

## 5.13 获取文件夹中的文件列表

### 问题

你想获取文件系统中某个目录下的所有文件列表。

### 解决方案

使用 `os.listdir()` 函数来获取某个目录中的文件列表：

```
import os
names = os.listdir('somedir')
```

结果会返回目录中所有文件列表，包括所有文件，子目录，符号链接等等。如果你需要通过某种方式过滤数据，可以考虑结合 `os.path` 库中的一些函数来使用列表推导。比如：

```
import os.path

# Get all regular files
names = [name for name in os.listdir('somedir')
         if os.path.isfile(os.path.join('somedir', name))]

# Get all dirs
dirnames = [name for name in os.listdir('somedir')
            if os.path.isdir(os.path.join('somedir', name))]
```

字符串的 `startswith()` 和 `endswith()` 方法对于过滤一个目录的内容也是很有用的。比如：

```
pyfiles = [name for name in os.listdir('somedir')
           if name.endswith('.py')]
```

对于文件名的匹配，你可能会考虑使用 `glob` 或 `fnmatch` 模块。比如：

```
import glob
pyfiles = glob.glob('somedir/*.py')

from fnmatch import fnmatch
pyfiles = [name for name in os.listdir('somedir')
           if fnmatch(name, '*.py')]
```

### 讨论

获取目录中的列表是很容易的，但是其返回结果只是目录中实体名列表而已。如果你还想获取其他的元信息，比如文件大小，修改时间等等，你或许还需要使用到 `os.path` 模块中的函数或者 `os.stat()` 函数来收集数据。比如：

```
# Example of getting a directory listing

import os
import os.path
import glob

pyfiles = glob.glob('*.py')

# Get file sizes and modification dates
name_sz_date = [(name, os.path.getsize(name), os.path.getmtime(name))
                 for name in pyfiles]
for name, size, mtime in name_sz_date:
    print(name, size, mtime)

# Alternative: Get file metadata
file_metadata = [(name, os.stat(name)) for name in pyfiles]
for name, meta in file_metadata:
    print(name, meta.st_size, meta.st_mtime)
```

最后还有一点要注意的就是，有时候在处理文件名编码问题时候可能会出现一些问题。通常来讲，函数 `os.listdir()` 返回的实体列表会根据系统默认的文件名编码来解码。但是有时候也会碰到一些不能正常解码的文件名。关于文件名的处理问题，在5.14和5.15小节有更详细的讲解。