

# day17-文件

模块：一个个py文件

文件：.py .doc .txt .md .exe 文件类型

文件操作-读写

I/O操作：

- 什么是 I o?
  - i: input 输入
  - o: output 输出
- 最常见 文件操作(文件读写)
  - 写：打开文件，写入内容，从内存 保存当前文件到 电脑磁盘中。o
  - 读：从磁盘中 读取文件 到 内存 Input

## • 文件操作 流程

三步：

- 打开文件操作对象
- 文件操作 读 写
- 关闭

## • 文件指针

文件指针 光标

## • open()

1.

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

- 功能：打开文件操作对象，返回IO操作流
- 参数
  - `file`：要打开的文件 路径 `path/url`
  - `mode`：打开文件模式，默认为 `r`。`r`：只读
  - `buffering`：缓存区大小设置，一般不设置，默认值即ok
    - `< 0`：交由系统自己来决定。
    - `0`：关闭缓存区
    - `>0`：手动自定义设置缓存区大小
  - `encoding`：编码方式设置。

## • 常用文件操作模式

模式	描述
<code>r</code>	以只读模式打开文件。文件的指针将会放在文件的开头，这是默认打开方式。如果文件不存在，则会抛出异常
<code>w</code>	以只写模式打开文件。如果文件存在则会发生覆盖。如果文件不存在，则会创建新文件
<code>a</code>	以追加模式打开文件。如果文件存在，文件指针将会放在文件末尾。如果文件不存在，则会创建新文件，进行写入
<code>r+</code>	以读写模式打开文件。文件指针将会放在开头。如果文件不存在，则会报错、
<code>w+</code>	以读写模式打开文件。如果文件存在则会覆盖。如果文件不存在，则创建新文件
<code>a+</code>	以读写模式打开文件。如果文件存在，文件指针将会放在文件末尾。如果不存在，创建新文件进行写入。
<code>rb</code>	以二进制只读模式打开
<code>rw</code>	以二进制只写模式打开
<code>ab</code>	以二进制追加模式打开

## • 路径

路径：表示文件所在位置

分类

- 相对路径：相对于当前文件路径去进行书写

111.txt

- 绝对路径：

完整路径：

- win系统：F:\python193\代码+笔记\day17\111.txt

- linux系统：

/home/usr/day17/111.txt

注意： 在使用绝对路径时，路径分隔符可能会识别为 转义字符，所以一定要注意，避免转义字符出现。可以使用 `r''` 原始字符串 或 `\\`

## • 文件读写

文件指针：

- 文件在进行读取内容的过程中，文件指针会随着读取字符逐一进行跟随移动。

### 1. 读

- `read([n=-1])`

- 功能：读取文件中所有内容。
- 默认情况下，不传参数，读取文件中所有内容。
- 可以指定参数，结合编码方式，utf-8,即读取的指定字符个数

- 一个字节占8位
- 一个英文占一个字节

- 一个中文占2个字节
- `readline()`
  - 功能：逐行读取文件内容
  - 读取内容是以行为单位
  - 注意：逐行读取也会识别到换行符 `'abc\n'`
- `readlines()`
  - 功能：逐行读取，读取到的每行字符串 形成一个列表。

从文件指针处开始，直到末尾。

- `tell()`
  - 功能：监听文件指针当前位置，以字节为单位。
  - 返回值：`int`类型，从0开始。开头首位为0
- `seek(offset)` 寻求 寻找 探索
  - 功能：修改 文件指针 设置文件指针`offset`(偏移值)
  - 文件指针偏移量 单位 字节
  - `seek(2,0)`
    - 表示 文件指针由原来 起始位置 移动到新位置 2

## 2. 写

- `write(s)`
  - 功能：写入 字符串
- `writelines(lines)`
  - 功能：写操作 写入 列表中所有的字符串元素
  - `lines`：列表 元素-字符串
    - 该参数可以为可迭代对象，但是注意，元素必须为字符串

- 文件对象属性

1. `name`

- 获取 文件对象的名字 路径

2. `mode`

- 获取 文件操作模式

3. `closed`

- 判断文件对象是否关闭
- 返回值：`bool True/False`

- 关闭

1. `close()`

- 功能：关闭
- 关闭文件操作IO流，释放系统资源
- 保存文件内容

2. `flush()`

- 功能：强制刷新 缓存区数据立即写入文件，文件IO流并没有关闭
- vs. `close()`

- 总结

`open()`

`read()` `readline()` `readlines()`

`write()` `writelines()`

`close()`

- 内置模块

time模块 date datetime  
random模块 sys模块 sys.exit()  
os模块 os.path模块