

查看文件和目录

方法名	说明
os.getcwd()	返回当前工作目录
os.path.realpath(path)	返回path的真实路径
os.stat(path)	获取path指定的路径的信息
os.path.getsize(path)	返回文件大小，如果文件不存在就返回错误
os.path.getctime(path)	返回文件 path 创建时间
os.path.getatime(path)	返回最近访问时间（浮点型秒数）
os.path.getmtime(path)	返回最近文件修改时间

补充：

1. os.stat(path)语法说明

获取path指定的路径的信息。返回信息如下：

stat 结构：

- **st_mode:** inode 保护模式
- **st_ino:** inode 节点号。
- **st_dev:** inode 驻留的设备。
- **st_nlink:** inode 的链接数。
- **st_uid:** 所有者的用户ID。
- **st_gid:** 所有者的组ID。
- **st_size:** 普通文件以字节为单位的大小；包含等待某些特殊文件的数据。
- **st_atime:** 上次访问的时间。
- **st_mtime:** 最后一次修改的时间。
- **st_ctime:** 由操作系统报告的"ctime"。在某些系统上（如Unix）是最新的元数据更改的时间，在其它系统上（如Windows）是创建时间（详细信息参见平台的文档）。

创建目录和文件

方法名	说明
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)。
os.makedirs(path)	递归文件夹创建函数
open(path,'w')	创建文件
os.path.exists	判断文件或文件夹是否存在

删除文件或目录

方法名	说明
os.remove(path)	删除文件 path. 如果path是一个目录， 抛出 OSError错误。如果要删除目录，请使用rmdir()
os.unlink(path)	同os.remove()
os.rmdir(path)	删除目录 path， 要求path必须是个空目录， 否则抛出OSError错误
os.removedirs(path)	递归地删除目录。类似于rmdir(), 如果子目录被成功删除， removedirs() 将会删除父目录； 但子目录没有成功删除， 将抛出错误

修改目录

方法名	说明
os.rename(src, dst)	重命名文件或目录， 从 src 到 dst
os.chdir(path)	改变当前工作目录
os.access(path, mode)	用当前的uid/gid尝试访问路径
os.chmod(path, mode)	更改权限
os.chown(path, uid, gid)	更改文件所有者

补充：

1. os.access()用当前的uid/gid尝试访问路径.

access()方法语法格式如下：

```
os.access(path, mode);
```

参数：

- **path** -- 要用来检测是否有访问权限的路径。
- **mode** -- mode为F_OK, 测试存在的路径, 或者它可以是包含R_OK, W_OK和X_OK或者R_OK, W_OK和X_OK其中之一或者更多。
 - **os.F_OK**: 作为access()的mode参数, 测试path是否存在。
 - **os.R_OK**: 包含在access()的mode参数中, 测试path是否可读。
 - **os.W_OK** 包含在access()的mode参数中, 测试path是否可写。
 - **os.X_OK** 包含在access()的mode参数中, 测试path是否可执行。

返回值：

如果允许访问返回 True , 否则返回False。

2. os.chmod() 方法用于更改文件或目录的权限。

chmod()方法语法格式如下：

```
os.chmod(path, mode)
```

参数：

- **path** -- 文件名路径或目录路径。
- **flags** -- 可用以下选项按位或操作生成, 目录的读权限表示可以获取目录里文件名列表, , 执行权限表示可以把工作目录切换到此目录, 删除添加目录里的文件必须同时有写和执行权限, 文件权限以用户id->组id->其它顺序检验,最先匹配的允许或禁止权限被应用。
 - **stat.S_IXOTH**: 其他用户有执行权0o001
 - **stat.S_IWOTH**: 其他用户有写权限0o002
 - **stat.S_IROTH**: 其他用户有读权限0o004
 - **stat.S_IRWXO**: 其他用户有全部权限(权限掩码)0o007
 - **stat.S_IXGRP**: 组用户有执行权限0o010
 - **stat.S_IWGRP**: 组用户有写权限0o020
 - **stat.S_IRGRP**: 组用户有读权限0o040
 - **stat.S_IRWXG**: 组用户有全部权限(权限掩码)0o070
 - **stat.S_IXUSR**: 拥有者具有执行权限0o100
 - **stat.S_IWUSR**: 拥有者具有写权限0o200
 - **stat.S_IRUSR**: 拥有者具有读权限0o400
 - **stat.S_IRWXU**: 拥有者有全部权限(权限掩码)0o700
 - **stat.S_ISVTX**: 目录里文件目录只有拥有者才可删除更改0o1000
 - **stat.S_ISGID**: 执行此文件其进程有效组为文件所在组0o2000
 - **stat.S_ISUID**: 执行此文件其进程有效用户为文件所有者0o4000
 - **stat.S_IREAD**: windows下设为只读
 - **stat.S_IWRITE**: windows下取消只读

返回值：

该方法没有返回值。

示例：

增加单个权限

修改权限会删除之前所有的权限,进行重新的更改

```
os.chmod('abc',stat.S_IREAD)
```

增加多个权限

多个权限使用+号进行连接

```
os.chmod('abc',stat.S_IREAD+stat.S_IWOTH)
```

判断目录或文件

方法名	说明
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.abs(path)	判断是否为绝对路径

合并和分割目录

方法名	说明
os.path.join(path1[, path2[, ...]])	把目录和文件名合成一个路径
os.path.split(path)	把路径分割成 dirname 和 basename，返回一个元组
os.path.splitext(path)	分割路径，返回路径名和文件扩展名的元组

子目录相关操作

方法名	说明
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
os.walk(path, visit, arg)	输出在文件夹中的文件名通过在树中游走，向上或者向下

补充：

1.walk()方法语法格式如下：

```
os.walk(top[, topdown=True[, onerror=None[, followlinks=False]]])
```

参数：

- **top** -- 是你所要遍历的目录的地址, 返回的是一个三元组(root,dirs,files)。
 - root 所指的是当前正在遍历的这个文件夹的本身的地址
 - dirs 是一个 list , 内容是该文件夹中所有的目录的名字(不包括子目录)
 - files 同样是 list , 内容是该文件夹中所有的文件(不包括子目录)
- **topdown** --可选, 为 True, 则优先遍历 top 目录, 否则优先遍历 top 的子目录(默认为开启)。如果 topdown 参数为 True, walk 会遍历top文件夹, 与top 文件夹中每一个子目录。
- **onerror** -- 可选, 需要一个 callable 对象, 当 walk 需要异常时, 会调用。
- **followlinks** -- 可选, 如果为 True, 则会遍历目录下的快捷方式(linux 下是软连接 symbolic link) 实际所指的目录(默认关闭), 如果为 False, 则优先遍历 top 的子目录。

返回值：

该方法没有返回值。

shutil高级模块

方法名	说明
shutil.copyfileobj(fsrc, fdst)	copy文件内容到另一个文件，可以copy指定大小的内容
shutil.copyfile(src,dst)	<code>copyfile()</code> 使用低级的方法 <code>copyfileobj()</code> 实现。然而传入 <code>copyfile()</code> 的参数是文件名称，而传入 <code>copyfileobj()</code> 是打开的文件描述符。可选的第三个参数是用来设置读取块的大小。
shutil.copy(src,dst)	复制文件的内容以及权限，先copyfile后copymode
shutil.copy2(src,dst)	复制文件的内容以及文件的所有状态信息。先copyfile后copystat
shutil.copytree(src, dst)	递归的复制文件内容及状态信息
shutil.rmtree(path)	递归地删除文件
shutil.move(src, dst)	递归的移动文件
make_archive(base_name, format)	压缩打包