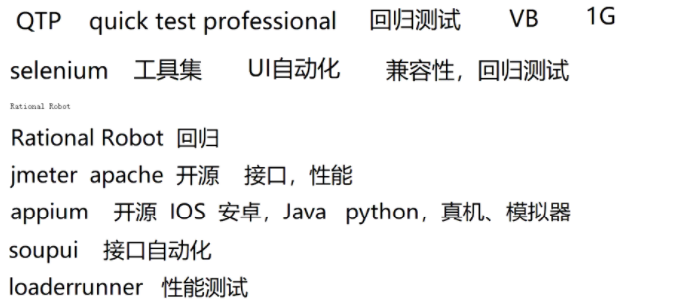
**自动化测试**：软件测试的自动化，在预设状态下运行应用程序或者系统，预设条件包括正常和异常，最后评估运行结果。将人为驱动的测试行为转化为机器执行的过程

**自动化测试的演变：测试工具----数据驱动----关键字驱动----测试框架----测试平台**

**常见的自动化测试工具：**



**Selenium相比QTP的优点：**

免费

小巧，QTP需要1G多

**多语言**：C、java、ruby、python、或都是C# ，你都 可以通过selenium 完成自动化测试，而QTP 只支持VBS

**支持多平台**：windows、linux、MAC ，支持多浏览器：ie、ffff、safari、opera、chrome

支持分布式测试用例的执行，可以将测试用例分布到不同的测试机器执行，相当于分发机的功能

**自动化测试工具**：屏蔽底层代码实现，以图形化界面的形式展示给用户

**自动化测试框架：**为了解决某类问题而开发的产品，用户只要使用框架内部的类和方法就能解决问题。

**自动化测试库：**提供类和方法来开发脚本

**UI自动化：**回归测试、兼容性测试（功能要稳定、界面元素稳定、基本不再有变化）

用例维护量大

页面相关性强，必须后期介入

UI测试适合与界面变动较小的项目

**前后端功能都开发完成的时候做**

**接口自动化：（后台开发完成时）**

通过抓包查看接口参数 Charles、fiddler

测试目的:输入输出 接口的正确性测试 大量的创建数据

可在产品前期介入

用例维护量小

页面相关性小

适合接口变动较小，界面变动频繁的项目

**自动化测试的优势：**

减少重复测试的时间，实现快速回归测试

创建优良可靠的测试过程，减少人为错误

可以执行一些手工测试困难或不可能进行的测试

测试具有一致性和重复性

测试脚本的重用性

**自动化测试的使用对象：**

**前提条件：需求变动不频繁、项目周期足够长、自动化测试脚本可重复使用**

1.需求变动频繁的项目，自动化脚本不能重复使用，维护成本太大，性价比低

2.项目周期短，自动化脚本编制完成后使用次数不多，性价比低

3.交互型较强的项目，需要人工干预的项目，自动化无法实施

**适合做自动化的项目：**

产品性项目：新版本是在旧版本的基础上进行改进，功能变不大的项目，但项目的新老功能都必须重复的进行回归测试。回归测试是自动化测试的强项。

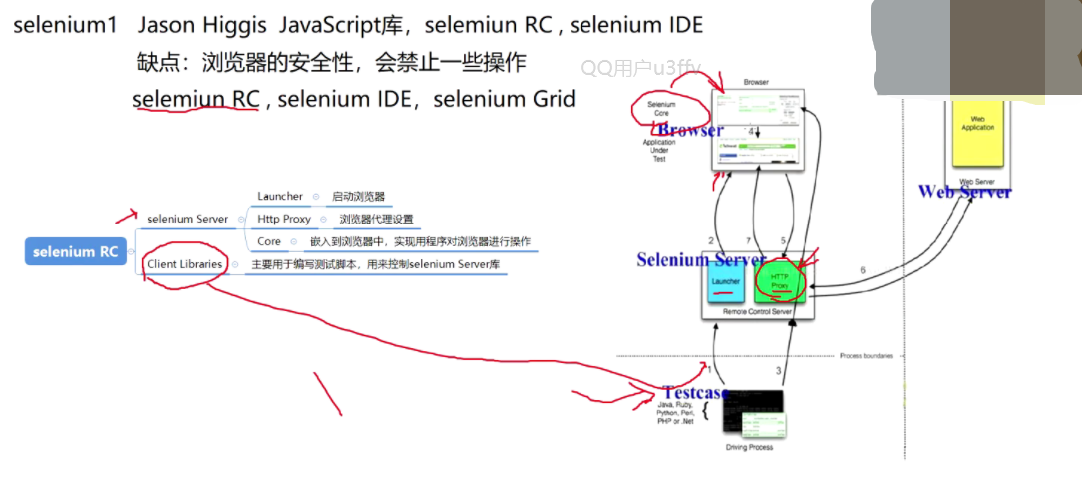
机械并频繁的测试：每次需要输入相同。大量的一些数据。并且在一个醒目中运行周期比较长

**自动化测试失败的因素：**

1.期望值过高，不可能达到100%

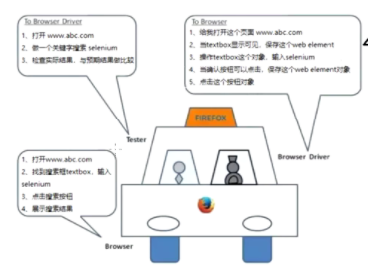
2.对收益和成本认识不清（成本＝实现成本＋运行维护成本）自动化测试的收益是由测试脚本的重复运行次数，或自动测试脚本的利用率决定的。

**Selenium版本发展：（问题：对网站存在安全证书问题）**



Selenium2： Google webdrive（原生浏览器驱动）





Selenium3:是对前面两个版本的升级 Edge Safari也可以使用驱动

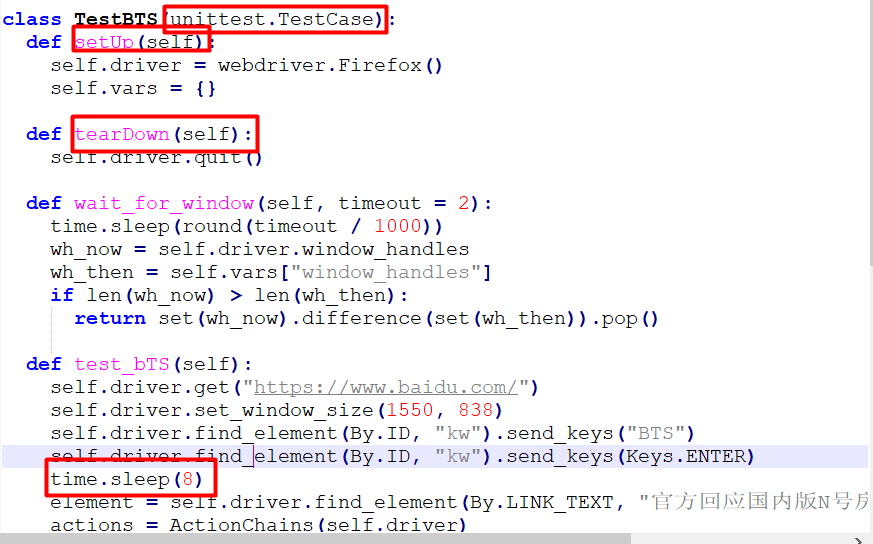
**录制脚本：**

首先选择一个进行搜索的网址

然后建立一个项目—输入URL—录制—录制完成进行导出（可以本地运行，也可以在Selenium直接运行）

**本地运行改造代码：**

import unittest//不使用pytest，而是unittest
添加time.sleep是为了解决页面定位不到关键字而出现报错



本地执行：

cmd:

cd 切盘符
python 文件名

PyCharm：

新建工程——文件夹——python file——编写脚本——右键点击RUN执行代码

crtl+/进行屏蔽

用过右击——检查搜索框id,name,tag\_name,class\_name等

#通过id 方式定位
browser.find\_element\_by\_id("kw").send\_keys("selenium")
#通过name 方式定位
browser.find\_element\_by\_name("wd").send\_keys("selenium")
#通过tag name 方式定位
browser.find\_element\_by\_tag\_name("input").send\_keys("selenium") 不能成功，因为input太多了不唯一。
#通过class name 方式定位
browser.find\_element\_by\_class\_name("s\_ipt").send\_keys("selenium")
#通过CSS 方式定位
browser.find\_element\_by\_css\_selector("#kw").send\_keys("selenium")
#通过xphan 方式定位比特科技
browser.find\_element\_by\_xpath("//\*[@id='kw']").send\_keys("selenium")
browser.find\_element\_by\_id("su").click()