

Python OS 文件/目录方法

os 模块提供了非常丰富的方法用来处理文件和目录。常用的方法如下表所示：

序号	方法及描述
1	os.access(path, mode) 检验权限模式
2	os.chdir(path) 改变当前工作目录
3	os.chflags(path, flags) 设置路径的标记为数字标记。
4	os.chmod(path, mode) 更改权限
5	os.chown(path, uid, gid) 更改文件所有者
6	os.chroot(path) 改变当前进程的根目录
7	os.close(fd) 关闭文件描述符 fd
8	os.closerange(fd_low, fd_high) 关闭所有文件描述符，从 fd_low (包含) 到 fd_high (不包含), 错误会忽略
9	os.dup(fd)

复制文件描述符 fd

10 [`os.dup2\(fd, fd2\)`](#)

将一个文件描述符 fd 复制到另一个 fd2

11 [`os.fchdir\(fd\)`](#)

通过文件描述符改变当前工作目录

12 [`os.fchmod\(fd, mode\)`](#)

改变一个文件的访问权限，该文件由参数fd指定，参数mode是Unix下的文件访问权限。

13 [`os.fchown\(fd, uid, gid\)`](#)

修改一个文件的所有权，这个函数修改一个文件的用户ID和用户组ID，该文件由文件描述符fd指定。

14 [`os.fdatasync\(fd\)`](#)

强制将文件写入磁盘，该文件由文件描述符fd指定，但是不强制更新文件的状态信息。

15 [`os.fdopen\(fd\[, mode\[, bufsize\]\]\)`](#)

通过文件描述符 fd 创建一个文件对象，并返回这个文件对象

16 [`os.fpathconf\(fd, name\)`](#)

返回一个打开的文件的系统配置信息。name为检索的系统配置的值，它也许是一个定义系统值的字符串，这些名字在很多标准中指定（POSIX.1, Unix 95, Unix 98, 和其它）。

17 [`os.fstat\(fd\)`](#)

返回文件描述符fd的状态，像stat()。

18 [`os.fstatvfs\(fd\)`](#)

返回包含文件描述符fd的文件的文件系统的信息，像 statvfs()

19 [`os.fsync\(fd\)`](#)

强制将文件描述符为fd的文件写入硬盘。

20 [`os.ftruncate\(fd, length\)`](#)

裁剪文件描述符fd对应的文件, 所以它最大不能超过文件大小。

21 [`os.getcwd\(\)`](#)

返回当前工作目录

22 [`os.getcwdu\(\)`](#)

返回一个当前工作目录的Unicode对象

23 [`os.isatty\(fd\)`](#)

如果文件描述符fd是打开的, 同时与tty(-like)设备相连, 则返回true, 否则False。

24 [`os.lchflags\(path, flags\)`](#)

设置路径的标记为数字标记, 类似 chflags(), 但是没有软链接

25 [`os.lchmod\(path, mode\)`](#)

修改连接文件权限

26 [`os.lchown\(path, uid, gid\)`](#)

更改文件所有者, 类似 chown, 但是不追踪链接。

27 [`os.link\(src, dst\)`](#)

创建硬链接, 名为参数 dst, 指向参数 src

28 [`os.listdir\(path\)`](#)

返回path指定的文件夹包含的文件或文件夹的名字的列表。

29 [`os.lseek\(fd, pos, how\)`](#)

设置文件描述符 fd 当前位置为 pos, how 方式修改: SEEK_SET 或者 0 设置从文件开始的计算的 pos; SEEK_CUR 或者 1 则从当前位置计算; os.SEEK_END 或者 2 则从文件尾部开始. 在 unix , Windows 中有效

30 [`os.lstat\(path\)`](#)

像 stat(), 但是没有软链接

31 [`os.major\(device\)`](#)

从原始的设备号中提取设备 major 号码 (使用 stat 中的 st_dev 或者 st_rdev field)。

32 [`os.makedev\(major, minor\)`](#)

以 major 和 minor 设备号组成一个原始设备号

33 [`os.makedirs\(path\[, mode\]\)`](#)

递归文件夹创建函数。像 mkdir(), 但创建的所有 intermediate-level 文件夹需要包含子文件夹。

34 [`os.minor\(device\)`](#)

从原始的设备号中提取设备 minor 号码 (使用 stat 中的 st_dev 或者 st_rdev field)。

35 [`os.mkdir\(path\[, mode\]\)`](#)

以数字 mode 的 mode 创建一个名为 path 的文件夹. 默认的 mode 是 0777 (八进制)。

36 [`os.mkfifo\(path\[, mode\]\)`](#)

创建命名管道, mode 为数字, 默认为 0666 (八进制)

37 [`os.mknod\(filename\[, mode=0600, device\]\)`](#)

创建一个名为 filename 文件系统节点 (文件, 设备特别文件或者命名 pipe)。

38 [`os.open\(file, flags\[, mode\]\)`](#)

打开一个文件, 并且设置需要的打开选项, mode 参数是可选的

39 [`os.openpty\(\)`](#)

	打开一个新的伪终端对。返回 pty 和 tty 的文件描述符。
40	<code>os.pathconf(path, name)</code> 返回相关文件的系统配置信息。
41	<code>os.pipe()</code> 创建一个管道。返回一对文件描述符(r, w) 分别为读和写
42	<code>os.popen(command[, mode[, bufsize]])</code> 从一个 command 打开一个管道
43	<code>os.read(fd, n)</code> 从文件描述符 fd 中读取最多 n 个字节，返回包含读取字节的字符串，文件描述符 fd 对应文件已达到结尾, 返回一个空字符串。
44	<code>os.readlink(path)</code> 返回软链接所指向的文件
45	<code>os.remove(path)</code> 删除路径为path的文件。如果path 是一个文件夹，将抛出OSError; 查看下面的rmdir()删除一个 directory。
46	<code>os.removedirs(path)</code> 递归删除目录。
47	<code>os.rename(src, dst)</code> 重命名文件或目录，从 src 到 dst
48	<code>os.renames(old, new)</code> 递归地对目录进行更名，也可以对文件进行更名。
49	<code>os.rmdir(path)</code>

删除path指定的空目录，如果目录非空，则抛出一个OSError异常。

50 [`os.stat\(path\)`](#)

获取path指定的路径的信息，功能等同于C API中的stat()系统调用。

51 [`os.stat_float_times\(\[newvalue\]\)`](#)

决定stat_result是否以float对象显示时间戳

52 [`os.statvfs\(path\)`](#)

获取指定路径的文件系统统计信息

53 [`os.symlink\(src, dst\)`](#)

创建一个软链接

54 [`os.tcgetpgrp\(fd\)`](#)

返回与终端fd（一个由os.open()返回的打开的文件描述符）关联的进程组

55 [`os.tcsetpgrp\(fd, pg\)`](#)

设置与终端fd（一个由os.open()返回的打开的文件描述符）关联的进程组为pg。

56 [`os.tempnam\(\[dir\[, prefix\]\]\)`](#)

返回唯一的路径名用于创建临时文件。

57 [`os.tmpfile\(\)`](#)

返回一个打开的模式为(w+b)的文件对象。这文件对象没有文件夹入口，没有文件描述符，将会自动删除。

58 [`os.tmpnam\(\)`](#)

为创建一个临时文件返回一个唯一的路径

59 [`os.ttyname\(fd\)`](#)

返回一个字符串，它表示与文件描述符fd 关联的终端设备。如果fd 没有与终端设备关联，则引发一个异常。

60	<code>os.unlink(path)</code> 删除文件路径
61	<code>os.utime(path, times)</code> 返回指定的path文件的访问和修改的时间。
62	<code>os.walk(top[, topdown=True[, onerror=None[, followlinks=False]])</code> 输出在文件夹中的文件名通过在树中游走，向上或者向下。
63	<code>os.write(fd, str)</code> 写入字符串到文件描述符 fd中. 返回实际写入的字符串长度
64	<code>os.path 模块</code> 获取文件的属性信息。

参考地址：

- <http://kuanghy.github.io/python-os/>
- http://python.usyiyi.cn/python_278/library/os.html

← Python File writelines() 方法

Python os.access() 方法 →

 点我分享笔记