文件操作

Python 使用 open() 方法，
可以根据指定文件名或标识符来打开文件。

```
# 语法格式------------------>>> open('你要打开的文件名字.后缀','打开文件的方式')
```

在电脑中要打开一个文件，需要使用open函数
函数在被使用的时候，需要传两个参数，第一个参数 ->要打开的文件名字.后缀 第二个参数 -> 打开文件的方式
open函数被调用后会返回一个文件对象。

```
# 实战：
# open函数会去寻找代码旁边的同级目录下的一个月光--胡彦斌.mp3 ，并打开
f = open("月光--胡彦斌.mp3",'rb')
# 读取f文件对象中的数据，并将读取结果返回保存到变量 res中
res = f.read()      # read() 是读取方法，它的作用是读取文件对象中的数据
#关闭文件
f.close()

# 用代码创建一个新文件，并将歌曲信息保存到这个新文件中，实现歌曲复制。

# w模式有一个帮助我们自动创建文件的功能。[w会去检测电脑中是否存在该文件，如果不存在则帮我创建后打开，如果存在则直接打开]
f = open('复制的歌曲.mp3','wb')
f.write(res)     #  .write(写入的数据)  往文件中写入东西
f.close()        # 歌曲写入好了，也不需要对文件进行操作了，直接关闭文件


"""
w 是写入普通的数据比如字符数据
wb 是写入二进制数据【图片、视频、音频】
b 用来处理媒体数据的
"""
"""
新需求：实现一个歌曲串烧的功能
整理：
目前我们已经读取了一首歌，并且将这个歌曲写入了一个 复制的歌曲 。
思考：
可以继续读取一首新的歌，然后继续往这个文件中写入数据.
"""
# 继续读取一首新的歌
f = open('大雨还在下.mp3','rb')
xinGe = f.read()    # 读取歌曲数据保存到变量 xinGe 中
print(xinGe)
f.close()

# 继续往  复制的歌曲.mp3  这个文件中写入数据 。 注意：追加写入数据之前，先要检查一下这个文件有没有问题。
```

```
# 同学回答的实现方式1、老师我们要用 wb 的方式继续往旧文件中写入数据  【再测试第一种方式】  -- 不通过
# 同学回答的实现方式2、我们要用 ab 的方式往旧文件中写入数据       【先测试第二种方式】 -- 通过

f = open('复制的歌曲二.mp3','ab')          #我们要用 ab 的方式往旧文件中写入数据
f.write(xinGe)
f.close()

f = open('复制的歌曲.mp3','wb')          # 用 wb 的方式继续往旧文件中写入数据
f.write(xinGe)
f.close()


f = open('哈哈哈哈哈.mp3','rb')
xinGe = f.read()   # 读取歌曲数据保存到变量 xinGe 中
print(xinGe)
f.close()

# rb  wb  ab  这个几个参数都是在打开文件的时候告诉电脑我需要的操作权限
# 有 b 是 binary二进制单词 的简称， 有这个 b 字母的 都是对媒体文件进程操作
```

```
#  普通数据的文件读写

# w  不仅可以帮我们开启写入数据的权限，还能帮我们创建一个文件
# w 还有一个功能，覆盖原文件
f = open('w我的文档.txt','w')
#写文件操作代码
f.write("哈哈哈哈哈\n")
f.write("我也觉得")
f.close()
# w+   是在 w 的基础上增加了文件读取的功能
f = open('ww我的文件.txt','w+')
#写文件操作代码
f.write("床前明月光。")

"为什么读取的数据是空的？ 这是因为我们写完数据之后，光标停在了文件末尾。后面就没有数据了，所以继续去执行读取的代码会读了个寂寞"
# 修改光标位置
f.seek(0)   # 将光标移到文件开头
print("当前光标位置",f.tell())    # 查看光标位置
x = f.read()
print("读取的数据是：",x)
f.close()

# a   追加、 创建一新文件当w使用
f = open('ww我的文件.txt','a')
f.write("\n这是我追加的数据。")
f.close()
f = open('a我的文件.txt','a')
f.write("这个是我用a追加的方式创建的文件")
f.close()
# a+    在a的功能基础上增加了文件读的功能
f = open('a我的文件.txt','a+')
```

```
f.write("这是我用a+追加的数据")
print(f.tell())      # 查看光标
f.seek(0)     #设置光标
print(f.read())    # 读取数据
f.close()

# r 读取数据
f = open('w我的文档.txt','r')
f.seek(0)
# print(f.read())
print(f.readlines())
f.close()     # 可能会常常忘记关闭
```

文件读写更加完美的写法【以后都用这种方式】

```
# 更加完美的方式去使用 open函数进行文件读写
# 好处：不再需要自己手动去关闭文件，自动关闭并清理数据
# 语法 ：  with open()as 小名：

# 实例：  将字符串列表['哈哈哈哈哈\n', '我也觉得']写入名为 哈哈哈哈的文件中
with open('哈哈哈哈.txt','w')as f:
    f.writelines(['哈哈哈哈哈\n', '我也觉得\n','这是我新写的'])
```

拓展：

```
# 1、导入 os
import os
x = os.getcwd()      # 获取当前目录（获取当前我这个代码文件所在的目录）
print(x)

x = os.listdir('D:\\pythonWork\\PyVIP22\\12、文件操作')       # 获取指定目录下的所有文件以及文件夹
print(x)

os.mkdir('D:\\pythonWork\\PyVIP22\\12、文件操作\\今天天气真好')       # 新建一个文件夹
```

课后作业：

1、文件读写操作案例敲一遍
2、使用os库的方法，要求至少使用4个方法（四个方法没有规定，自己选择）