

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
面向对象编程	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)
```

排序算法

插入排序

插入排序是一种简单直观的排序算法。它的工作原理是通过构建有序序列，对于未排序数据，在已排序序列中从后向前扫描，找到相应位置并插入。

从第一个元素，如果第1个元素小于第0个元素，就把第1个元素和第0个元素交换（前2个位置已经排好序）

到第二个元素，如果第2个元素小于第1个元素，就把第2个元素和第1个元素交换，然后比较一直交换到前3个元素排好位置

以此类推....

```
def insert_sort(arr):  
    for i in range(1, len(arr)):  
        key = arr[i]  
        j = i  
        while j >= 1 and key <= arr[j - 1]:  
            arr[j] = arr[j - 1]  
            j = j - 1  
        arr[j] = key
```

希尔排序

选择排序

冒泡排序

归并排序

快速排序

Python的类

创建新类

使用 **class** 语句来创建一个新类，**class** 之后为类的名称并以冒号结尾，一般遵循习惯类的名称首字母大写。以下创建一个简单的类

```
class Student:
    def __init__(self, name, ID):
        self.name = name
        self.ID = ID
    def get_ID(self):
        return self.ID
```