位置参数：

传入的变量值需要与函数参数相对应

默认参数(降低函数难度）：

当你修改了函数参数个数时，就需要输入指定参数个数，可以通过设置默认参数是，在你传参时，会自动根据设置的变量值进行计算

def power(x, n=2):
s = 1
while n > 0:
n = n - 1
s = s \* x
return s
>>>power(5)
25
>>>pwer(5，3)
125

可变参数：

将参数作为list或者Tuple进行传递,可以实现，但是没次都需要进行构建

def calc(numbers):
sum = 0
for n in numbers:
sum = sum + n \* n
return sum
>>> calc([1, 2, 3])
14

之间在参数前面加\*（可以传递任意参数)

def calc(\*numbers):
sum = 0
for n in numbers:
sum = sum + n \* n
return sum

关键字参数

可变参数允许你传入0个或任意个参数，这些可变参数在函数调用时自动组装为一个tuple。而关键字参数允许你传入0个或任意个含参数名的参数，这些关键字参数在函数内部自动组装为一个dict。

def person(name, age, \*\*kw):
print('name:', name, 'age:', age, 'other:', kw)
#KW表示为非必选参数，可以传递任意个关键字参数，堆kw的改变不会影响dict值

命名关键字参数

限制关键字参数的名称，只接收指定关键字参数

def person(name, age, \*, city, job):
print(name, age, city, job)

注意：使用命名关键字参数时，如果没有可变参数，就必须加一个\*作为特殊分隔符。如果缺少\*，Python解释器将视为位置参数。

可变参数与关键字参数区别：

\*args是可变参数，args接收的是一个tuple；

\*\*kw是关键字参数，kw接收的是一个dict。

以及调用函数时如何传入可变参数和关键字参数的语法：

可变参数既可以直接传入：func(1, 2, 3)，又可以先组装list或tuple，再通过\*args传入：func(\*(1, 2, 3))；

关键字参数既可以直接传入：func(a=1, b=2)，又可以先组装dict，再通过\*\*kw传入：func(\*\*{'a': 1, 'b': 2})。