

# 文件和目录操作

# os模块（第10章）

- os模块：与操作系统相关的可移植(portable)的函数，包括进程管理、信号处理、底层的文件读写以及文件和目录的操作等

函数	功能
getcwd()	返回当前工作目录
chdir(path)	改变工作目录到path
mkdir(path, mode=0o777)	创建目录，要求path的父目录存在
makedirs(name, mode=0o777)	创建目录，根据需要自动创建中间缺失的目录
rmdir(path)	删除目录，只有空目录才可以删除
removedirs(path)	删除目录以及目录的父目录，如此继续，直到非空目录或path的起始点为止
remove(path); unlink(path)	删除文件（非目录），remove和unlink完成相同的功能
rename(src, dst)	将原来的目录或者文件的路径名从src改为dst，一般不允许跨越文件系统
replace(src, dst)	与rename的功能类似，存在时会直接覆盖。
stat(path)	返回path对应的文件或目录的属性，返回一个stat_result对象，可通过属性访问具体的信息
system(cmd)	执行外部程序cmd，建议采用模块subprocess
startfile(path, operation='open')	使用关联的应用程序打开文件path，或者启动应用程序path

# OS模块

- 列出当前目录下，所有扩展名为.pyc的文件

[name for name in os.listdir() if name.endswith('.pyc')]

[item.name for item in os.scandir() if item.name.endswith('.py')]

函数	功能
listdir(path='.')	返回一个列表，其中的元素是目录path中的文件和目录的名字
scandir(path='.')	返回一个迭代器对象，其中的元素为DirEntry对象，对应着目录path中的文件和目录。支持上下文协议，相比listdir()，提供缓冲支持，效率更高
walk(top, topdown=True)	遍历top开始的目录树。返回一个迭代器，其元素为目录树中的目录节点，包括(dirpath, dirnames, filenames)，分别给出了节点的路径名，该目录中的子目录名列表，该目录中的文件名列表。topdown=True时，首先输出父目录，然后子目录。topdown=False时，即bottom-up，首先输出子目录，然后父目录

# 递归遍历目录树

- 采用递归函数实现

walk\_demo.py

```
def visit_tree(path):  
    if not os.path.isdir(path):  
        return  
    for lists in os.listdir(path):  
        sub_path = os.path.join(path, lists)  
        print(sub_path)  
        if os.path.isdir(sub_path):  
            visit_tree(sub_path)
```

- 采用walk实现

```
def visit_tree2(path):  
    if not os.path.isdir(path):  
        return  
    list_dirs = os.walk(path)  
    for root, dirs, files in list_dirs:  
        for d in dirs:  
            print(os.path.join(root, d))  
        for f in files:  
            print(os.path.join(root, f))
```

# os.path模块

	函数	功能
目录和文件	abspath(path)	返回path对应的绝对路径名
	basename(path)	返回path的最后一个组成部分
	dirname(path)	返回path的目录部分
	realpath(path, *, strict=False)	返回path对应的绝对路径名，会解析符号连接，如果支持strict=True时，会检查path是否存在
	normpath(path)	返回对path规范化后的路径名，如多个路径分隔符代替以一个路径分隔符
分割和合并	join(path, *paths)	将两个或多个path通过路径分隔符合并在一起
	split(path)	将path根据最后一个路径分隔符分割成两个部分(dirname, basename)
	splittext(path)	将path根据扩展分隔符分割成(root, ext)
	splitdrive(path)	将path分割成(driver_or_unc, path)
文件状态	exists(path)	判断path是否存在
	getsize(path)	文件的大小（字节为单位）
	getatime(path) getmtime(path) getctime(path)	获得path的访问时间、修改时间和状态改变时间
	isdir(path) isfile(path) islink(path)	判断是否是目录、普通文件、符号连接

# shutil模块

- 文件拷贝、移动和删除
- 归档压缩和解压缩

函数	功能
copyfile(src, dst)	复制文件src到dst
copy(src, dst)	如果dst是目录，复制文件src到目录dst。否则等价于copyfile
copytree(src, dst)	递归复制目录src到dst
rmtree(path)	递归删除目录path
move(src, dst)	类似于os.rename，但允许src和dst在不同的文件系统
make_archive(basename, fmt, root_dir, base_dir)	归档压缩，fmt可取值zip/tar/gztar等，产生的归档文件为basename+fmt指出的后缀。root_dir为归档之前临时的工作目录, base_dir为要归档的文件所在的目录
unpack_archive(path, extract_dir, fmt=None)	展开归档压缩文件。path为待展开的文件，extract_dir表示在那个目录展开，缺省根据path的后缀决定归档格式，也可通过fmt明确指定

## pathlib模块（python3.4引入）

- 关于文件和目录的操作，分布在多个模块中，包括os、os.path、glob和shutil。pathlib将这些和文件、目录相关的操作有机地整合在一起，不是通过字符串表示文件系统路径，而是引入了Path等对象，表示文件系统路径。

