python中函数的定义

在前面加 def 函数名() :

函数的参数

1.缺省参数

函数传递参数时

定义了默认的参数，这个参数就是缺省参数

缺省参数的定义只能在最后定义

带有参数的函数调用时

给参数传值时，可以使用 “参数 = 一个值”

这样可以不顾及变量的顺序

函数参数的传递

匹配模型：

位置：从左到右进行匹配

一般情况是通过位置进行匹配，把参数传递给函数头部的参数名称，匹配顺序为从左到右

关键字参数：通过参数名进行匹配

函数调用时可以定义哪一个参数接受这个值，方法：在调用的时候使用 语法：name=value

默认参数：为没有传入值得参数定义参数值

函数可以为参数定义接受的默认值，语法：name=value

可变参数：收集任意多的基于位置或关键字的参数

函数定义中的特殊参数，以\*开头，接收任意多个参数（这个参数是一个列表）

函数调用时对元组、字典进行拆包

在函数调用时，使用\*语法，会解包元组的集合

使用\*\*语法，会解包字典的集合

匹配语法

函数参数匹配表

|  |  |  |
| --- | --- | --- |
| 语法 | 位置 | 解释 |
| func(value) | 调用者 | 常规参数：通过位置进行匹配 |
| func(name=value) | 调用者 | 关键字参数：通过变量名匹配 |
| func(\*sequence) | 调用者 | 以name传递所有的对象，并作为独立的基于位置的参数 |
| func(\*\*dict) | 调用者 | 以name成对的传递所有的关键字/值，并作为独立的关键字参数 |
| def func(name) | 函数 | 常规参数：通过位置或变量名进行匹配 |
| def func(name=value) | 函数 | 默认参数值，如果没有在调用中传递的话 |
| def func(\*name) | 函数 | 匹配并收集（在元组中）所有包含位置的参数 |
| def func(\*\*name) | 函数 | 匹配并收集（在字典中）所有包含位置的参数 |
| def func(\*args,name)  def func(\*,name=value) | 函数 | 参数必须在调用中按照关键字传递 |

def 语句的格式

def <name>(arg1,arg2,…,argN)

<statements>

return <value>

用例

*# 2018年4月1日19:36:22  
# python3  
# jiang  
  
'''  
函数的定义，以及调用  
'''  
  
# 不传参的函数*print(**"# 不传参的函数"**)  
**def** shuchu\_1\_100():  
 ine = 1  
 ine\_sum = 0  
 **while** ine < 100:  
 ine\_sum += ine  
 ine += 1  
 **return** ine\_sum  
  
print(shuchu\_1\_100(),end=**'\n\n'**)  
  
*# 传参的函数*print(**'# 传参的函数'**)  
**def** shuchu(Max\_p):  
 ine = 1  
 ine\_sum = 0  
 **while** ine < Max\_p:  
 ine\_sum += ine  
 ine += 1  
 **return** ine\_sum  
  
print(shuchu(int(input(**'输入最大值'**))),end=**'\n\n'**)  
  
*# 返回多个值*print(**'# 返回多个值'**)  
**def** test():  
 a = 11  
 b = 22  
 c = 33  
 **return** [a,b,c]*# 用列表封装变量*num = test()  
print(num)

**def** test(a,b=22):*# b=22 b是缺省参数* result = a+b  
 print(result)

-------------------------------------------------------------------------------