# 一 读取

```
fileName = r'demo1.txt'
with open(fileName,encoding='utf-8') as f:
    print(f.readline(),end='')  # 读取一行内容
    print(f.readline())
    print(f.readline())
    print(f.readline())
```

```
fileName = r'demo1.txt'
with open(fileName,encoding='utf-8') as f:
    l = f.readlines()  # 一次性读完  保存到列表里面
    print(l[1:3])
```

# 二 写入

```
write
open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None,
closefd=True, opener=None)

r: 只读
w: 可写
a: 表示追加

fileName = r'demo2.txt'
with open(fileName,'w',encoding='utf-8') as f:
    f.write('hello world')   #  只能传递字符串


fileName = r'demo3.txt'
with open(fileName,'w',encoding='utf-8') as f:
    f.write('nihao , 世界\n')   #  只能传递字符串
    f.write('ni\n')   #  只能传递字符串
    f.write('nihao 世界\n')   #  只能传递字符串
    f.write('nih , 世界\n')   #  只能传递字符串


fileName = r'demo3.txt'
with open(fileName,'w',encoding='utf-8') as f:
    a = f.write('nih')   #  只能传递字符串
    print(a)

fileName = r'demo3.txt'
with open(fileName,'a',encoding='utf-8') as f:
    f.write('\nnihdasda')   #  只能传递字符串
```

# 三 ATM项目搭建

# MVC 开发模式 ==》 架构分明

MOBA 游戏

坦克 治疗(辅助) 输出 1+1+1 》 3

包的核心作用: 分类，更好的管理

m==》 model ==》 模块层 ==》 数据库，增删改查

v ==> view ==》 视图层 ==》 用户看到的东西

c ==》 controller ==》 控制层 ==》 用户的交互，控制行为

```
view(视图): 用于数据的显示，显示的媒介一般是HTML 和json
model(模型): 处理项目的数据逻辑，一般用于处理sql数据
controller(控制器): 负责交互
```

# 项目开发的流程

1.没有开发思路

2.BUG

3.后期有需求(增加功能，修改功能)

养成好习惯:

先做好项目的架构分析，架构设计(用什么语言，框架，数据，系统，底层设计)

做好内部功能的区分

写一个readme文件(说明书，项目如何使用，注意事项) ==》 帮助文档

1. 准备计划，分析项目逻辑，细节
2. 新建项目文件
3. 建立架构

# 项目环境的搭建

1. 设计目录结构，分层结构

```
1.建立api文件夹  api: 应用程序编程接口
    -user_interface.py 用户业务逻辑
    -bank_interface.py 银行业务逻辑
2.建立config文件夹   配置
    -setting.py  设置
3.建立core文件夹   核心文件
    -src.py   功能文件
4.建立db文件夹  database  存放，操作数据
    -db_handle.py  数据处理层，增删改查
    -user_data  用户的数据文件文件夹
5.建立lib文件夹  自定义模块
    -common.py   常用的意思

run.py  ===>  项目的启动文件
```