**1、字符串文本解析函数**

（1）split

s.split(x)

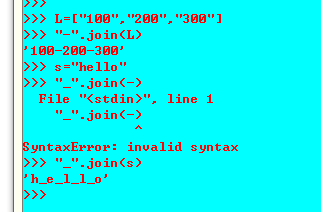
拆分字符串，通过制定分隔符x对字符串进行切片，返回分隔后的字符串列表



（2）join

s.join(iterable)

可迭代对象中的字符串，返回一个中间用s分隔的字符串



**1、函数**

（1）概念

函数是可以重复执行的语句块【函数可以视为一段程序的集合，并且给予一个名称来替代】

（2）作用

可以重复使用，提高代码的重用性

（3）语法

def 函数名（参数列表【形参】）

语句块（代码块）

（4）说明

#函数名必须是标识符，符合变量的命名规则

#函数有自己的名字，要让函数处理外部数据，通过参数列表对此函数传入一些数据，如果不需要传入数据，参数列表可以为空，语句部分不能为空，如果为空则需要pass语句来填充

#函数是一个变量，不能轻易对其赋值

（5）函数的调用

#语法

函数名（传递参数【实参】）

#说明

a.函数调用是一个表达式

b.如果没有return语句，此时函数执行完毕后返回None对象

c.如果函数需要返回其他的对象需要用return语句

（6）return语句

#语法

return[表达式] []部分可以省略

#作用

用于函数中，将结束前函数的执行，返回到调用该函数的地方，同时返回一个对象的引用关系

#说明

a. return语句后的表达式可以省略，省略相当于return None

b. 如果函数内没有return语句，则函数执行完最后一条语句返回None，相当于在最后加上了一条return None 语句

（7）函数的类型

#有参数有返回值

#无参数有返回值

#无参数无返回值

#有参数无返回值

（8）函数的文档字符串

#作用

描述/说明函数功能的字符串

#语法

def 函数名（参数列表）

函数的文档字符串

语句块

#说明

文档字符串使用惯例；他的首行简述函数功能，第二行空行，第三行为函数的具体的描述

（9）函数的实参传递方式

#位置传参

实参的对应关系是按照位置来匹配的

def my\_fun(a,b,c):

pass

my\_fun(1,2,3)

说明：

实参和形参通过位置进行匹配的传递

形参个数和实参个数必须相同

#序列传参

序列传参是指在调用函数的过程中，用\*将序列拆解后按照位置传参的方式进行传参

def my\_fun(a,b,c)

pass

L=[1,2,3]

my\_fun(\*L)

#关键字传参

关键字传参是指传是指传参时，按照形参的名称给形参赋值

形参和实参按照名称进行匹配

def my\_fun(a,b,c)

pass

my\_fun(b=2,c=3,a=1)

#字典关键字传参

是指参数为字典，将字典用\*\*拆解后按照关键字传参的方式传参

def my\_fun(a,b,c)

pass

d= (“a”:1,”b”:2,”c”:3)

my\_fun(\*\*d)

说明：

字典的键名必须和形参名一致

字典的键必须为字符串

字典键名要在形参中存在

#函数的综合传参

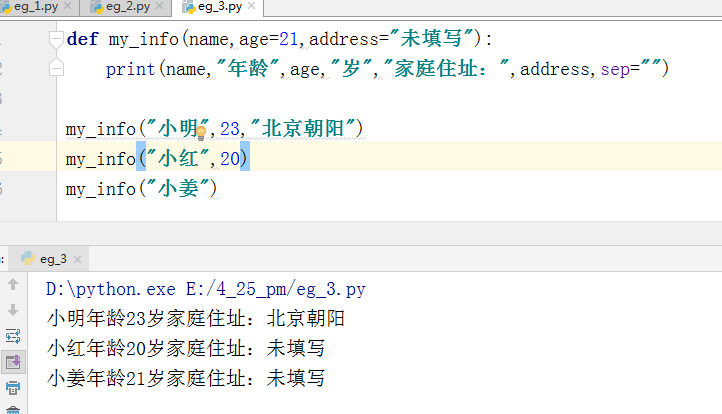
函数的传参方式，在能确定每个形参都能唯一匹配一个实参的情况下也可以任意传参，传参时，位置传参和序列传参要先于关键字传参和字典传参

（10）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=默认名2，……）

语句块



缺省参数必须自右至左依次存在，如果有一个参数是缺省参数，其右侧的所有参数都必须为缺省参数

def test(a,b=10,c): #这种写法是错误的！！

pass

缺省参数可以有1个或者多个，甚至全部参数都可以为缺省参数

（11）函数的形参定义方式

#位置形参

a.概念

按照参数位置接收实参

b.语法

def 函数名（形参1，形参2，形参3）

pass

#星号元组形参

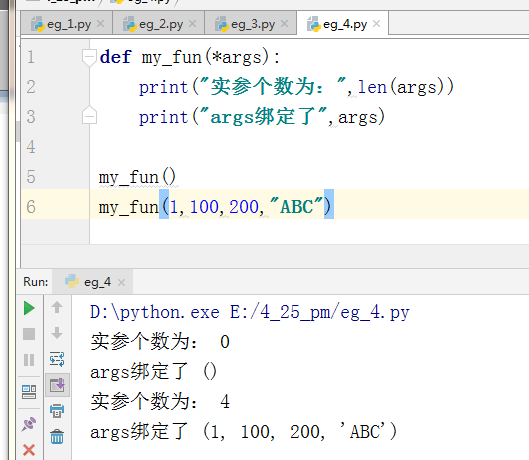
a. 语法

def 函数名（\*元组形参名）

pass

b. 作用

收集多合的位置形参



#命名关键字形参

a.语法

def 函数名（\*,命名关键字形参）

语句块

def 函数名（\*args, 命名关键字形参）

语句块

b. 作用

所用的命名关键字形参必须用关键字传参或者关键字传参传递数据

#双星号字典形参

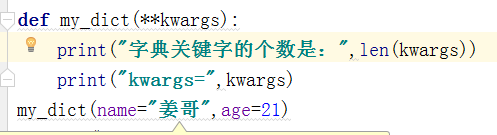
a. 语法

def 函数名（\*\*,字典形参）

pass

b. 作用

收集多余的关键字传参



函数形参自右至左的顺序为：

位置形参，星号元组形参，命名关键字形参，双星号字典形参

练习：

写一个函数，可以传入任意一个实参的数字，返回所有实参的和

def my\_sum()

…..

print(my\_sum(1,2,3,4,5,6,7,8,9,10)) #和为55

