcB()
funcA()

```

变量a的变量范围
变量b的变量范围
变量c的变量范围

运行结果

0 1 2

2、非本地变量的更改

在函数中想要改变其外部的变量的时候需要注意几个要素，像下图那样简单地记述代入命令是无法改变外部变量的。

```

a = 0
def funcA():
    a = 1
    b = 1
    def funcB():
        b = 2
        c = 2
        print(a, b, c)
    funcB()
    print(b)
funcA()
print(a)

```

别的变量
funcA() 的本地变量
funcB() 的本地变量
还是原来的1
还是原来的0

如果将数值直接代入的话，会变成全新的本地函数。



运行结果

1 2 2
1
0

如果想要更改函数外部的变量的话，需要使用global或者nonlocal命令来声明变量并非本地变量。

```

a = 0
def funcA():
    global a
    a = 1
    b = 1
    def funcB():
        nonlocal b
        b = 2
        c = 2
        print(a, b, c)
    funcB()
    print(b)
funcA()
print(a)

```

声明这个变量是全局变量
声明这个变量不是本地变量

运行结果

1 2 2
2
1

