

Table of Contents

前言	1.1
python基础	1.2
文件读写	1.2.1
OS模块	1.2.2
多进程和多线程	1.2.3
异常处理	1.2.4
数据结构与算法	1.3
排序算法	1.3.1
类	1.3.2
机器学习	1.4

前言

python文件处理

open 方法

Python `open()` 方法用于打开一个文件，并返回文件对象，在对文件进行处理过程都需要使用到这个函数，如果该文件无法被打开，会抛出 `OSError`。

注意：使用 `open()` 方法一定要保证关闭文件对象，即调用 `close()` 方法。

`open()` 函数常用形式是接收两个参数：`file, mode, encoding`。

```
open(file, mode='r', encoding=None)
```

`mode` 参数常用的有

- `r` 只读
- `a+` 打开一个文件用于读写。如果该文件已存在，文件指针将会放在文件的结尾。文件打开时会是追加模式。如果该文件不存在，创建新文件用于读写。
- `w+` 打开一个文件用于读写。如果该文件已存在则打开文件，并从开头开始编辑，即原有内容会被删除。如果该文件不存在，创建新文件。

file 对象

- `file.close()` 关闭文件
- `file.seek()` 0代表从头开始，1代表当前位置，2代表文件最末尾位置
- `file.readlines()`
- `file.writelines()`

OS模块

目录操作-增删改查

- `os.listdir()` 列出指定目录下所有的
- `os.makedirs()` 创建一个文件夹
- `os.rmdir()` 删除一个文件夹，但是必须是空文件夹

路径信息

- `os.path.abspath(path)` 显示当前绝对路径
- `os.path.dirname(path)` 返回该路径的父目录

```
os.path.dirname(os.path.abspath('hello.py'))
```

- `os.path.isfile(path)` 是文件则返回True,和`os.path.isdir(path)`相对应
- `os.path.join(path,name)` #连接目录与文件名或目录 结果为`path/name`
- `os.getcwd()` 显示当前python的工作目录

重命名

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