你可以定义一个由自己想要功能的函数，以下是简单的规则：

- 函数代码块以 **def** 关键词开头，后接函数标识符名称和圆括号 **()**。
- 任何传入参数和自变量必须放在圆括号中间，圆括号之间可以用于定义参数。
- 函数的第一行语句可以选择性地使用文档字符串—用于存放函数说明。
- 函数内容以冒号起始，并且缩进。
- **return [表达式] 结束函数，选择性地返回一个值给调用方。不带表达式的return相当于返回None。**

函数定义

```
def myMethod(x,y):
    return x**y
```

函数调用

```
myMethod(x,y)
```

按值传递参数和按引用传递参数

在 Python 中，所有参数（变量）都是按引用传递。如果你在函数里修改了参数，那么在调用这个函数的函数里，原始的参数也被改变了。

默认参数

调用函数时，如果没有传递参数，则会使用默认参数。以下实例中如果没有传入 age 参数，则使用默认值：

return语句

return [表达式] 语句用于退出函数，选择性地向调用方返回一个表达式。不带参数值的return语句返回None。之前的例子都没有示范如何返回数值，以下实例演示了 return 语句的用法：

变量作用域

Pyhton 中，程序的变量并不是在哪个位置都可以访问的，访问权限决定于这个变量是在哪里赋值的。
变量的作用域决定了在哪一部分程序可以访问哪个特定的变量名称。两种最基本的变量作用域如下：

- 全局变量
- 局部变量

全局变量和局部变量

定义在函数内部的变量拥有一个局部作用域，定义在函数外的拥有全局作用域。

局部变量只能在其被声明的函数内部访问，而全局变量可以在整个程序范围内访问。调用函数时，所有在函数内声明的变量名称都将被加入到作用域中。如下实例：

```
#!/usr/bin/python3

total = 0; # 这是一个全局变量
# 可写函数说明
def sum( arg1, arg2 ):
    #返回2个参数的和."
    total = arg1 + arg2; # total在这里是局部变量.
    print ("函数内是局部变量 : ", total)
    return total;

#调用sum函数
sum( 10, 20 );
print ("函数外是全局变量 : ", total)
```

以上实例输出结果：

```
函数内是局部变量 :  30
函数外是全局变量 :  0
```