【白话python连载（7）】python的函数和模块--动手打包共享

* 从小入手-python小应用
* python的函数详细介绍

函数的定义

python**函数标准的定义是**：为了最大程度的重用代码和最小化代码冗余而提供的基本程序结构；以便将复杂的程序系统分解为可管理的若干部分。

在python中定义函数，借助关键字def进行定义，自定义一个函数名称。并且指定函数所需的相关参数，最终在函数体中，完成特定的一个功能。同时也可以定义该函数的返回值，供其他功能模块使用，如果不定义函数返回值，则默认返回None。例如我们之前实现的求一个数绝对值的函数体（在调用过程中即可直接获得绝对值结果），计算九九乘法表的函数等。通过python的语法格式缩进表示函数头和函数体的关系，在函数头中不能缺少：。

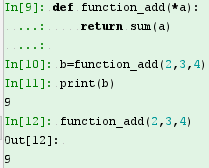
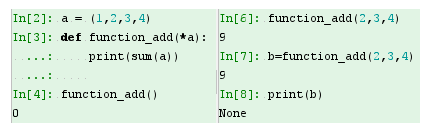
```

def function\_name(parameter)：

function\_content

return something

```

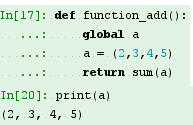
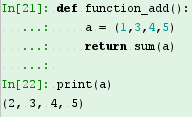


在实际编写一个函数时，若未完全设计好函数的功能， 可以优先利用pass语句做占位符，保证函数语义的完整性，以便之后再完善补充函数功能。

全局变量和局部变量

函数中关于变量的作用域问题，python程序中的变量按照作用域分，可以分为全局变量和局部变量，如果一个程序中的变量定义在函数体中，则该变量为局部变量，如果定于在主程序中，则为全局变量，一般而言局部变量的优先级高于全局变量的优先级，局部变量的值会覆盖全局变量的值。因此在编写程序过程中，为了不污染其他函数中的变量

在函数中，如果想给全局变量赋值，则需要用关键字global对变量进行声明。在函数体中使用global对a变量进行重新赋值后，a变量在全局范围将进行更新，否则a变量的值将保持不变，如我们想将a变量的值进行还原，不用global关键字，则不会改变a变量的值。

函数的参数与调用

上述实例中的求和函数未指定函数参数，但是一般情况下，设定函数参数可以更好的实现函数功能。参数的确定是在函数名之后的（）中进行，在python函数中参数分为五大类，设定函数参数是至关重要的一步。定义好一个函数后，直接使用函数名+参数即可完成对某一个函数的调用。依次来看每类参数的定义和调用方式。

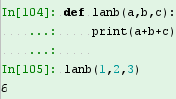
|  |
| --- |
| 参数类型及调用实例 |
| 位置参数：一般函数默认通过位置传递参数，从左到右  对于位置参数在调用时必须进行指定，不然会报错 |
| 默认参数：默认参数的值必须是不可变的对象，对于多个参数而言这样不仅可以降低调用的复杂度，而且可以接受调用过程中更新变化的值。默认参数顾名思义若不指定则直接使用默认的值，指定的话，可以按照位置参数指定方法，依次按顺序提供每个默认参数的值；还可以按照参数名称直接指定，而不用考虑参数的前后位置。    如果在定义参数中使用了可变的对象pycharm会进行较为人性化的提醒，虽然强制定义也是可行的，但是可能影响到之后程序的功能，如下面例子可知默认参数的值是不断变化的，因此尽可能按照要求使用不可变对象。 |
| 可变参数：使用\*进行标注，接受任何可变数量参数，在调用时，可以直接给定任意个数的符合要求的参数。这些参数最终会以元组形式呈现。 |
| 命名关键字参数：只用于接收指定关键字的参数值，\*表示间隔符。之后的参数就是命名关键字参数，可以有默认值，或者只有关键字。在调用时没有默认值的参数都必须指定，而且命名关键字必须同时传入关键字和参数值。 |
| 关键字参数：接受任意关键字的参数值，使用\*\*表示关键字参数。在调用时可以传入0-多个关键字=参数值（必须一起）形式的参数，或者是\*\*在这之后直接加字典或者Json格式的数组。最终这些参数以字典格式呈现。 |

匿名函数

对于功能较为简单的函数，可以使用匿名函数替代标准的函数的定义，从而简化函数的应用。在python中通过关键字lambda来创建匿名函数，无需指定函数名，函数体是表达式而非代码块，而且只能使用自己命名空间中的参数值，而不能使用全局变量中的参数值。具体结构如下：（：之前为参数，之后为函数体表达式）

lambda [arg1 [,arg2,.....argn]]:expression

匿名函数是一个函数对象返回值是匿名函数表达式结果，如果将匿名函数赋值给一个变量则该变量相当于函数名，通过调用该变量和相关参数即可实现标准函数的功能。来看下面两个等价的实例：

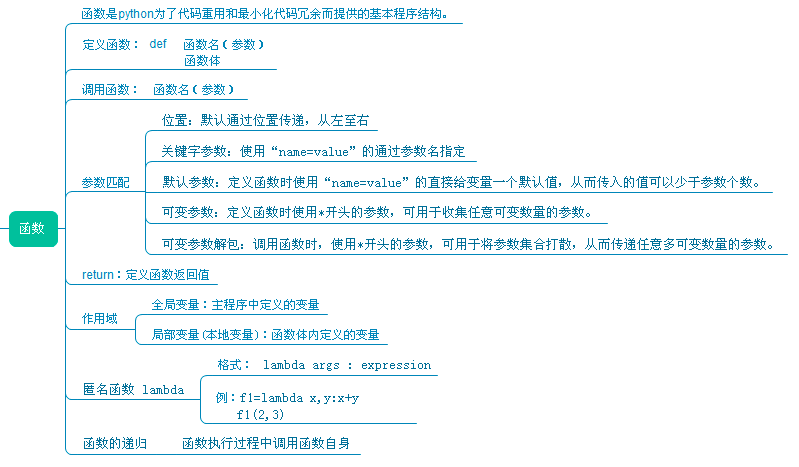
其他函数

递归函数：在函数体中调用函数本身。在之前的介绍中，我们有实现过一个通过递归方式进行阶乘的计算。直至函数中所传入的参数为1时，停止递归调用。

*#问题描述：使用递归调用的方法计算某个数的阶乘***import** sys  
**def Factorial**(*num*)**:  
 if** *num* **==** 1 **:  
 return** 1  
 **else:** result **=** *num* **\* Factorial**(*num***-**1)  
 **return** result  
  
**if** \_\_name\_\_ **== '\_\_main\_\_' :***)* result **= Factorial**(5)  
 print(**'5的阶乘是：{}'**.**format**(result))

高级函数：返回值也是一个函数

* python的模块详细介绍



默认参数可以简化函数的调用

一是必选参数在前，默认参数在后，否则Python的解释器会报错（思考一下为什么默认参数不能放在必选参数前面）；

二是如何设置默认参数。

当函数有多个参数时，把变化大的参数放前面，变化小的参数放后面。变化小的参数就可以作为默认参数。

可见，默认参数降低了函数调用的难度，而一旦需要更复杂的调用时，又可以传递更多的参数来实现。无论是简单调用还是复杂调用，函数只需要定义一个。

默认参数必须指向不变对象

。但是请注意，参数定义的顺序必须是：必选参数、默认参数、可变参数/命名关键字参数和关键字参数。

