

关于本节中要掌握的内容

- 文件路径
- 文件读写

文件路径

文件路径是文件在计算机中的位置。例如：

```
# windows中的文件路径  
C:\study\python-code\py_study\04-file\pdf.md  
# unix系的文件路径, 包括linux、Mac OS  
/home/peng/study/qemu-system-need/run.sh
```

- windows中的文件路径的根目录是盘符号，如上图的'C:'；unix系的文件路径的根目录是 '/'
- windows、MAC OS都是不区分大小写的；linux是区分大小写的
- windows使用"作为文件夹之间的分隔符；unix系使用 '/' 作为文件夹之间的分隔符

相对路径和绝对路径

下面是某个目录的所有文件信息，假设我们当前处于/home/tom/study

```
study
├── a
│   └── 1.txt
├── b
│   ├── 2.txt
│   └── 3.bin
└── c
    ├── 4.txt
    └── d
        └── 1.xz
4 directories, 5 files
```

相对路径，一般是相对于当前工作目录的某个文件路径。例如，1.txt的相对路径就是'a/1.txt'或者'./a/1.txt'，这里的'.'指的就是当前目录的路径，类似于'/home/tom/study'

绝对路径，指的文件从根目录开始的位置。例如1.txt的绝对路径就是'/home/tom/study/a/1.txt'

所有文件的相对路径和绝对路径

- 相对路径

```
a/1.txt  
b/2.txt  
b/3.bin  
c/4.txt  
d/1/xz
```

- 绝对路径

```
home/tom/study/a/1.txt  
home/tom/study/b/2.txt  
home/tom/study/b/3.bin  
home/tom/study/c/4.txt  
home/tom/study/d/1/xz
```

使用Python创建示例中的所有文件夹

1.在当前目录下创建一个名叫study的文件夹

```
from pathlib import Path  
Path('study').mkdir()
```

2.分别创建a、b、c、d文件夹

```
study_path = Path('study')  
(study_path / 'a').mkdir()  
(study_path / 'b').mkdir()  
(study_path / 'c' / 'd').mkdir(parents=True)
```

- 使用'/'运算符来拼接路径
- 在mkdir中使用parents=True来自动创建中间文件夹

自己使用相对路径创建所有文件夹

使用绝对路径创建示例中的所有文件夹

1. 获取当前目录所在的路径

```
from pathlib import Path  
path = Path.cwd()
```

2. 创建study文件夹和其他文件夹

```
study_path = path / 'study'  
(study_path / 'a').mkdir(parents=True, exist_ok=True)  
(study_path / 'b').mkdir(parents=True, exist_ok=True)  
(study_path / 'c' / 'd').mkdir(parents=True, exist_ok=True)
```

- 使用cwd来获取当前工作路径
- 在mkdir中使用了exist_ok来确保文件夹已存在不会报错

自己使用绝对路径创建所有文件夹

家目录

一般来说，计算机上的每个用户都有自己的专属文件夹，用以工作，这个专属文件夹一般称为家目录，例如：

- windows

```
print(Path.home())  
# C:\Users\peng
```

- linux

```
print(Path.home())  
# /home/peng
```

文件路径的组成

- 一个文件的路径由以下三部分组成：

1.anchor, 文件系统的根文件夹

2.parent, 该文件的文件夹

3.name, 文件的名称

```
p = Path.cwd()
print(p.anchor)
print(p.parent)
print(p.name)
```

- 文件名由以下两部分组成：

1.stem, 文件的基本名称

2.suffix, 文件的后缀名

文件读写

对于文件，我们可以将它的类型粗略分为两种：

- 1.文本类型，只包含基本的文本字符，是给人类看的文件，例如：txt、py文件、md文件....
- 2.二进制类型，通常可包含所有的二进制，是给机器看的文件，例如：word、ppt、PDF、elf。只可以由特定的文件打开，如果以文本文件打开，会发现它就是一堆乱码。

这里的文件读写，我们讨论的都是文本类型的。

对于文件的权限，我们可以粗略的分别三种：

- 1.读，代表对文件有读的权限
- 2.写，代表对文件有写的权限
- 3.执行，代表文件有执行的权限

打开一个文件并写入内容

1.打开文件

```
f = open(Path.cwd() / 'study' / 'a' / '1.txt', 'w', encoding='utf-8')
```

2.写入文本

```
f.write('Hello, world 1!\n')  
f.write('Hello, world 2!\n')
```

3.关闭文件

```
f.close()
```

- encoding设置编码为utf-8，才可以写入中文
- w会覆盖原有文件的内容哦

自己在文件尝试写入任意内容

在写入的文件中追加内容

1.打开文件

```
f = open(Path.cwd() / 'study' / 'a' / '1.txt','a', encoding='utf-8')
```

2.追加文本

```
f.write('Hello, world 3!\n')
```

3.关闭文件

```
f.close()
```

- a会在原有的文件内容后追加，而不会覆盖哦

读取文件

1.遍历文件并输出

```
with open(Path.cwd() / 'study' / 'a' / '1.txt',"r", encoding='utf-8') as f:  
    for idx,line in enumerate(f):  
        print(f"{idx} {line}",end='')
```

- r读取文件
- 因为文件行已经包含了换行符，所以print的end要设置为空字符
- 如果需要同时写和读，可以使用w+或者a+

作业

- 1.尝试自己完成倍投法则，并将倍投法则的数据记录在文本中，然后再次将数据读出来，并输出图像
- 2.你计划为学生制作10道选择题并为20个学生随机打乱它们的顺序。你需要保存20份打乱顺序的试卷(n.paper)，以及每套试卷的答案(n.answer)，确保每个问题的每个选项也是打乱的。