<https://t66y.com/htm_data/25/1710/2733076.html>

<https://t66y.com/htm_data/25/1710/2733765.html>

https://t66y.com/thread0806.php?fid=25

1.函数注释

import math

def add(x,y):

'计算两个数的和'

return x+y

print((add(2,6)))

print(add.\_\_doc\_\_)

help(add)

help(abs)

2. 传递参数

值传递：数值，字符串，布尔

引用传递：复合类型，列表，字典，对象

3.关键字参数和参数默认值

'''关键字参数,默认参数

'''

#1.位置参数

def greet(name,greeting):

return '问候语:{}，姓名:{}'.format(greeting,name)

print(greet('leon','helo'))

#2.关键字参数

print(greet(greeting="nihao",name='yu'))

#关键字/位置参数混合使用:

# a.关键字必须跟在位置参数后面；

#b.同一个形参，不能同时使用位置参数和关键字参数；

#3.默认值参数

#a.如果一个参数有默认值，那么该参数后面的所有参数必须都有默认参数；

def fun(a,b,c,d,e,f=20,g=5):

print(a,b,c,d,e,f,g)

fun(1,2,3,4,5)

4.可变参数

#可变参数

#可变参数在函数内部表现形式是一个元组

def printParams(\*params):

print(params)

printParams("hello",123)

def printParams1(\*params):

for item in params:

print('<',item,'>',end=' ')

printParams1(123,True,'hello')

print()

#一般情况下，可变参数放在普通参数后面

def printParams2(value,\*params):

print('['+str(value)+']')

for item in params:

print('<',item,'>',end=' ')

printParams2('aaaaa',123,True)

#可变参数，后面的参数必须是关键字参数；

#有默认参数，可以不传，要么必须关键字传入

def printParams3(value1,\*param,value2=77,value3=88):

print('[', value1, ']')

print('[', value2, ']')

print('[', value3, ']')

for item in param:

print('<',item,'>')

printParams3(123,True,456,'hello')

5.序列作为函数的参数值

#将序列作为函数的参数值

def printParam1(s1,s2):

print(s1,s2)

printParam1('hello','world')

strList = ['Hello','World']

printParam1(\*strList)

#

def printParam2(\*ss):

for s in ss:

print('<',s,'>',end=' ')

printParam2(\*strList)

print()

printParam2(\*'abcdefghijkl')

print()

printParam2(\*[1,2,3,4,5,6])

print()

#传递字典

def printParams3(\*\*ss):

for item in ss.items():

print(item)

print('{}={}'.format(item[0],item[1]))

dict={'a':2,'b':3,'c':456}

printParams3(\*\*dict)

def printParams4(ss):

for item in ss.items():

print(item)

print('{}={}'.format(item[0],item[1]))

dict1={'a':2,'b':3,'c':456}

printParams4(dict1)

6.作用域

m=10

def fun5():

m=100

print('fun5')

def fun6():

print(m)

print('fun6')

return fun6

fun5()()

fun5

100

fun6

7.函数递归

def fun(n):

if n==0 or n==1: #终止条件

return 1

return n\*fun(n-1)

print(fun(3))

def fibnonacci(n):

if n==0 or n==1: #终止条件

return n

return fibnonacci(n-1)+ fibnonacci(n-2)

print(fibnonacci(8))