函数的传参(把数据给函数)

函数的形参(接收函数调用传过来的数据)

python 函数的参数传递

传递方式:

1. 位置传参

1.1 序列传参

2. 关键字传参

2.1 字典关键字传参

位置传参:

实际参数(实参) 的对应关系与形式参数(形参)的对应关系是

按位置来依次对应的.

示例见:

position\_give\_args.py

说明:

实参和形参通过位置进行传递和匹配

实参的个数必须与形参的个数相同

序列传参

序列传参是指在函数调用过程中,用\*(星号) 将序列拆解后

按位置进行传递的传参方式

说明:

序列传参时,序列拆解的位置将与形参一一对应

示例见:

sequence\_give\_args.py

关键字传参:

关键字传参是指传参时,按着形参名称给形参赋值

实参和形参按名称进行匹配

说明:

可以不按位置进行匹配

示例见:

keywords\_give\_args.py

字典关键字传参:

是指实参为字典,将字典用\*\* 拆解后进行关键字传参的传

递方式

说明:

字典的键名和形参名必须一致

字典的键名必须为符合标识符命名规则的字符串

字典的键名要在形参中存在

示例见:

dict\_keyword\_give\_args.py

函数的综合传参

1. 函数的传参方式在能确定形参能唯一匹配到相应实参的情

况下可以任意组合

2. 函数的位置传参要先于关键字传参

-------- 以下讲的内容是函数的形参-------

函数的缺省参数

语法:

def 函数名(形参名1=默认实参1, 形参名2=默认实参2,

形参名3=默认实参3, .....)

作用:

让函数的调用者少传递参数来调用函数

示例见:

default\_args.py

说明:

1. 缺省参数必须自右至左依次存在,如果一个参数有缺省

参数, 则其右侧的所有参数都必须有缺省参数

2. 缺省参数可以有0个,1个或多个,甚至全部都有缺省参数

如:

def fn(a, b=10, c): # 错误

pass

def fn(a=0, b=None, c=False): # 是这对的

Pass

函数形参的定义方式

位置形参

星号元组形参

命名关键字形参

双星号字典形参

位置形参:

语法:

def 函数名(形参变量1, 形参变量2, ....):

语句块

星号元组形参

语法:

def 函数名(\*元组形参名):

语句块

作用:

收集多余的位置传参

说明:

元组形参名一般命名为'args'

\*元组形参一个函数只能有一个

示例见:

star\_tuple\_args.py

练习:

写一个程序，mysum 可以传入任意个实参的参数，

此函数的功能是返回所有实参的和

def mysum(\*args):

....

print(mysum(1,2,3,4)) # 10

print(mysum(1,2,3)) # 6

命名关键字形参

语法

def 函数名(\*,命名关键字形参1, 命名关键字形参2,..):

语句块

或

def 函数名(\*args,命名关键字形参1, 命名关键字形参2,..):

语句块

作用:

强制所有的命名关键字形参都必须用关键字传参或字典关

键字传参

示例见:

named\_keywords\_args.py

思考:

print()　函数的形参列表是如何定义的

练习:

已知内建函数 max 的绑助文档为:

max(...)

max(iterable)---> value

max(args1, arg2, \*args) --> value

仿造max函数，写一个与max功能完全一样的mymax函数

(要求: 不允许调用内键的max函数)

如:

def mymax(...):

...

print(mymax([6, 8, 3, 5])) # 8

print(mymax(100, 200)) # 200

print(mymax(1, 3, 5, 9, 7)) # 9

print(mymax()) # 报错

双星号字典形参

语法:

def 函数名(\*\*字典形参名):

语句块

作用:

收集多余的关键字传参

说明:

字典形参名一般命名为'kwargs'

一个函数内字典形参最多只有一个

示例见:

dict\_kwargs.py

函数的形参说明:

位置形参，缺省参数，星号元组形参，双星号字典形参可以混

合使用

函数形参自左至右的定义顺序为:

位置形参

星号元组形参

命名关键字形参

双星号字典形参

示例:

def fn(a, b, \*args, c, \*\*kwargs):

pass

fn(1, 2, 3, 4, c=30, d=40, e=50)

函数的不定长参数:

有两种: 星号元组形参，双星号字典形参

用不定长参数可以接收任意的传参

如:

def fn(\*args, \*\*kwargs):

print(args, kwargs)

#形参实参传递讲解  
def f1(a,b,\*args,c,\*\*kwargs):  
 print(a,b,args,c,kwargs)  
 print('ok')  
  
f1(1,2,3,4,d=6,c=5,e=7)  
f1(\*'hello',4,5,6,c=99,d=88,e=77)

f1(\*'hello',4,5,6,c=99,\*\*{‘n’:999,’m’:888})