File 方法

1. open()方法

语法：**open(file, mode='r', buffering=-1, encoding=None,**
errors=None, newline=None, closefd=True, opener=None)

一般用法：**open(file, mode='r',encoding='utf-8')**

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

参数：

file：必选参数，文件路径（相对或绝对路径）

mode：文件打开模式

'r'	只读
'w'	打开一个文件只用于写入。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。
'x'	写模式，新建一个文件，如果该文件已存在则会报错
'a'	打开一个文件用于追加。如果该文件已存在，文件指针将会放在文件的结尾。如果该文件不存在，创建新文件进行写入。
'b'	二进制模式
't'	文本模式（默认），不用写，相对于二进制模式
'+'	打开一个文件进行更新(可读可写)

2. close()方法

语法： **fileObject.close()**

功能： 用于关闭文件 IO

3. flush()方法

语法： **fileObject.flush()**

功能： 刷新缓冲区，即将缓冲区中的数据立刻写入文件，同时清空缓冲区，不需要是被动的等待输出缓冲区写入。一般情况下，文件关闭后会自动刷新缓冲区，但有时你需要在关闭前刷新它，这时就可以使用 flush()方法。

4. fileno()方法

语法： **fileObject.fileno()**

功能： 返回文件描述符 fp

5. read()方法

语法： **fileobject.read(size=-1)**

功能： 从文当中最多读取 size 个字符（无论中文、英文），换行也算一个字符，当 size<0 时，读取到 EOF

6. readline()方法

语法： **fileobject.readline(size=-1)**

功能： 从文件读取整行，包括 "\n" 字符。如果指定了一个非负数的参数，则返回指定大小的字节数，包括 "\n" 字符。

7. readlines()方法

语法： **fileobject.readlines()**

功能： 读取所有行(直到结束符 EOF)并返回列表，每行包括换行符为列表的一个元素，当文档为空时，返回空列表

8. seek()方法

语法：**fileobject.seek(offset, whence=0)**

参数：

offset： 开始的偏移量，也就是代表需要移动偏移的字节数，如果是负数表示从倒数第几位开始。

whence： 给 offset 定义一个参数，表示要从哪个位置开始偏移；0 代表从文件开头开始算起，1 代表从当前位置开始算起，2 代表从文件末尾算起。

功能：移动文件读取指针到指定位置

9. tell()方法

语法： **fileobject.tell()**

功能： 返回文件指针位置

10. truncate()方法

语法： **fileobject.truncate(pos=None)**

功能： 无参数时，从开头截取到当前文件指针位置作为文件新内容，有参数时，从开头截取 pos 个字节，Windows 下换行符算 2 个字节，当 pos 大于文件字节大小时用空格字符在末尾填充到 pos 大小

11. write()方法

语法： **fileobject.write(str)**

功能： 向文件中写入指定字符串，如果文件打开模式带 b，那写入文件内容时，str (参数)要用 encode 方法转为 bytes 形式，否则报错

12. writelines()方法

语法： **fileObject.writelines(seq)**

功能： 将字符串序列写入文件