



ZWSOFT

# Python 学习笔记

刘子耀 (Bond Liu)

二〇二〇年七月

ZWSOFT

# Python模块

## os模块

### os模块的部分常用属性（Windows）

os模块常用属性	
<code>os.name</code>	返回电脑的操作系统类型（Windows）
<code>os.curdir</code>	当前工作目录即：'.'
<code>os.sep</code>	返回路径分隔符，'\'
<code>os.extsep</code>	返回文件扩展名分隔符，Windows下则返回'.'
<code>os.linesep</code>	返回文本文件行分隔符，即'\n\r'

### os模块常用方法

常用方法	方法简介
<code>os.remove()</code>	删除文件
<code>os.unlink()</code>	删除文件
<code>os.rename()</code>	重命名文件
<code>os.listdir()</code>	列出指定目录下所有文件
<code>os.chdir()</code>	改变当前工作目录
<code>os.getcwd()</code>	获取当前工作目录
<code>os.mkdir()</code>	新建一级目录
<code>os.rmdir()</code>	删除空目录(删除非空目录, 使用 <code>shutil.rmtree()</code> )
<code>os.makedirs()</code>	创建多级目录

常用方法	方法简介
os.removedirs()	删除多级目录
os.stat(file)	获取文件属性
os.chmod(file)	修改文件权限
os.utime(file)	修改文件时间戳
os.name(file)	获取操作系统标识
os.system()	执行操作系统命令
os.popen()	执行操作系统命令，并接收返回值
os.execvp()	启动一个新进程
os.fork()	获取父进程ID，在子进程返回中返回0
os.execvp()	执行外部程序脚本（Unix）
os.spawn()	执行外部程序脚本（Windows）
os.access(path, mode)	判断文件权限(详细参考cnblogs)
os.wait()	暂时未知

## os.path模块常用方法

常用方法	方法简介
os.path.basename(filename)	返回文件路径的文件名部分
os.path.join(dirname, basename)	将文件路径和文件名凑成完整文件路径
os.path.abspath(name)	获得绝对路径
os.path.splitunc(path)	把路径分割为挂载点和文件名
os.path.normpath(path)	规范path字符串形式
os.path.exists()	判断文件或目录是否存在
os.path.isabs()	如果path是绝对路径，返回True

常用方法	方法简介
<code>os.path.realpath(path)</code>	返回path的真实路径
<code>os.path.relpath(path,[start])</code>	从start开始计算相对路径,start为空则默认从当前工作目录开始
<code>os.path.normcase(path)</code>	转换path的大小写和斜杠
<code>os.path.isdir()</code>	判断name是不是一个目录, name不是目录就返回false
<code>os.path.isfile()</code>	判断name是不是一个文件, 不存在返回false
<code>os.path.islink()</code>	判断文件是否连接文件,返回boolean
<code>os.path.ismount()</code>	指定路径是否存在且为一个挂载点, 返回boolean
<code>os.path.samefile()</code>	是否相同路径的文件, 返回boolean
<code>os.path.getatime()</code>	返回最近访问时间 浮点型
<code>os.path.getmtime()</code>	返回上一次修改时间 浮点型
<code>os.path.getctime()</code>	返回文件创建时间 浮点型
<code>os.path.getsize()</code>	返回文件大小 字节单位
<code>os.path.commonprefix(list)</code>	返回list(多个路径)中, 所有path共有的最长的路径
<code>os.path.lexists()</code>	路径存在则返回True,路径损坏也返回True
<code>os.path.expanduser(path)</code>	把path中包含的"~"和"user"转换成用户目录
<code>os.path.expandvars(path)</code>	根据环境变量的值替换path中包含的"\$name"和"\${name}"
<code>os.path.sameopenfile(fp1,fp2)</code>	判断fp1和fp2是否指向同一文件
<code>os.path.samestat(stat1,stat2)</code>	判断stat tuple stat1和stat2是否指向同一个文件
<code>os.path.splitdrive(path)</code>	一般用在windows下, 返回驱动器名和路径组成的元组

常用方法	方法简介
<code>os.path.walk(path, visit,arg)</code>	遍历path，给每个path执行一个函数详细见手册
<code>os.path.supports_unicode_filenames()</code>	设置是否支持unicode路径名

# sys模块

常用方法	方法简介
<code>sys.argv</code>	命令行参数List，第一个元素是程序本身路径,实现从程序外部向程序传递参数
<code>sys.path</code>	返回模块的搜索路径，初始化时使用PYTHONPATH环境变量的值
<code>sys.modules.values</code>	返回系统导入的模块字段，key是模块名，value是模块
<code>sys.stdout</code>	标准输出
<code>sys.stdin</code>	标准输入
<code>sys.stderr</code>	错误输出
<code>sys.exec_prefix</code>	返回平台独立的python文件安装的位置
<code>sys.copyright</code>	记录python版权相关的东西
<code>sys.api_version</code>	解释器的C的API版本
<code>sys.executable</code>	Python解释程序路径
<code>sys.version_info</code>	‘final’表示最终,也有‘candidate’表示候选,表示版本级别，是否有后继的发行
<code>sys.byteorder</code>	本地字节规则的指示器，big-endian平台的值是‘big’，little-endian平台的值是‘little’
<code>sys.modules.keys()</code>	返回所有已经导入的模块列表
<code>sys.exit(n)</code>	退出程序，正常退出时exit(0)

常用方法	方法简介
sys.hexversion	获取Python解释程序的版本值，16进制格式如：0x020403F0
sys.version	获取Python解释程序的版本信息
sys.platform	返回操作系统平台名称
sys.stdout.write('aaa')	标准输出内容
sys.stdout.writelines()	无换行输出
sys.stdin.read()	输入一行
sys.stdin.readline()	从标准输入读一行，sys.stdout.write("a")屏幕输出a
sys.exc_clear()	用来清除当前线程所出现的当前的或最近的错误信息
sys.getdefaultencoding()	返回当前你所用的默认的字符编码格式
sys.getfilesystemencoding()	返回将Unicode文件名转换成系统文件名的编码的名字
sys.builtin_module_names	Python解释器导入的内建模块列表
sys.getwindowsversion()	获取Windows的版本
sys.setdefaultencoding(name)	用来设置当前默认的字符编码(详细使用参考文档)
sys.displayhook(value)	如果value非空，这个函数会把他输出到sys.stdout(详细使用参考文档)
sys.exc_info()	获取当前正在处理的异常类,exc_type、exc_value、exc_traceback当前处理的异常详细信息

## shutil模块

常用方法	方法简介
shutil.copyfileobj(src,dst[,length=16*1024])	将src的内容覆盖copy给dst(先open得到fileobj),length为dst每次读取的长度，用做缓冲区大小

常用方法	方法简介
shutil.copyfile(src,dst)	将src文件内容复制至dst文件
shutil.copymode(src, dst)	将src文件权限复制至dst(必须存在)文件。 文件内容，所有者和组不受影响
shutil.copystat(src, dst)	将权限，上次访问时间，上次修改时间 以及src的标志复制到dst(必须存在)。 文件内容，所有者和组不受影响
shutil.copy(src, dst)	会在dst目录下创建与src同名的文件， 若该目录下存在同名文件，将会报错提示已经 存在同名文件。权限会被一并复制。本质是 先后调用了copyfile与copymode而已
shutil.copy2(src, dst)	会在dst目录下创建与src同名的文件，若该目录下 存在同名文件，将会报错提示已经存在同名文件。 权限、上次访问时间、上次修改时间 和src的标志会一并复制至dst。 本质是先后调用了copyfile与copystat方法而已
shutil.ignore_patterns(*patterns)	忽略模式，用于配合copytree()方法，传递文件 将会被忽略，不会被拷贝 patterns：文件名称（元祖）
shutil.copytree(src, dst, symlinks=False, ignore=None)	将src文件夹里的所有内容拷贝至dst文件夹
shutil.rmtree(path, ignore_errors=False, onerror=None)	ignore_errors：是否忽略错误，默认False onerror：定义错误处理函数，需传递一个可 执行的处理函数，该处理函数接收三个参数： 函数、路径和excinfo
shutil.move(src, dst)	将src移动至dst目录下。若dst目录不存在， 则效果等同于src改名为dst。若dst目录存在， 将会把src文件夹的所有内容移动至该目录下面
shutil.disk_usage(path)	获取当前目录所在硬盘使用情况

常用方法	方法简介
<code>shutil.chown(path, user=None, group=None)</code>	修改路径指向的文件或文件夹的所有者或分组 path : 路径 user : 所有者, 传递user的值必须是真实的, 否则将报错no such user group : 分组, 传递group的值必须是真实的, 否则将报错no such group
<code>shutil.which(cmd, mode=os.F_OK os.X_OK, path=None)</code>	获取给定的cmd命令的可执行文件的路径
<code>shutil.make_archive(base_name, format, root_dir, ...)</code>	生成压缩文件 base_name : 压缩文件的文件名, 不含扩展名, 因为会根据压缩格式生成相应的扩展名 format : 压缩格式 root_dir : 将制定文件夹进行压缩
<code>shutil.get_archive_formats()</code>	获取支持的压缩文件格式。目前支持的有 : tar、zip、gztar、bztar、xztar
<code>shutil.unpack_archive(filename, extract_dir=None, format=None)</code>	解压操作。 filename : 文件路径 extract_dir : 解压至的文件夹路径。不存在则自动生成 format : 解压格式, 默认为None, 会根据扩展名自动选择解压格式
<code>shutil.get_unpack_formats()</code>	获取支持的解压文件格式。目前支持的有 : tar、zip、gztar、bztar和xztar

# string模块




|||||