**Selenium+Unittest自动化测试框架：**

1. **环境搭建：**

①windows 64位

②python 3.6版本 环境变量配置

③pip install selenium 安装selenium库

**一、POM设计思想：**

1.底层封装，如定位方法，显式等待、断言；（文件：common）

2.页面元素封装，每个操作写一个方法； （文件：pages）

3.设计测试用例，调用页面封装方法； （文件：cases）

4.执行所有以test\*.py文件的测试用例； （文件：run\_cases）

5.生成自动化测试报告； （文件：report）

1. **总结：**
2. python3 + pycharm +Chrome +Chrome驱动；
3. Python环境变量配置；
4. 安装pip、selenium、ddt等（详细看需要）
5. 封装公共模块如：显式等待元素定位、鼠标悬停、断言等模块方法；
6. 主要思路：

①封装基础模块→②页面元素封装→③用例设计、ddt参数化→④run执行所有用例生成测试报告→⑤错误时/可发送邮件

1. 注意点：

①用例都是以test开头，执行报错，可能是程序目录有问题；

②页面封装每个操作都需要调试好，单元测试；

③最好每个页面的每一个操作封装一个方法；

④ddt参数化需要灵活使用，不然报错难以控制；

⑤命名需要规范，类大写字母开头，common、cases、pages等命名；

⑥必须的模块，如生成报告的HTMLTestRunner、dtt参数化excel数据的xlrd封装；

⑦富有文本定位、日历控件等等难元素定位操作用JS语法，操作windows外面安装AutoIt;

**三、Git的使用：**

查询状态：git status

①本地目录建仓：git init →②添加远程仓库：git add \* →③提交到缓存：git commit -m “xxx”④推送：git remote add origin <https://xxxx> →⑤提交到maseter分支：git push -u origin master

**下载**：git clone <https://xxxx>

**更新：**①git status ②git add \* ③git commit -m “xxx” ④git pull ⑤git push origin master

**Git Bash：**  
①删除：.SSH文件下的known\_hosts(.SSH在C:\Users\Windows用户名目录下)   
②输入：ssh-keygen -t rsa -C "你的名字/你的邮箱"  
③用记事本打开目录下的id\_rsa.pub，将里面的内容复制git上的SSH秘钥