文件

1. 什么是文件
   计算机中的文件是以计算机硬盘为载体存在计算机上的信息集合
2. 文件可以是图片，视频，音乐等
   文件通常有文件扩展名，用于指示文件的类型
   .txt .py .exe .jpg .gif .mp4 .mp3 .avi .html .doc .ppt .pdf .md

计算机：内存（运存）+硬盘
运存：一旦断电，数据丢失，运行速度相当快---负责程序的运行
硬盘：即使断电，数据依然存在，运行速度慢---负责文件存储

- 打开一个文件

open():用于打开文件，返回一个file对象
1. open(name，[mode=r]，[buffering=-1])
   name:文件名--文件名+扩展名
   mode:打开文件的模式，默认是'r'---只读
   buffering:缓冲区---设置缓冲区大小
   强制：flush（清空缓存，并保存到硬盘中）
   0：关闭缓冲区
   1：打开缓冲区
   大于1：设置缓冲区的大小
   -1：缓冲区的大小交给系统管理（默认值）
2. IO： input Output
   input：从硬盘到系统
   output：从系统到硬盘
3. stream（流）
   可以想象成一个管道，数据就是管道中水

open的使用：
写入：

```
f=open('text1.txt','w')
f.write('从前有只羊\n爱跳墙\n终于跳过去\n墙外十只狼')
f.close()
```

读取：

```
f=open('text1.txt','r')
s=f.read()
print(s)
f.close()
```

模式：mode	描述
r	只读
w	只写，文件不存在，会创建新的，存在则覆盖之前的内容
x	只写，文件不存在创建新的，存在报错
a	追加模式，可读不存在则创建，存在则追加
r+	读写
w+	写读
x+	写读
a+	读写
rb	只读，以二进制形式
wb	只写以二进制形式
xb	只写，以二进制形式
ab	追加，以二进制形式

- File对象的属性

```
1. closed
        如果IO流（文件对象）已经关闭返回True
2. mode
        返回文件对象的访问模式
3. name
        返回文件的名字+扩展名
```

- File的方法

```
1. close()
        关闭文件----非常重要
2. read([count])
        读取文件中内容
        count：多少个字符
        不设置count：读取所有
3. readlines()
        读取每行的所有内容，打包成列表
4. readline ( )
        读取一行数据
        追加读取，读过的不能再次读取
5. tell()
        返回当前文件的指针位置
        一个中文占用两个字节   \n也是占用两个
6. seek(count,[from])
        count:移动多个字节
```

```
        from：0从起始位置  1从当前位置开始  2从文件末尾开始
7. write(str)
        讲字符串写入到文件中
8. writelines(seq)
        向文件中写入一个字符串序列(序列里面都是字符串格式)
9. flush()
        刷新(清空)文件对象内部的缓冲区，直接把缓冲区的数据立刻写入到文件中
```

文件系统

- os模块

```
os:win  mac Linux  Unix Android

.py---PVM---OS---硬盘
通过Python---OS：
    1. 获取平台信息
    2. 对目录的操作
    3. 判断
```

```
import os
1. os.getcwd()
        返回当前文件的完成路径---返回当前的工作目录
        C:\Users\Administrator\Desktop\新建文件夹
2. os.chdir()
        改变工作目录
3. os.mkdir()
        创建单层目录
4. os.makedirs(r'a\b')
        创建多层目录，以存在则报错
5. os.listdir(path)
        展示制定路径下的所有文件
6. os.remove()
        按照路径删除指定文件
7. os.rmdir
        删除单层目录指定路径，如果目录不为空，不能删除
8. os.removedirs()
        删除多层  ，如果有文件删除不了
        os.removedirs(r'C:\Users\Administrator\Desktop\b\a')  #删除b和a目录

9. os.rename(path1,path2)
        文件重命名:把path1的名字修改path2的名字
10. os.system('cmd')
        调用黑窗口
11. os.curdir
        表示当前目录  '.'
12. os.pardir
        表示上一级目录  '..'
13. os.linesep
        返回当前操作系统的换行符win：  \r\n  Linux/Unix:\n
14. os.sep
```

```
        返回当前操作系统的路径分割符
        win：\\   Linux/Unix:/
15. os.name
        返回当前操作系统的名字
        win：nt，Linux：posix
16. os.getenv('path')
        读取环境变量
```

- os.path模块

```
import os.path as p
1. p.basename(path)
        返回路径中的最后一个目录或者是文件
2. p.dirname(path)
        返回当前目录的上一级path路径
3. p.join(path1,path2...)
        将多个路径合成一个
4. p.split(path)
        分割路径（路径，文件名/目录名）
5. p.splitext(path)
        分割路径（路径+文件名，后缀）
6. p.getsize(path)
        返回文件的大小--字节，中文两个字节
7. p.getatime(path)
        返回文件最近一次访问时间---时间戳
8. p.getctime(path)
        返回文件的创建时间   时间戳
9. p.getmtime(path)
        返回文件最近修改的时间    时间戳
10. p.exists()
        判断路径或者文件是否存在
11. p.isabs(path)
        判断是否是绝对路径
12. p.isdir(path)
        判断是否是目录
13. p.isfile(path)
        判断是否是文件
14. p.islink(path)
        判断是否是链接
15. p.ismount()
        判断是否是挂载点----盘符C:\---挂载点----绝对路径开头
16. p.samefile(path,path2)
        判断两个路径下的文件是否是同一个文件
```

永久储存

```
import pickle
可以将各种类型的数据转成二进制文件放到文件中
```

- 写入

```
import pickle
##l=[1,2,3,4]
##f=open('hehe.txt','wb')
##
##
##pickle.dump(l,f)
##f.close()
```

- 读取

```
import pickle

f2=open('hehe.txt','rb')
print(pickle.load(f2))
f2.close()
```

- 打开图片

```
from PIL import Image

img=Image.open(r'C:\Users\Administrator\Desktop\02.jpg')
img.show()
```