os模块

https://www.cnblogs.com/ginvip/p/6439679.html

https://www.cnblogs.com/sunyang945/p/7900957.html

os模块就是对操作系统进行操作，使用该模块必须先导入模块

import os

环境变量

环境变量就是一些命令的集合

操作系统的环境变量就是操作系统在执行系统命令时搜索命令的目录的集合

函数

os.sep:取代操作系统特定的路径分隔符

os.name:指示你正在使用的工作平台。比如对于Windows，它是'nt'，而对于Linux/Unix用户，它是'posix'。

os.getcwd:得到当前工作目录，即当前python脚本工作的目录路径。

os.getenv()和os.putenv:分别用来读取和设置环境变量

os.listdir():返回指定目录下的所有文件和目录名

os.remove(file):删除一个文件

os.stat（file）:获得文件属性

os.chmod(file):修改文件权限和时间戳

os.mkdir(name):创建目录

os.rmdir(name):删除目录

os.removedirs（r“c：\python”）:删除多个目录

os.system():运行shell命令

os.exit():终止当前进程

os.linesep:给出当前平台的行终止符。例如，Windows使用'\r\n'，Linux使用'\n'而Mac使用'\r'

os.path.split():返回一个路径的目录名和文件名

os.path.isfile()和os.path.isdir()分别检验给出的路径是一个目录还是文件

os.path.existe():检验给出的路径是否真的存在

os.listdir(dirname):列出dirname下的目录和文件

os.getcwd():获得当前工作目录

os.curdir:返回当前目录（'.'）

os.chdir(dirname):改变工作目录到dirname,即去掉文件名，返回目录路径

os.path.basename(‘path/filename’): 去掉目录路径，返回文件名

os.path.isdir(name):判断name是不是目录，不是目录就返回false

os.path.isfile(name):判断name这个文件是否存在，不存在返回false

os.path.exists(name):判断是否存在文件或目录name

os.path.getsize(name):或得文件大小，如果name是目录返回0L

os.path.abspath(name):获得绝对路径

os.path.isabs():判断是否为绝对路径

os.path.normpath(path):规范path字符串形式

os.path.split(name):分割文件名与目录（事实上，如果你完全使用目录，它也会将最后一个目录作为文件名而分离，同时它不会判断文件或目录是否存在）

os.path.splitext():分离文件名和扩展名

os.path.join(path,name):连接目录与文件名或目录

os.path.basename(path):返回文件名

os.path.dirname(path):返回文件路径

os.path.getatime\ctime\mtime： 分别返回最近访问、创建、修改时间

文件操作

os.mknod("text.txt") # 创建空文件

fp = open("text.txt",w) # 直接打开一个文件，如果文件不存在就创建文件

open模式

w 写方式

a 追加模式打开(从EOF开始，必要时创建新文件)

r+ 以读写模式打开

w+ 以读写模式打开

a+ 以读写模式打开

rb 以二进制读模式打开

wb 以二进制写模式打开 (参见 w )

ab 以二进制追加模式打开 (参见 a )

rb+ 以二进制读写模式打开 (参见 r+ )

wb+ 以二进制读写模式打开 (参见 w+ )

ab+ 以二进制读写模式打开 (参见 a+ )

fp.read([size]) #size为读取的长度，以byte为单位

fp.readline([size]) #读一行，如果定义了size，有可能返回的只是一行的一部分

fp.readlines([size]) #把文件每一行作为一个list的一个成员，并返回这个list。其实它的内部是通过循环调用readline()来实现的。如果提供size参数，size是表示读取内容的总长，也就是说可能只读到文件的一部分。

fp.write(str) #把str写到文件中,write()并不会在str后加上一个换行符

fp.writelines(seq) #把seq的内容全部写到文件中(多行一次性写入)。这个函数也只是忠实地写入，不会在每行后面加上任何东西。

fp.close() #关闭文件。python会在一个文件不用后自动关闭文件，不过这一功能没有保证，最好还是养成自己关闭的习惯。 如果一个文件在关闭后还对其进行操作会产生ValueError

fp.flush() #把缓冲区的内容写入硬盘

fp.fileno() #返回一个长整型的”文件标签“

fp.isatty() #文件是否是一个终端设备文件（unix系统中的）

fp.tell() #返回文件操作标记的当前位置，以文件的开头为原点

fp.next() #返回下一行，并将文件操作标记位移到下一行。把一个file用于for … in file这样的语句时，就是调用next()函数来实现遍历的。

fp.seek(offset[,whence]) #将文件打操作标记移到offset的位置。这个offset一般是相对于文件的开头来计算的，一般为正数。但如果提供了whence参数就不一定了，whence可以为0表示从头开始计算，1表示以当前位置为原点计算。2表示以文件末尾为原点进行计算。需要注意，如果文件以a或a+的模式打开，每次进行写操作时，文件操作标记会自动返回到文件末尾。

fp.truncate([size]) #把文件裁成规定的大小，默认的是裁到当前文件操作标记的位置。如果size比文件的大小还要大，依据系统的不同可能是不改变文件，也可能是用0把文件补到相应的大小，也可能是以一些随机的内容加上去。

目录操作

os.mkdir("file") # 创建目录

shutil.copyfile("oldfile","newfile") #复制文件:oldfile和newfile都只能是文件

shutil.copy("oldfile","newfile") #oldfile只能是文件夹，newfile可以是文件，也可以是目标目录

shutil.copytree("olddir","newdir") #复制文件夹.olddir和newdir都只能是目录，且newdir必须不存在

os.rename("oldname","newname") #重命名文件(目录).文件或目录都是使用这条命令

shutil.move("oldpos","newpos") #移动文件(目录)

os.rmdir("dir") #只能删除空目录

shutil.rmtree("dir") #空目录、有内容的目录都可以删

os.chdir("path") #转换目录，换路径

函数使用

1. getcwd()

获取当前工作目录(当前工作目录默认都是当前文件所在的文件夹)

eg:

>>> import os

>>> result = os.getcwd()

>>> print (result)

C:\Python34

2. chdir()

改变当前工作目录

eg:

>>> os.chdir('/home/sy')

>>> result = os.getcwd()

>>> print(result)

home/sy

3. open()

打开文件

eg:

# 打开相对路径下的文件

>>> open('02.txt','w')

# 操作时如果书写完整的路径则不需要考虑默认工作目录的问题,按照实际书写路径操作

>>> open('/home/sy/下载/02.txt','w')

4. listdir()

获取指定文件夹中所有内容的名称列表

eg:

>>> result = os.listdir('/home/sy')

>>> print(result)

5. mkdir()

创建文件夹

eg:

>>> os.mkdir('girls')

>>> os.mkdir('boys',0o777)

6. makedirs()

递归创建文件夹

eg:

>>> os.makedirs('/home/sy/a/b/c/d')

7. rmdir()

删除空目录

eg:

>>> os.rmdir('girls')

8. removedirs()

递归删除文件夹，必须都是空目录

rg:

>>> os.removedirs('/home/sy/a/b/c/d')

9. rename()

文件或文件夹重命名

eg:

>>> os.rename('/home/sy/a','/home/sy/danran'

>>> os.rename('02.txt','002.txt')

10. stat()

获取文件或文件夹的信息

eg:

>>> result = os.stat('/home/sy/PycharmProject/Python3/10.27/01.py')

>>> print(result)

11. system()

执行系统命令(危险函数)

eg:

>>> result = os.system('ls -al')

>>> print(result)

12. getenv()

获取系统的环境变量

eg:

>>> result = os.getenv('PATH')

>>> print(result.split(':'))

13. putenv()

将一个目录添加到环境变量中(临时增加仅对当前脚本有效)

eg:

>>> os.putenv('PATH','/home/sy/下载')

>>> os.system('syls')

14. exit()

退出终端的命令

15. curdir()

表示当前文件夹，.表示当前文件夹，一般情况下可省略

eg:

>>> print(os.curdir())

16. pardir()

表示上一层文件夹,..表示上一层文件夹,不可省略!

eg:

print(os.pardir(..))

17. mkdir()

新建目录

eg:

>>> os.mkdir('../../../man') #相对路径,从当前目录开始查找

>>> os.mkdir('/home/sy/man1') #绝对路径,从根目录开始查找

18. name()

获取代表操作系统的名称字符串

eg:

>>> print(os.name())

19. sep()

获取系统路径间隔符号(window ->\ linux ->/)

eg:

>>> print(os.sep())

20. extsep()

获取文件名称和后缀之间的间隔符号(window & linux -> .)

eg:

>>> print(os.extsep)

21. linesep()

获取操作系统的换行符号(window -> \r\n linux/unix -> \n)

eg:

>>> print(repr(os.linesep))

os.path子模块函数

1. abspath()

将相对路径转化为绝对路径

eg:

>>> path = './boys' #相对

>>> result = os.path.abspath(path)

>>> print(result)

2. dirname()

获取完整路径当中的目录部分

eg:

>>> path = '/home/sy/boys'

>>> result = os.path.dirname(path)

>>> print(result)

3. basename()

获取完整路径当中的主体部分

eg:

>>> path = '/home/sy/boys'

>>> result = os.path.basename(path)

>>> print(result)

4. split()

将一个完整的路径切割成目录部分和主体部分

eg:

>>> path = '/home/sy/boys'

>>> result = os.path.split(path)

>>> print(result)

5. join()

将2个路径合并成一个

eg:

>>> var1 = '/home/sy'

>>> var2 = '000.py'

>>> result = os.path.join(var1,var2)

>>> print(result)

6. splitext()

将一个路径切割成文件后缀和其他两个部分,主要用于获取文件的后缀

eg:

>>> path = '/home/sy/000.py'

>>> result = os.path.splitext(path)

>>> print(result)

7. getsize()

获取文件的大小

eg:

>>> path = '/home/sy/000.py'

>>> result = os.path.getsize(path)

>>> print(result)

8. isdir()

检测是否是文件夹

eg:

>>> result = os.path.isdir(path)

>>> print(result)

9. islink()

检测是否是链接

eg:

>>> path = '/initrd.img.old'

>>> result = os.path.islink(path)

>>> print(result)

10. getctime()

获取文件的创建时间

get create time

eg:

>>> import time

>>> filepath = '/home/sy/下载/chls'

>>> result = os.path.getctime(filepath)

>>> print(time.ctime(result))

11. getmtime()

获取文件的修改时间

get modify time

eg:

>>> import time

>>> filepath = '/home/sy/下载/chls'

>>> result = os.path.getmtime(filepath)

>>> print(time.ctime(result))

12. getatime()

获取文件的访问时间

get active time

eg:

>>> import time

>>> filepath = '/home/sy/下载/chls'

>>> result = os.path.getctime(filepath)

>>> print(time.ctime(result))

>>> result = os.path.getmtime(filepath)

>>> print(time.ctime(result))

>>>

>>> result = os.path.getatime(filepath)

>>> print(time.ctime(result))

13. exists()

检测某个路径是否真实存在

eg:

>>> filepath = '/home/sy/下载/chls'

>>> result = os.path.exists(filepath)

>>> print(result)

14. isabs()

检测一个路径是否是绝对路径

eg:

>>> path = '/boys'

>>> result = os.path.isabs(path)

>>> print(result)

15. samefile()

检测2个路径是否是同一个文件

eg:

>>> path1 = '/home/sy/下载/001'

>>> path2 = '../../../下载/001'

>>> result = os.path.samefile(path1,path2)

>>> print(result)

16. os.environ()

用于获取和设置系统环境变量的内置值

eg:

>>> import os

#获取系统环境变量,getenv()效果

>>> print(os.environ['PATH'])

17. environ()

设置系统环境变量 putenv()

eg:

>>> os.environ(['PATH']) += ':/home/sy/下载'

>>> os.system('chls')