## 常用方法

#### Py2,py3切换-virtualenv

virtualenv --no-site-packages venv

**virtualenv envpy3 -p C:\Python34\python.exe**

**cd Scripts**

activate.bat

deactivate.bat

#### 数学计算

##### [除、取整、取余计算](http://blog.csdn.net/u014647208/article/details/53368244)

除法

>>>2/4

求余

>>>10%2

取整

>>>2//4

求最大值

Max()

**拼接list为一个字符串**

str = "".join([x.decode() for x in res])

**open**

with open(mc["monkey\_log"], encoding='utf-8') as monkeylog:

看activity

adb logcat ActivityManager:I \*:s

问题

[ZeroDivisionError: integer division or modulo by zero](https://www.liaoxuefeng.com/wiki/001374738125095c955c1e6d8bb493182103fac9270762a000/001386832284796780f5db7b5744bf9989f8d845ef77712000)

根据错误类型ZeroDivisionError，我们判断，int(s)本身并没有出错，但是int(s)返回0，在计算10 / 0时出错，至此，找到错误源头。

**Pickle**

path="pickle.pkl"

s=getcpuinfo()

f=open(path, 'a+')

data1= Pickle.dump(s, f)

print data1

f=open(path, 'a+')

data2= Pickle.load(f)

print data2

#### 列表方法：

|  |  |
| --- | --- |
| 方法 | 说明 |
| append( item ) | 在列表末尾插入（item ） |
| count( element ) | 返回element在列表中出现的次数 |
| extend( newlist ) | 将newlist的元素插入列表末尾 |
| index( element ) | 返回element在列表中的索引，如果不存在，则引发ValueError异常 |
| insert( index , item ) | 在index位置插入item |
| pop( [index] ) | index参数为可选的。如无参调用，则会删除并返回列表最后一个元素。如指定index参数，则删除并返回位置index的元素 |
| remove( element ) | 删除首次在列表中出现的element，如果列表不存在element，则引发 ValueError异常 |
| reverse( ) | 当场反转列表内容（不创建反转列表的拷贝） |
| sort | 当场对列表内容排序。可选参数compare-function是一个函数，它指定了比较条件 |
|  | compare-function取得列表的任意两个元素（x和y）。如果x应出现在y之前，就范围-1；如果x和y的顺序无关紧要，返回0；如果x应出现在y之后，则返回1 |

#### 字典方法：

备注：

     字典方法中的copy( )方法为浅拷贝（含义见说明）。字典深拷贝方法为（假设i为字典）：

w=i[:]

这表示w字典为i字典的一个深拷贝。

|  |  |
| --- | --- |
| 方法 | 说明 |
| clear( ) | 从字典删除所有项 |
| copy( ) | 创建并返回字典的浅拷贝（新字典元素为原始字典的引用） |
| get(key [ ,returnvalue] ) | 返回key的值，若无key而指定了returnvalue，则返回returnvalue 值，若无此值则返回None |
| has\_key( key ) | 如果key存在于字典中，就返回1（真）；否则返回0（假） |
| items( ) | 返回一个由元组构成的列表，每个元组包含一对键-值对 |
| keys( ) | 返回一个由字典所有键构成的列表 |
| popitem( ) | 删除任意键-值对，并作为两个元素的元组返回。如字典为空，则返回KeyError异常 |
| setdefault( key [,dummvalue]) | 具有与get方法类似的行为。如key不在字典中，同时指定了dummvalue，就将键（key）和指定的值（dummvalue）插入字典，如果没有指定dummvalue，则值为None |
| update( newDictionary ) | 将来自newDictionary的所有键-值添加到当前字典，并覆盖同名键的值 |
| values( ) | 返回字典所有值组成的一个列表 |
| iterkeys( ) | 返回字典键的一个迭代器 |
| iteritems( ) | 返回字典键-值对的一个迭代器 |
| itervalues( ) | 返回字典值的一个迭代器 |

#### 字符串方法

##### 1、去空格及特殊符号

复制代码代码如下:

s.strip().lstrip().rstrip(',')

##### 2、复制字符串

复制代码代码如下:

#strcpy(sStr1,sStr2)

sStr1 = 'strcpy'

sStr2 = sStr1

sStr1 = 'strcpy2'

print sStr2

##### 3、连接字符串

复制代码代码如下:

#strcat(sStr1,sStr2)

sStr1 = 'strcat'

sStr2 = 'append'

sStr1 += sStr2

print sStr1

##### 4、查找字符

复制代码代码如下:

#strchr(sStr1,sStr2)

# < 0 为未找到

sStr1 = 'strchr'

sStr2 = 's'

nPos = sStr1.index(sStr2)

print nPos

##### 5、比较字符串

复制代码代码如下:

#strcmp(sStr1,sStr2)

sStr1 = 'strchr'

sStr2 = 'strch'

print cmp(sStr1,sStr2)

##### 6、扫描字符串是否包含指定的字符

复制代码代码如下:

#strspn(sStr1,sStr2)

sStr1 = '12345678'

sStr2 = '456'

#sStr1 and chars both in sStr1 and sStr2

print len(sStr1 and sStr2)

##### 7、字符串长度

复制代码代码如下:

#strlen(sStr1)

sStr1 = 'strlen'

print len(sStr1)

##### 8、将字符串中的大小写转换

复制代码代码如下:

S.lower() #小写

S.upper() #大写

S.swapcase() #大小写互换

S.capitalize() #首字母大写

String.capwords(S) #这是模块中的方法。它把S用split()函数分开，然后用capitalize()把首字母变成大写，最后用join()合并到一起

#实例：

#strlwr(sStr1)

sStr1 = 'JCstrlwr'

sStr1 = sStr1.upper()

#sStr1 = sStr1.lower()

print sStr1

##### 9、追加指定长度的字符串

复制代码代码如下:

#strncat(sStr1,sStr2,n)

sStr1 = '12345'

sStr2 = 'abcdef'

n = 3

sStr1 += sStr2[0:n]

print sStr1

##### 10、字符串指定长度比较

复制代码代码如下:

#strncmp(sStr1,sStr2,n)

sStr1 = '12345'

sStr2 = '123bc'

n = 3

print cmp(sStr1[0:n],sStr2[0:n])

##### 11、复制指定长度的字符

复制代码代码如下:

#strncpy(sStr1,sStr2,n)

sStr1 = ''

sStr2 = '12345'

n = 3

sStr1 = sStr2[0:n]

print sStr1

##### 12、将字符串前n个字符替换为指定的字符

复制代码代码如下:

#strnset(sStr1,ch,n)

sStr1 = '12345'

ch = 'r'

n = 3

sStr1 = n \* ch + sStr1[3:]

print sStr1

##### 13、扫描字符串

复制代码代码如下:

#strpbrk(sStr1,sStr2)

sStr1 = 'cekjgdklab'

sStr2 = 'gka'

nPos = -1

for c in sStr1:

    if c in sStr2:

        nPos = sStr1.index(c)

        break

print nPos

##### 14、翻转字符串

复制代码代码如下:

#strrev(sStr1)

sStr1 = 'abcdefg'

sStr1 = sStr1[::-1]

print sStr1

##### 15、查找字符串

复制代码代码如下:

#strstr(sStr1,sStr2)

sStr1 = 'abcdefg'

sStr2 = 'cde'

print sStr1.find(sStr2)

##### 16、分割字符串

复制代码代码如下:

#strtok(sStr1,sStr2)

sStr1 = 'ab,cde,fgh,ijk'

sStr2 = ','

sStr1 = sStr1[sStr1.find(sStr2) + 1:]

print sStr1

#或者

s = 'ab,cde,fgh,ijk'

print(s.split(','))

##### 17、连接字符串

复制代码代码如下:

delimiter = ','

mylist = ['Brazil', 'Russia', 'India', 'China']

print delimiter.join(mylist)

##### 18、PHP 中 addslashes 的实现

复制代码代码如下:

def addslashes(s):

    d = {'"':'\\"', "'":"\\'", "\0":"\\\0", "\\":"\\\\"}

    return ''.join(d.get(c, c) for c in s)

s = "John 'Johny' Doe (a.k.a. \"Super Joe\")\\\0"

print s

print addslashes(s)

##### 19、只显示字母与数字

复制代码代码如下:

def OnlyCharNum(s,oth=''):

    s2 = s.lower();

    fomart = 'abcdefghijklmnopqrstuvwxyz0123456789'

    for c in s2:

        if not c in fomart:

            s = s.replace(c,'');

    return s;

print(OnlyStr("a000 aa-b"))

##### 20、截取字符串

复制代码代码如下:

str = '0123456789′

print str[0:3] #截取第一位到第三位的字符

print str[:] #截取字符串的全部字符

print str[6:] #截取第七个字符到结尾

print str[:-3] #截取从头开始到倒数第三个字符之前

print str[2] #截取第三个字符

print str[-1] #截取倒数第一个字符

print str[::-1] #创造一个与原字符串顺序相反的字符串

print str[-3:-1] #截取倒数第三位与倒数第一位之前的字符

print str[-3:] #截取倒数第三位到结尾

print str[:-5:-3] #逆序截取，具体啥意思没搞明白？

##### 21、字符串在输出时的对齐

复制代码代码如下:

S.ljust(width,[fillchar])

#输出width个字符，S左对齐，不足部分用fillchar填充，默认的为空格。

S.rjust(width,[fillchar]) #右对齐

S.center(width, [fillchar]) #中间对齐

S.zfill(width) #把S变成width长，并在右对齐，不足部分用0补足

##### 22、字符串中的搜索和替换

复制代码代码如下:

S.find(substr, [start, [end]])

#返回S中出现substr的第一个字母的标号，如果S中没有substr则返回-1。start和end作用就相当于在S[start:end]中搜索

S.index(substr, [start, [end]])

#与find()相同，只是在S中没有substr时，会返回一个运行时错误

S.rfind(substr, [start, [end]])

#返回S中最后出现的substr的第一个字母的标号，如果S中没有substr则返回-1，也就是说从右边算起的第一次出现的substr的首字母标号

S.rindex(substr, [start, [end]])

S.count(substr, [start, [end]]) #计算substr在S中出现的次数

S.replace(oldstr, newstr, [count])

#把S中的oldstar替换为newstr，count为替换次数。这是替换的通用形式，还有一些函数进行特殊字符的替换

S.strip([chars])

#把S中前后chars中有的字符全部去掉，可以理解为把S前后chars替换为None

S.lstrip([chars])

S.rstrip([chars])

S.expandtabs([tabsize])

#把S中的tab字符替换没空格，每个tab替换为tabsize个空格，默认是8个

##### 23、字符串的分割和组合

复制代码代码如下:

S.split([sep, [maxsplit]])

#以sep为分隔符，把S分成一个list。maxsplit表示分割的次数。默认的分割符为空白字符

S.rsplit([sep, [maxsplit]])

S.splitlines([keepends])

#把S按照行分割符分为一个list，keepends是一个bool值，如果为真每行后而会保留行分割符。

S.join(seq) #把seq代表的序列──字符串序列，用S连接起来

##### 24、字符串的mapping，这一功能包含两个函数

复制代码代码如下:

String.maketrans(from, to)

#返回一个256个字符组成的翻译表，其中from中的字符被一一对应地转换成to，所以from和to必须是等长的。

S.translate(table[,deletechars])

# 使用上面的函数产后的翻译表，把S进行翻译，并把deletechars中有的字符删掉。需要注意的是，如果S为unicode字符串，那么就不支持 deletechars参数，可以使用把某个字符翻译为None的方式实现相同的功能。此外还可以使用codecs模块的功能来创建更加功能强大的翻译表。

##### 25、字符串还有一对编码和解码的函数

复制代码代码如下:

S.encode([encoding,[errors]])

# 其中encoding可以有多种值，比如gb2312 gbk gb18030 bz2 zlib big5 bzse64等都支持。errors默认值为"strict"，意思是UnicodeError。可能的值还有'ignore', 'replace', 'xmlcharrefreplace', 'backslashreplace' 和所有的通过codecs.register\_error注册的值。这一部分内容涉及codecs模块，不是特明白

S.decode([encoding,[errors]])

##### 26、字符串的测试、判断函数，这一类函数在string模块中没有，这些函数返回的都是bool值

复制代码代码如下:

S.startswith(prefix[,start[,end]])

#是否以prefix开头

S.endswith(suffix[,start[,end]])

#以suffix结尾

S.isalnum()

#是否全是字母和数字，并至少有一个字符

S.isalpha() #是否全是字母，并至少有一个字符

S.isdigit() #是否全是数字，并至少有一个字符

S.isspace() #是否全是空白字符，并至少有一个字符

S.islower() #S中的字母是否全是小写

S.isupper() #S中的字母是否便是大写

S.istitle() #S是否是首字母大写的

##### 27、字符串类型转换函数，这几个函数只在string模块中有

复制代码代码如下:

string.atoi(s[,base])

#base默认为10，如果为0,那么s就可以是012或0x23这种形式的字符串，如果是16那么s就只能是0x23或0X12这种形式的字符串

string.atol(s[,base]) #转成long

string.atof(s[,base]) #转成float

这里再强调一次，字符串对象是不可改变的，也就是说在python创建一个字符串后，你不能把这个字符中的某一部分改变。任何上面的函数改变了字符串后，都会返回一个新的字符串，原字串并没有变。其实这也是有变通的办法的，可以用S=list(S)这个函数把S变为由单个字符为成员的list，这样的话就可以使用S[3]='a'的方式改变值，然后再使用S=" ".join(S)还原成字符串

Partition

#### OS.path

os.path.abspath(path) #返回绝对路径

os.path.basename(path) #返回文件名

os.path.commonprefix(list) #返回list(多个路径)中，所有path共有的最长的路径。

os.path.dirname(path) #返回文件路径

os.path.exists(path)  #路径存在则返回True,路径损坏返回False

os.path.lexists  #路径存在则返回True,路径损坏也返回True

os.path.expanduser(path)  #把path中包含的"~"和"~user"转换成用户目录

os.path.expandvars(path)  #根据环境变量的值替换path中包含的”$name”和”${name}”

os.path.getatime(path)  #返回最后一次进入此path的时间。

os.path.getmtime(path)  #返回在此path下最后一次修改的时间。

os.path.getctime(path)  #返回path的大小

os.path.getsize(path)  #返回文件大小，如果文件不存在就返回错误

os.path.isabs(path)  #判断是否为绝对路径

os.path.isfile(path)  #判断路径是否为文件

os.path.isdir(path)  #判断路径是否为目录

os.path.islink(path)  #判断路径是否为链接

os.path.ismount(path)  #判断路径是否为挂载点（）

os.path.join(path1[, path2[, ...]])  #把目录和文件名合成一个路径

os.path.normcase(path)  #转换path的大小写和斜杠

os.path.normpath(path)  #规范path字符串形式

os.path.realpath(path)  #返回path的真实路径

os.path.relpath(path[, start])  #从start开始计算相对路径

os.path.samefile(path1, path2)  #判断目录或文件是否相同

os.path.sameopenfile(fp1, fp2)  #判断fp1和fp2是否指向同一文件

os.path.samestat(stat1, stat2)  #判断stat tuple stat1和stat2是否指向同一个文件

os.path.split(path)  #把路径分割成dirname和basename，返回一个元组

os.path.splitdrive(path)   #一般用在windows下，返回驱动器名和路径组成的元组

os.path.splitext(path)  #分割路径，返回路径名和文件扩展名的元组

os.path.splitunc(path)  #把路径分割为加载点与文件

os.path.walk(path, visit, arg)  #遍历path，进入每个目录都调用visit函数，visit函数必须有

3个参数(arg, dirname, names)，dirname表示当前目录的目录名，names代表当前目录下的所有

文件名，args则为walk的第三个参数

os.path.supports\_unicode\_filenames  #设置是否支持unicode路径名

## 常见问题

#### WindowsError: [Error 32]，

通过[os.remove(file)出错 WindowsError: [Error 32]](http://blog.csdn.net/summermaoz/article/details/68484278" \t "_blank), 可以推测，应该是文件打开没有关闭，我这里没有文件的操作，那么原因就是之前有一个同名的日志文件了，删掉，问题就解决了。

#### [python报错UnicodeDecodeError: 'ascii' codec can't decode byte 0xe8 in position 0 解决方案](http://www.cnblogs.com/pengei/p/6407077.html)

修改xlwt包中workbook.py中workbook类下的\_init\_函数的”encoding"默认值为：“encoding='gbk‘或’utf8‘中文搞定。

C:\Python27\Lib\site-packages\xlwt\Workbook.py

def \_\_init\_\_(self, encoding='utf-8', style\_compression=0):

参考: http://tieba.baidu.com/p/3123283499