对文件的操作

对文件的操作：读和写

函数 open( ) :打开

格式：变量名=open（文件名，权限，编码方式）

例：

f=open（‘a.txt’，‘w’,encoding=‘utf-8’）

1、文件名

(1)如果操作是当前路径下的文件，直接写文件名

(2)如果是其他路径下面的文件要加上路径，路径要不转移

(1) \n、\t、\r (2) [\\n、\\t、\\r](file:///\\n、\\t、\\r) 转意，一个斜杠是转意，两个斜杠是不转意。在路径前加r也是不转意

例：f=open(f‘D：\users\ZH\AppData\a.txt’,‘w’，encoding=‘utf-8’

2、权限

r读 w写

w写 write（‘字符串’）

例 f.write(‘qwert’) 不具备换行功能

r 读 read( ) 读取文件中所有内容，结果是字符串

readlines( ) 读取文件中所有的内容，结果是列表，每一行就是列表中的每个元素。

readline( ) 每次只读取文件中一行，本身是有迭代功能

注：每执行结束后要关闭文件

f.close：关闭文件

例：向文件中写入十行内容 排量操作

f=open(r‘a.txt’,‘w’，encoding=‘utf-8’）

for i in range（10）

f.write（'{ }\n.format(‘abc’)）

f.close

a 追加 write（‘字符串’）写的模式

例：f=open（r‘a.txt’,‘a’，encoding=‘utf-8’）

权限 w+ r+ a+ 可读、可写、可追加

wb rb a+ 以二进制的方式读取、写入

函数

函数：可重复使用的，具有某种功能的代码块

优点：1、节省代码 2、重复使用

定义格式：def 函数名( ):

代码块

函数名的命名规则：1、字符串、数字、下划线

2、以字符串开头

3、不能以Python中的内置函数命名

4、不能以Python的文件命名

调用函数:1、加括号带表执行此函数 函数名( )

2、不加括号代表函数名

一、变量的作用域：局部变量和全局变量

在函数里面定义的变量是局部变量：在局部范围内有效的

在文件中定义的变量是全局 ：在全局范围内有效的

globl 变量名：将局部变量变为全局变量

二、结果赋值

格式：return 数据

1. 赋值 将return后面的数据赋值给调用者
2. 结束标志 看到return表示函数到此结束

例 def abc( )

b=sum(range(101))

print(101)

return 1000

print(abc( ))

三、参数传递 给函数中传入参数

1、必须参数又叫必须传入的函数

格式：def 函数名（参数名）： 或(变量名)

例： def ab(x)

Print(x+1)

ab( )

用法： def ab(x)

print(x+1)

ab(10)

1. 参数名的命名符合变量名的规则
2. 参数名个数不限制
3. 每增加一个参数名用逗号隔开

2，默认参数

格式：def 函数名(参数名=值)

特点：可传入也可不传入

例： def abc(a=100,b=200)

Print(a+b)

abc( )

1. 可变长参数

可以传入多个数据

格式： def 函数名(\*参数名)

例： def abc(\*args)

Print(args)

abc(12,23,‘qw’，3)

原理：接受到多少个数据以后，变成元组传到函数中

格式： def abc(\*\*kwargs):

Print(kwargs)

abc(name=‘小王’，age=20,sex=‘男’)

三个参数连用或两个参数连用是有优先级的：必须参数>默认参数>可变长参数

Python的内置函数：help 查看某些数据类型的函数

例：print(help(str))

匿名函数

Lambda

格式：函数名=lambda： 表达式

匿名函数属于自定义函数的一种lambda 和 def

Lambda一般用于只有一个表达式的函数，不能处理复杂的，具有逻辑结构的代码。 def：能够处理任何的代码

例：sum=lambda x,y : print(x,y)

Print(10,90)

列表推导式

将控制语句写在列表中，使产生的结果直接存在列表中

格式： 列表变量名=[变量名 控制语句]

例1：a=[ ]

for i in range(10)

1. append(i)

print(a)

列表推导式

例2： a=[ i for i in range(10)]

Print(a)

例1和例2两者相同

内置函数

Ard 是将字符转化为ascii中的数字

Chr 是将数字转化为ascii中的字符

例： print(ord(‘a’))

Print(chr(97))

divmod:整除求余

例：a,b=divmod(100,16)

Print(a,b)

hex(123):十进制转十六进制

bin(123): 十进制转二进制

oct(123):十进制转八进制

Int(标识):任何进制数转十进制

With 语句 ：上下文管理器

格式：with 操作的东西 as 变量名：

例with open（‘a.txt’,‘r’,encoding=‘utf-8’）as f

With 语句原理：\_enter\_、\_exit\_

可以保证：不管在处理过程中是否发生错误或者发生异常都会执行\_exit\_(清理)操作，释放被访问的资源

用于：访问数据库、socket、对文件的操作等，需要执行执行关闭命令的场景