

Python 系统操作文件速查表

一、导入模块

使用 `os` 和 `shutil` 模块来完成系统管理相关操作。

```
import os  
import shutil
```

二、创建目录

创建单个目录

```
os.mkdir('test')
```

创建多级目录

```
os.makedirs('A/B/C')
```

三、切换和获取当前工作目录

获取当前工作目录

```
current_dir = os.getcwd()  
print(current_dir)
```

切换工作目录

```
os.chdir('path/to/directory')
```

四、目录内容列表

获取目录下所有文件和子目录

```
contents = os.listdir('directory_path')  
for item in contents:  
    print(item)
```

五、判断文件/目录是否存在

判断文件是否存在

```
file_exists = os.path.isfile('file_path')
if file_exists:
    print('文件存在')
else:
    print('文件不存在')
```

判断目录是否存在

```
dir_exists = os.path.isdir('directory_path')
if dir_exists:
    print('目录存在')
else:
    print('目录不存在')
```

六、测试文件类型

判断是否为符号链接

```
is_symlink = os.path.islink('path')
if is_symlink:
    print('是符号链接')
else:
    print('不是符号链接')
```

七、文件的日期和大小

获取文件的修改时间（以秒为单位的时间戳）

```
modification_time = os.path.getmtime('file_path')
import time
print(time.ctime(modification_time)) # 转换为可读性更好的时间格式
```

获取文件大小（以字节为单位）

```
file_size = os.path.getsize('file_path')
print(f'文件大小: {file_size} 字节')
```

八、文件和目录的删除

删除文件

```
os.remove('file_path')
```

删除空目录

```
os.rmdir('directory_path')
```

删除非空目录及其所有内容

```
shutil.rmtree('directory_path')
```

九、文件和目录的复制

复制文件

```
shutil.copy('source_file_path', 'destination_file_path')
# 如果要保留文件的元数据（如权限、修改时间等），使用 copy2
shutil.copy2('source_file_path', 'destination_file_path')
```

复制目录及其所有内容

```
shutil.copytree('source_directory_path', 'destination_directory_path')
```

十、重命名和移动

重命名文件或目录

```
os.rename('old_name', 'new_name')
```

移动文件或目录

```
shutil.move('source_path', 'destination_path')
# 移动操作本质上也是重命名，如果目标路径是不同的目录，就是移动
```

十一、注意事项

1. **权限问题**: 所有文件和目录操作都可能受到系统权限的限制。确保运行 Python 脚本的用户具有执行相关操作的权限。
2. **路径处理**: 在不同操作系统上，路径分隔符可能不同 (Windows 上是 \, Linux 和 macOS 上是 /)。为了跨平台兼容性，建议使用 `os.path.join()` 来构建路径。例如：

```
path = os.path.join('parent_dir', 'child_dir', 'file.txt')
```

3. **shutil 操作**: `shutil` 模块的一些函数（如 `rmtree` 和 `copytree`）执行的是递归操作，使用时要谨慎，避免误删或误复制大量数据。