

`os` 和 `shutil` 是 Python 中用于文件和目录操作的重要模块。以下是它们的基本用法：

1. os 模块

`os` 模块提供了与操作系统交互的方法，如文件和目录操作、环境变量、进程管理等。

1.1 获取当前工作目录

```
import os

cwd = os.getcwd() # 获取当前工作目录
print("当前工作目录:", cwd)
```

1.2 切换目录

```
os.chdir('/path/to/directory') # 切换到指定目录
```

1.3 创建目录

```
os.mkdir('test_dir') # 创建单层目录
os.makedirs('test_dir/sub_dir') # 递归创建多级目录
```

1.4 删除目录

```
os.rmdir('test_dir') # 只能删除空目录
os.removedirs('test_dir/sub_dir') # 递归删除多级空目录
```

1.5 创建和删除文件

```
with open('test.txt', 'w') as f:
    f.write('Hello, world!')
```

```
os.remove('test.txt') # 删除文件
```

1.6 列出目录下的文件

```
files = os.listdir('.') # 列出当前目录的所有文件和文件夹
print(files)
```

1.7 检测文件和目录

```
os.path.exists('test.txt') # 是否存在
os.path.isfile('test.txt') # 是否是文件
os.path.isdir('test_dir') # 是否是目录
```

1.8 获取文件信息

```
size = os.path.getsize('test.txt') # 获取文件大小
print(f"文件大小: {size} 字节")
```

2. shutil 模块

shutil 模块主要用于高级文件和目录操作，如复制、移动和删除文件夹等。

2.1 复制文件

```
import shutil

shutil.copy('test.txt', 'copy_test.txt') # 复制文件
shutil.copy2('test.txt', 'copy_test.txt') # 复制文件并保留元数据
```

2.2 复制目录

```
shutil.copytree('test_dir', 'backup_dir') # 复制整个目录
```

2.3 移动文件或目录

```
shutil.move('test.txt', 'new_folder/test.txt') # 移动文件
```

2.4 删除目录

```
shutil.rmtree('test_dir') # 递归删除整个目录
```

2.5 获取磁盘使用情况

```
disk_usage = shutil.disk_usage("/")  
print(f"总空间: {disk_usage.total}, 已用: {disk_usage.used}, 可用: {disk_usage.free}")
```

3. 示例：批量移动文件

```
import os  
import shutil  
  
src_folder = "source_folder"  
dst_folder = "destination_folder"  
  
if not os.path.exists(dst_folder):  
    os.makedirs(dst_folder)  
  
for file_name in os.listdir(src_folder):  
    full_file_name = os.path.join(src_folder, file_name)  
    if os.path.isfile(full_file_name):  
        shutil.move(full_file_name, dst_folder)
```

这样，`source_folder` 下的所有文件都会被移动到 `destination_folder`。

总结

- `os` 适用于基本文件和目录操作，如创建、删除、修改路径等。
- `shutil` 适用于高级文件操作，如复制、移动、删除整个目录等。

你有什么具体需求，可以告诉我，我帮你写代码！ 😊