郑学晶

13466334193

开始学习selenium  
1、下载资料SeleniumFiles1   
2、下载ECShop安装包（包含安装文件、视频、文档），安装配置ECShop，上课当天早晨再次确认ECShop能正常访问使用。     
3、电脑上安装准备好Python和Pycharm。  
SeleniumFiles1和ECShop的下载链接统一为：IMG_256https://pan.baidu.com/s/1\_or2LRVovZcyfUx53-NUyg   
提取码：1eba

Day01

1. Selenium介绍
2. Selenium是什么？

Selenium是一款功能自动化测试工具（也有人把它叫做功能自动化测试框架），只适用于PC端（电脑上）的Web应用程序的测试，不适用于C/S结构的被测系统的测试。

说明：有人把功能自动化，叫做UI自动化、Web自动化。

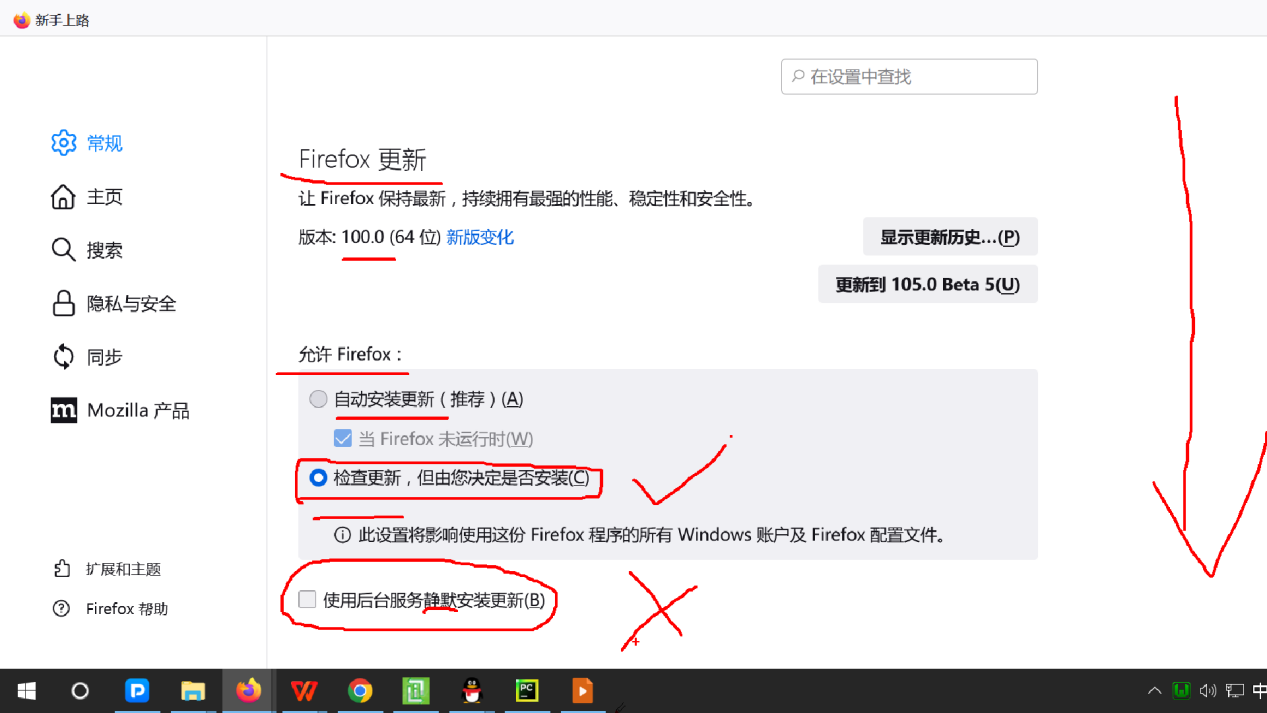
1. Selenium主要用途：重复测试
2. 回归测试
3. 兼容性测试
4. 大数据量测试：一般十几条或几十条数据。
5. Selenium优点：
6. 开源免费
7. 支持多种操作系统：Windows、Mac、Linux等。
8. 支持多种浏览器：Firefox（火狐）、Chrome（谷歌）、IE、Edge、Safari等
9. 支持多种编程语言：Python、Java、C#、JS、Ruby等。
10. 课程里使用的Selenium版本：3.141.0
11. Selenium几个组件：
12. Selenium WebDriver：最重要！书写代码。
13. Selenium IDE：初学者使用，有界面，可以录制回放，做简单的自动化测试。---了解！
14. Selenium Grid：并行、分布式测试，提升测试效率。
15. 官网：https://www.selenium.dev/
16. Selenium IDE---了解！
17. Selenium IDE：Selenium集成开发环境。
18. 支持：Firefox、Chrome、Edge等。
19. 安装Firefox：

先关闭360等杀毒软件。

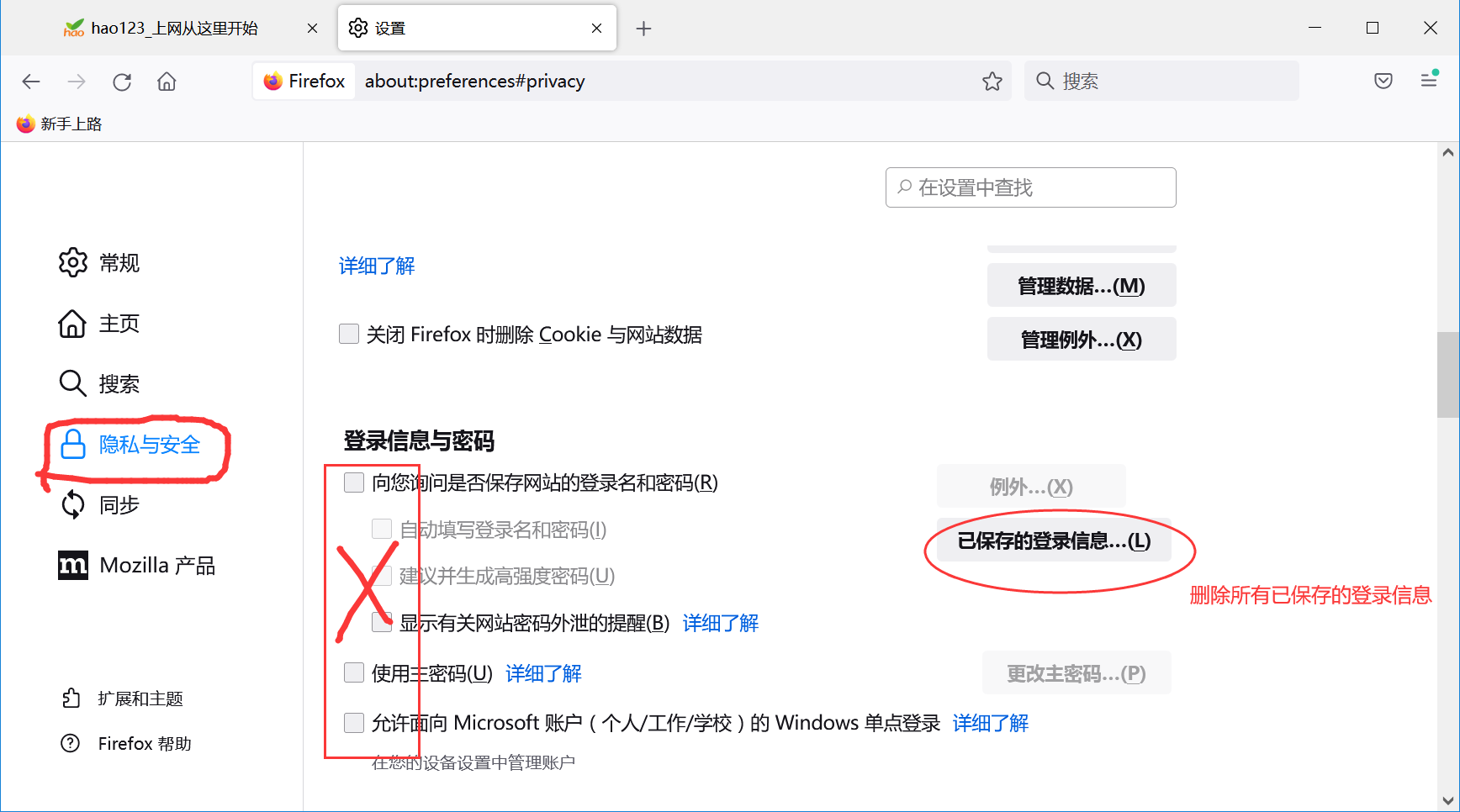
安装包：Firefox-latest.exe。

1. 配置Firefox：Firefox---右上角三条横线---设置

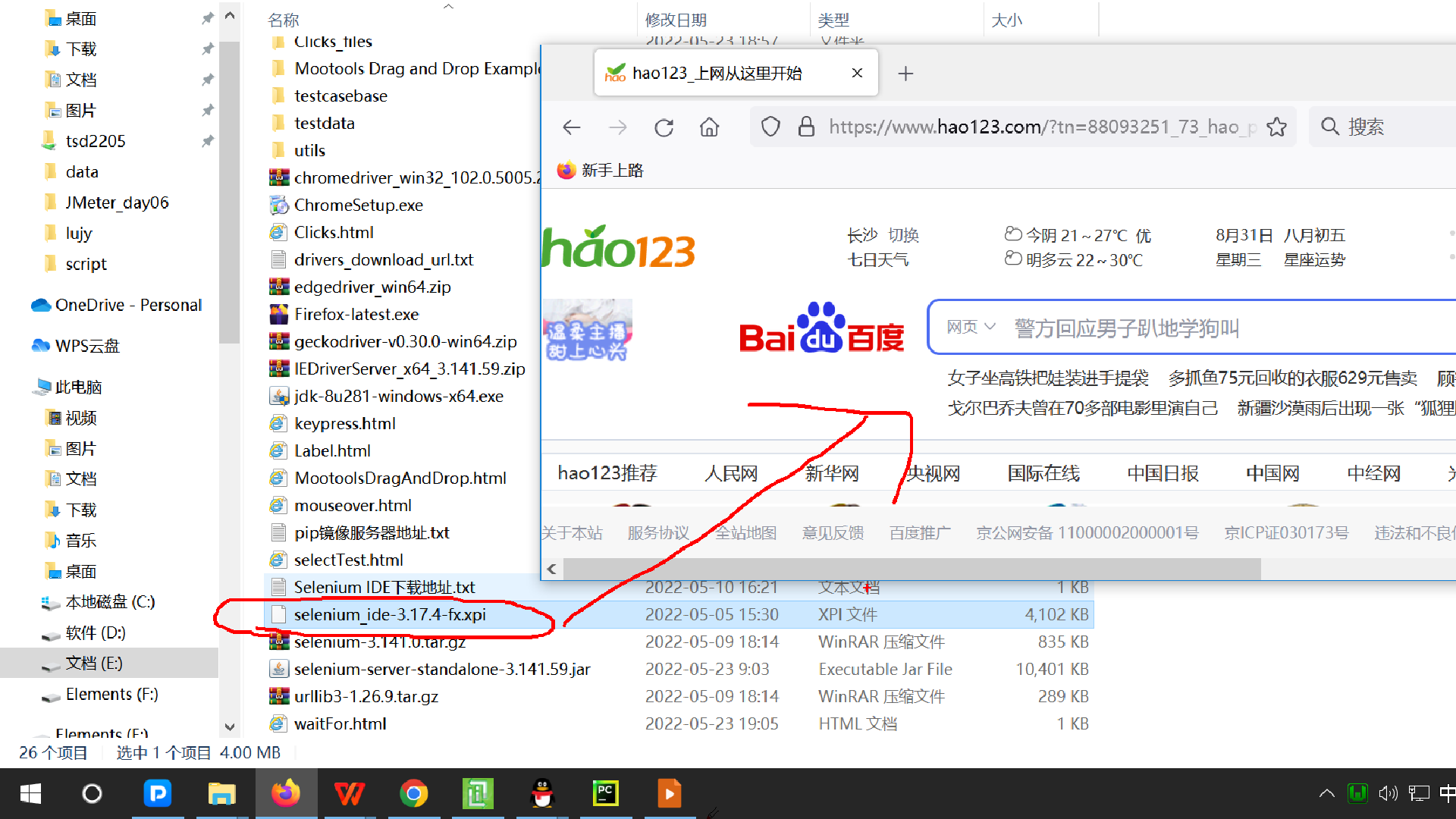
（1）取消更新



1. 取消保存登录信息和密码



1. 安装Selenium IDE
2. 方式一：拖拽安装包（selenium\_ide-3.17.4-fx.xpi）到Firefox窗口里，点击“添加”，点击“好的”。

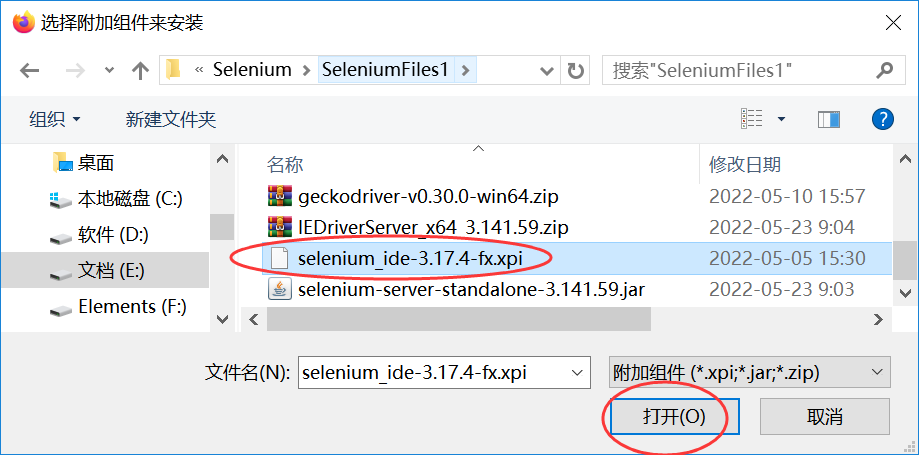




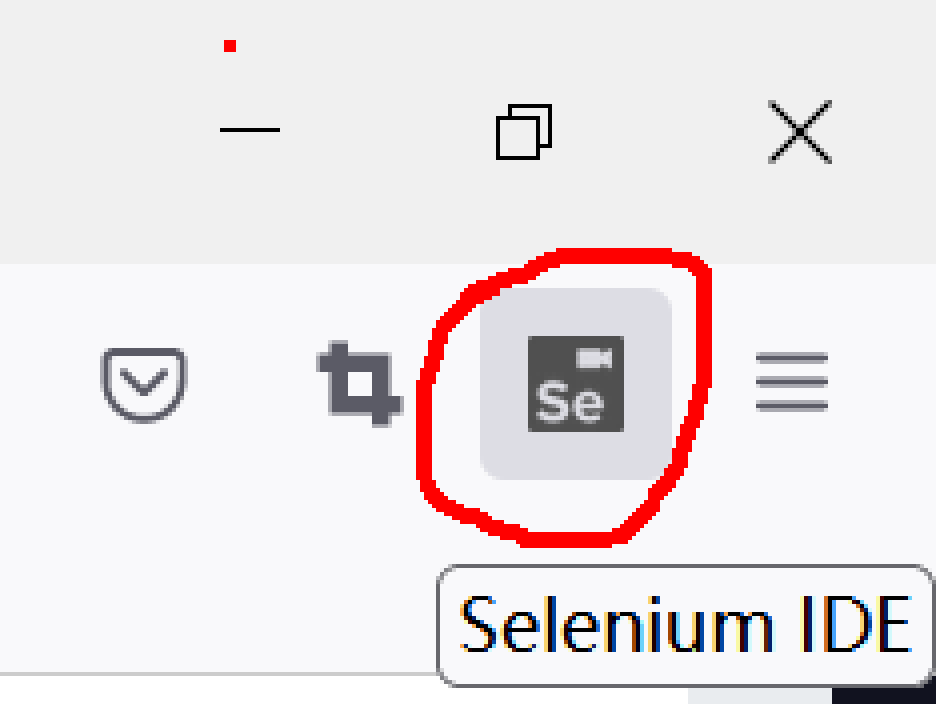


1. 方式二：如果无法拖拽，Firefox右上角---扩展和主题---齿轮图标---从文件安装附加组件---选择安装包---添加---好的。





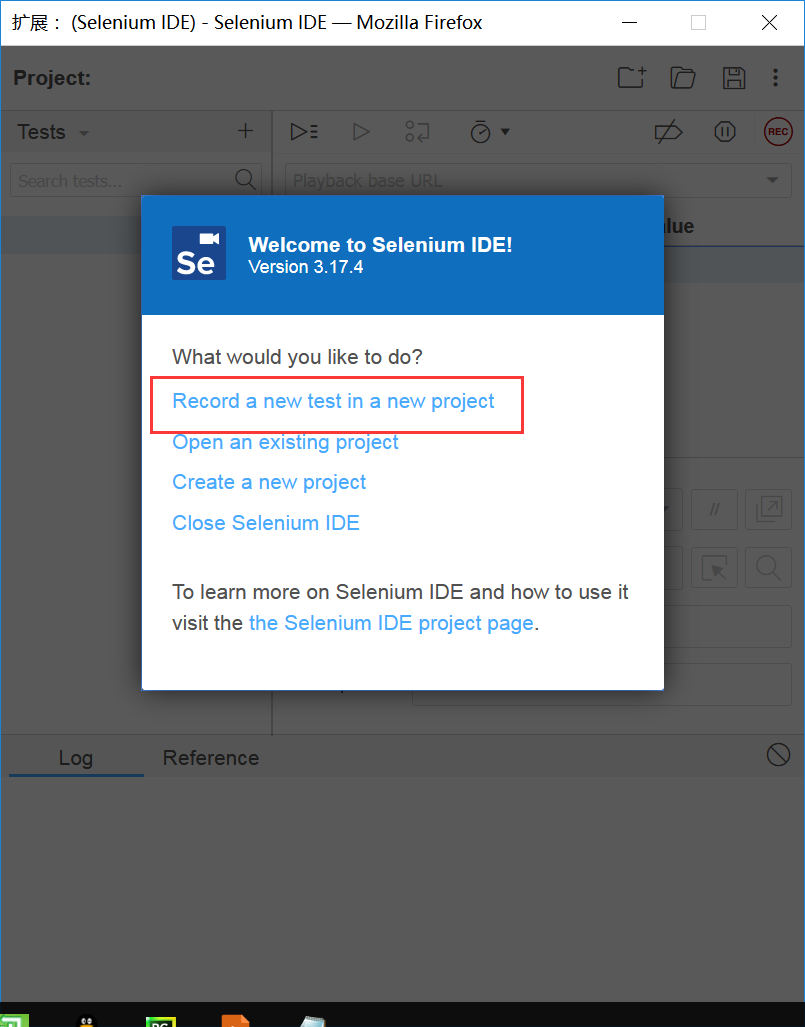
1. 启动Selenium IDE：



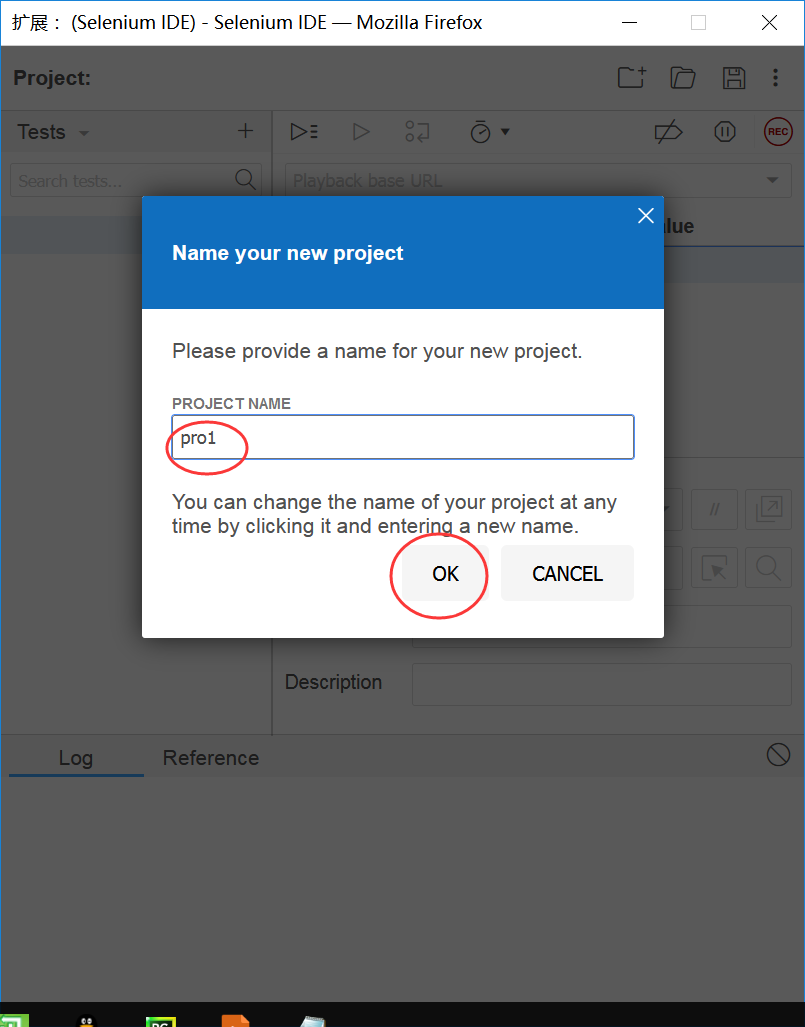
1. 使用Selenium IDE做简单的功能自动化测试：录制回放机制

（1）启动Firefox，启动Selenium IDE

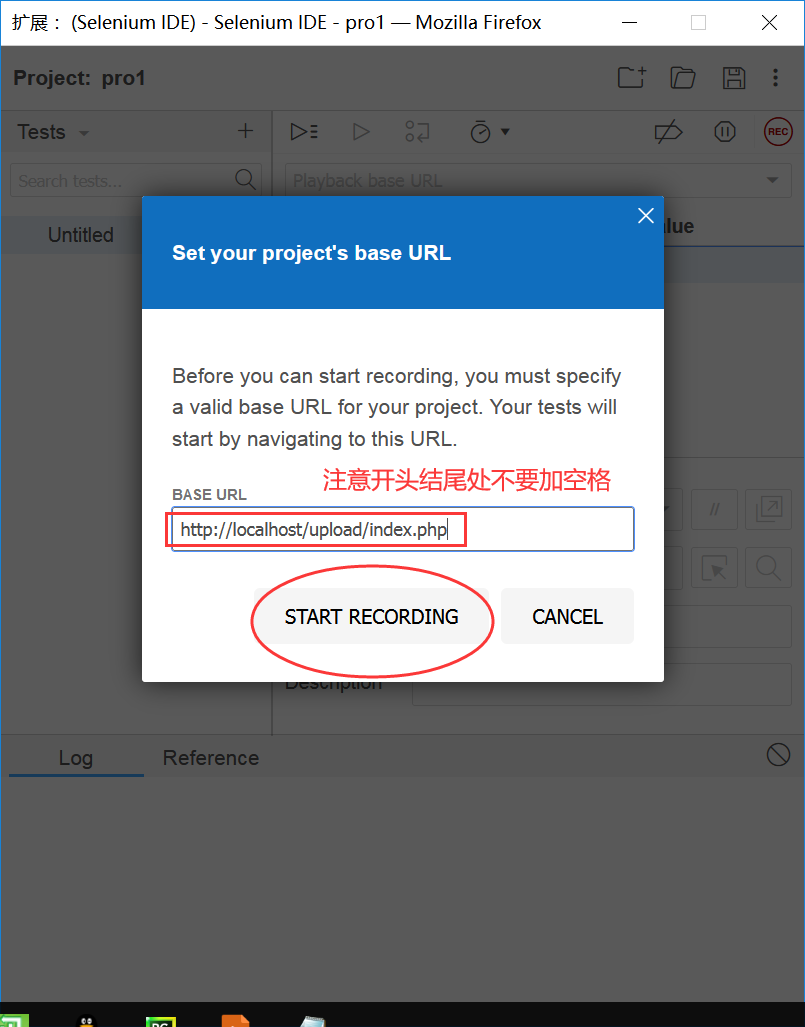
（2）Record a new test in a new project



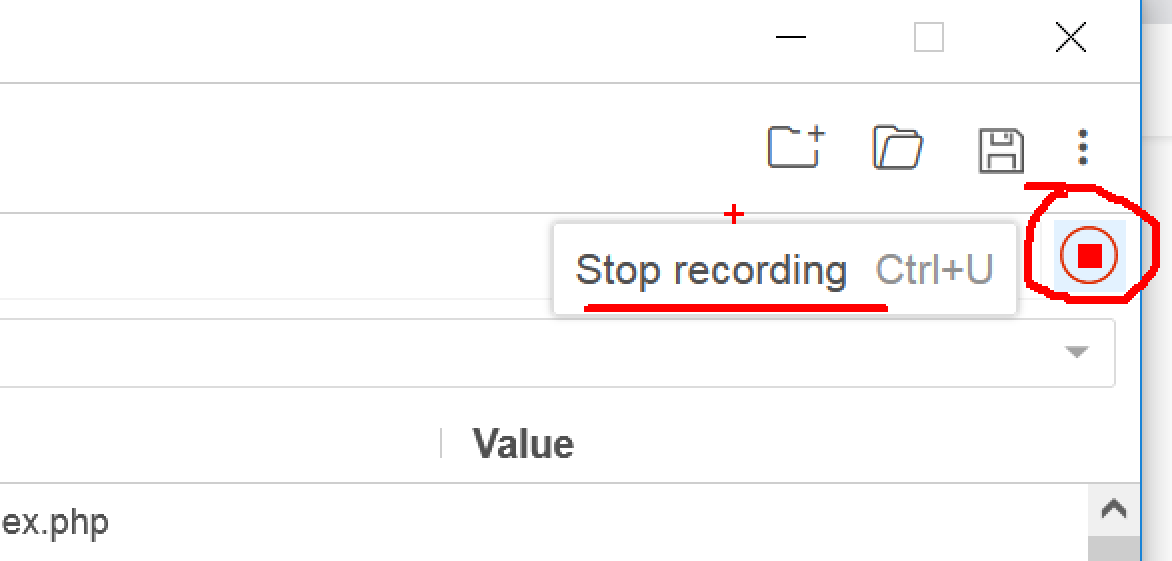
1. 输入PROJECT NAME项目名称（工程名称）:自定义名称（假设：pro1）,点击OK

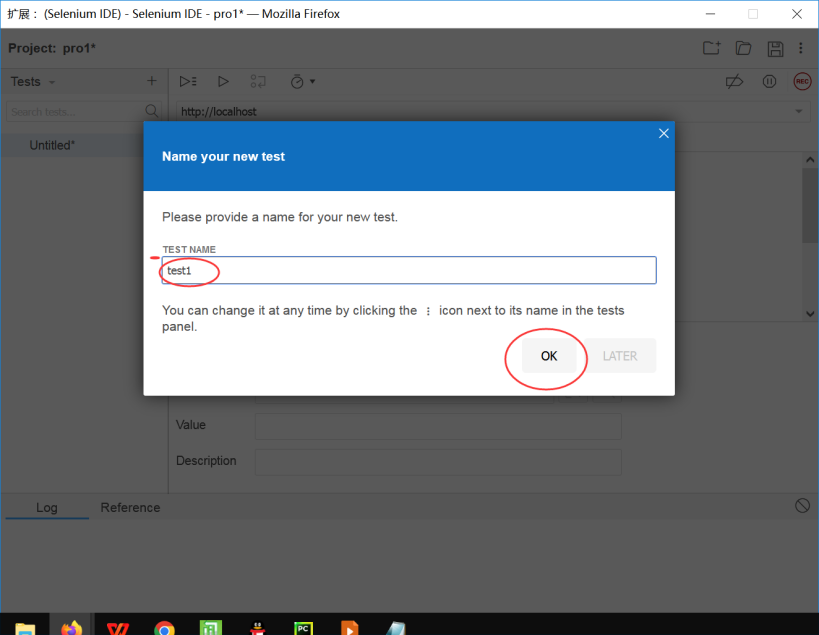


1. 输入BASE URL（被测系统的访问地址，假设是ECSHop前台首页，http://localhost/upload/index.php），点击START RECORDING(开始录制)

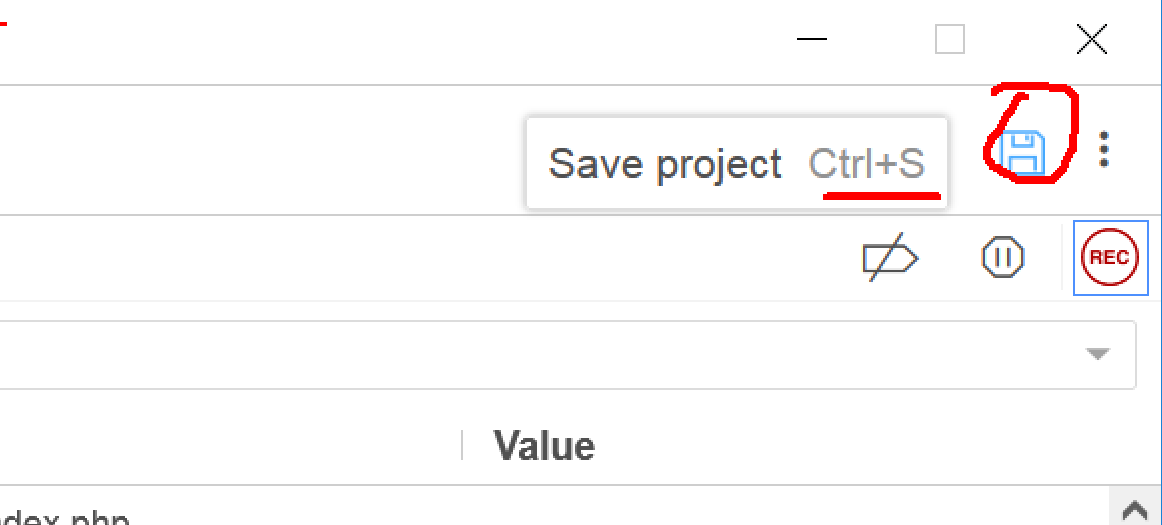


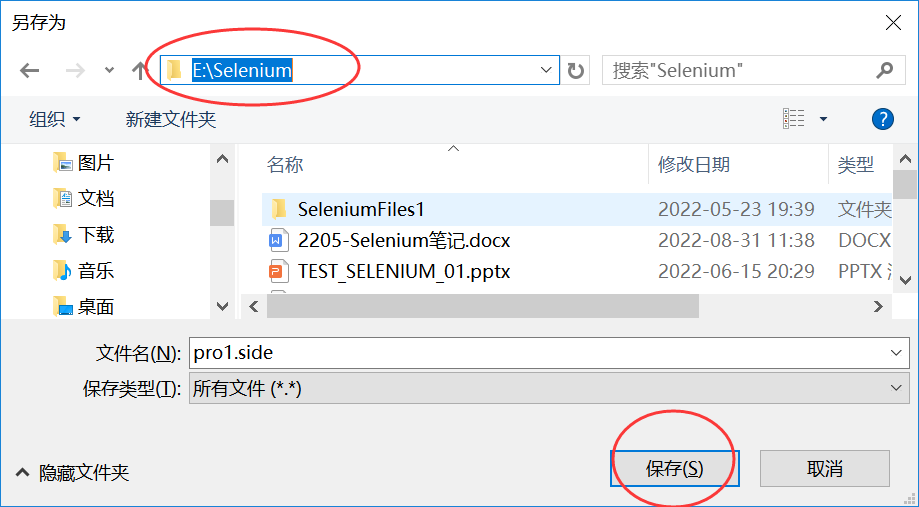
1. 在Selenium IDE所自动启动的浏览器窗口里，按照设计好的测试用例文档里的描述一步一步的进行操作，最后关闭浏览器。（假设：点击“登录”，输入用户名111，输入密码222，点击“立即登陆”）
2. 点击“停止录制”，在弹出框里输入TEST NAME(测试名称，自定义，假设test1)，点击OK



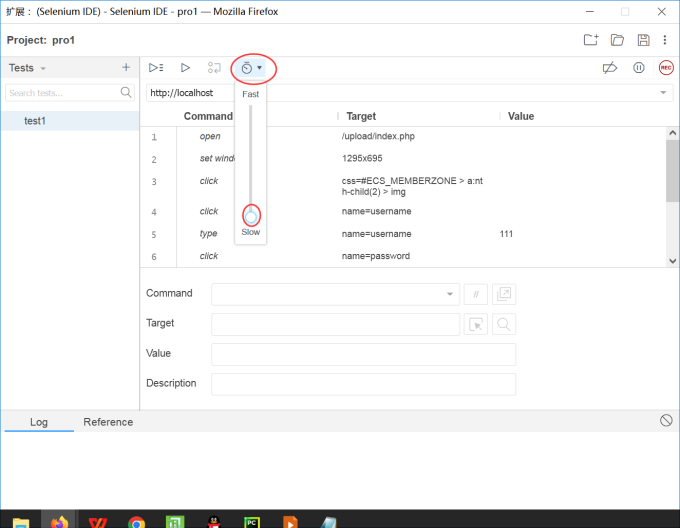


1. 保存：Ctrl+S，选择自定义文件夹路径，保存

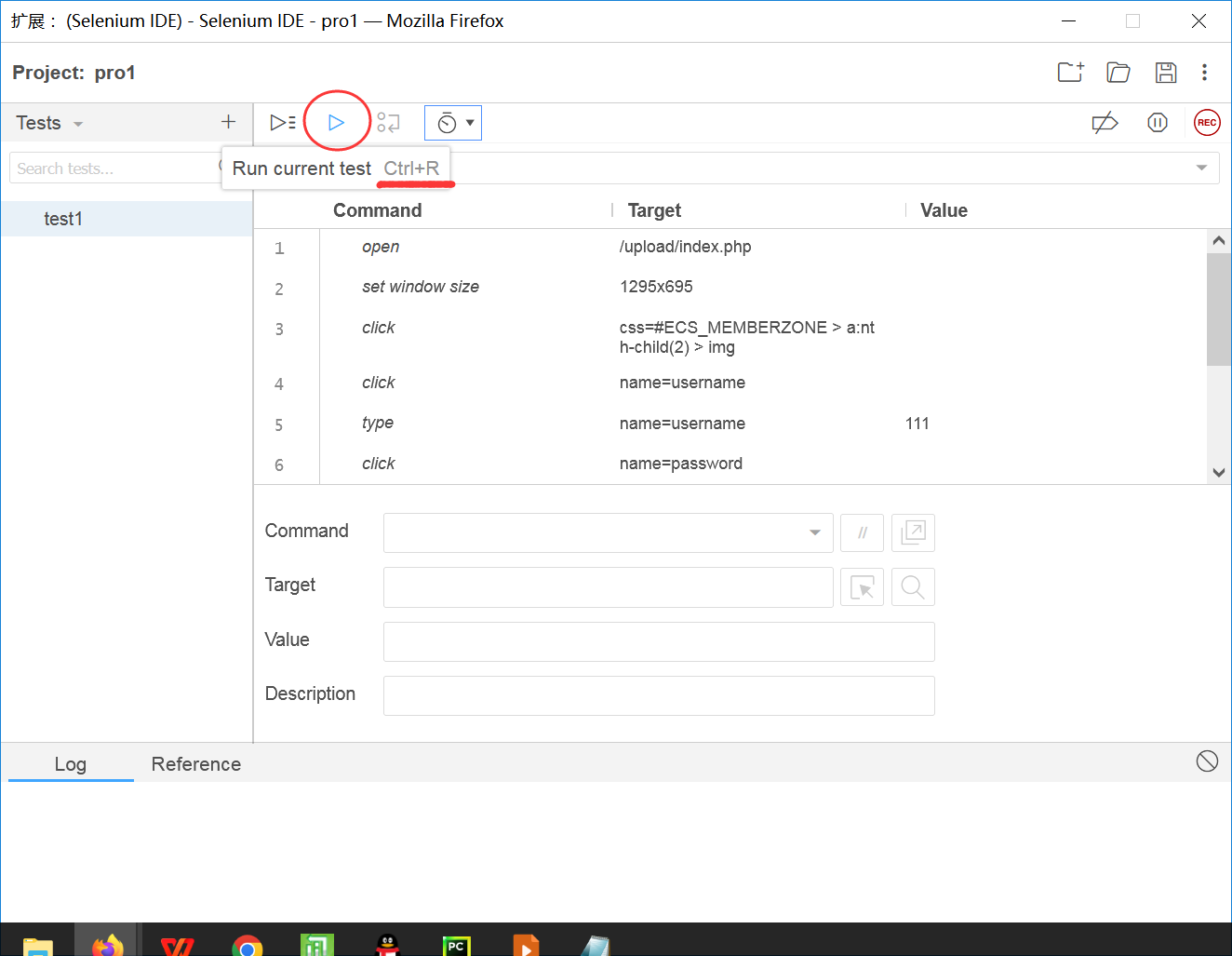




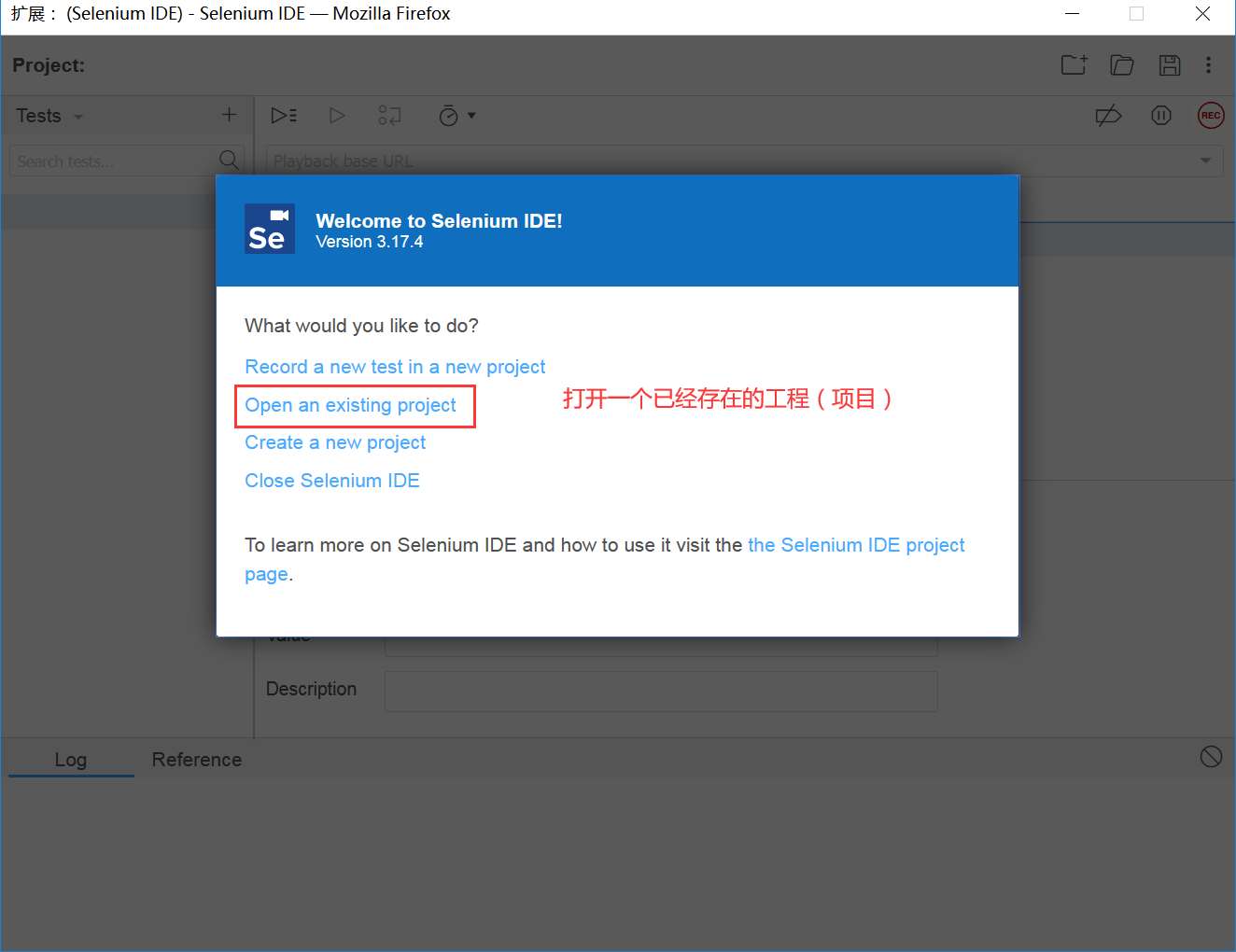
1. 调整速度：Slow慢速 ---每次打开程序运行前都需要调整。



1. 运行代码（回放）：点击第二个三角号



1. 第二天打开前一天所录制的代码：





注意：每次都需要调整运行速度。

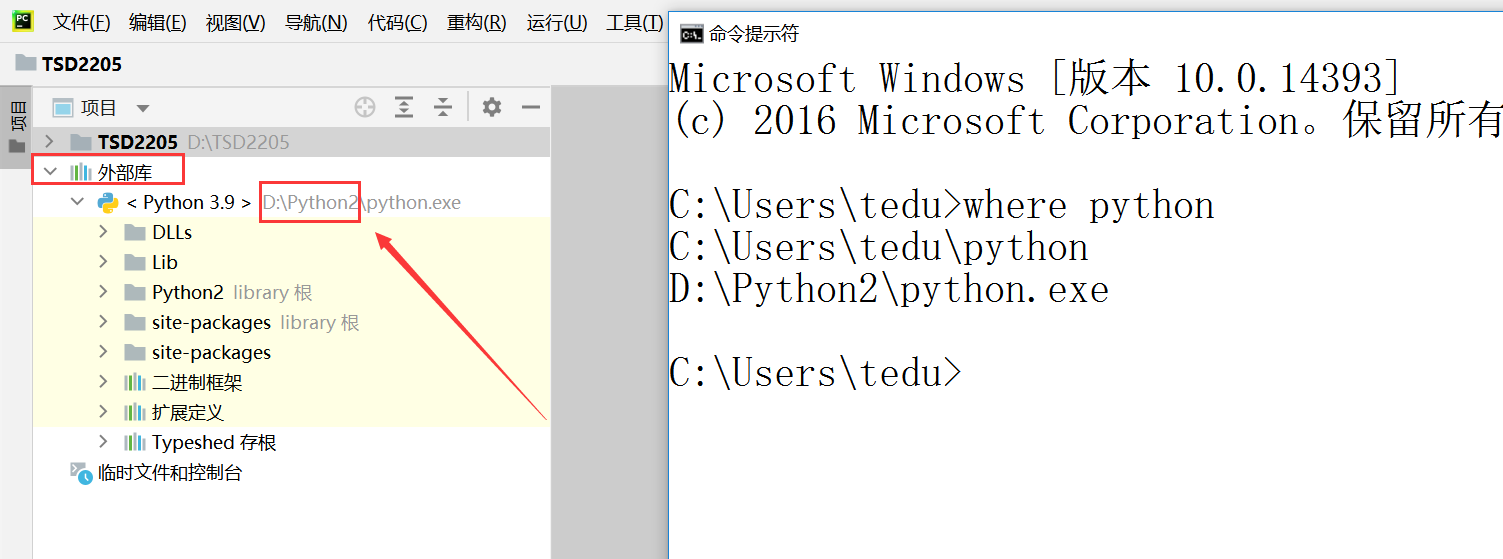
练习：用Selenium IDE录制ECShop前台页里，点击“高级搜索”，输入价格范围100到999，点击“立即搜索”的步骤。保存到pro2.side(其中包含test2)，回放调试代码。

1. Selenium WebDriver环境搭建
2. 安装Python---略！

C:\Users\tedu>where python

C:\Users\tedu\python

D:\Python2\python.exe

、

说明：Python 3.x的版本都可以（3.6、3.7、3.8、3.9）

C:\Users\tedu>python

Python 3.9.5 (tags/v3.9.5:0a7dcbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>>

1. 安装Pycharm---略！

2018、2020、2021、2022版都可以。

1. 安装selenium库：不是python自带的库，是第三方库，所以需要单独安装。
2. 在线安装：cmd里输入

pip install selenium==3.141.0 -i 网址

网址是下载这个库的服务器地址（镜像地址、镜像源）

<https://pypi.tuna.tsinghua.edu.cn/simple>

http://pypi.douban.com/simple/ 豆瓣

http://mirrors.aliyun.com/pypi/simple/ 阿里

http://pypi.hustunique.com/simple/ 华中理工大学

http://pypi.sdutlinux.org/simple/ 山东理工大学

http://pypi.mirrors.ustc.edu.cn/simple/ 中国科学技术大学

https://pypi.tuna.tsinghua.edu.cn/simple/ 清华大学

http://pypi.hustunique.com/ 华中科技大学

http://mirrors.cloud.tencent.com/pypi/simple 腾讯

https://repo.huaweicloud.com/repository/pypi/simple/ 华为

示例:

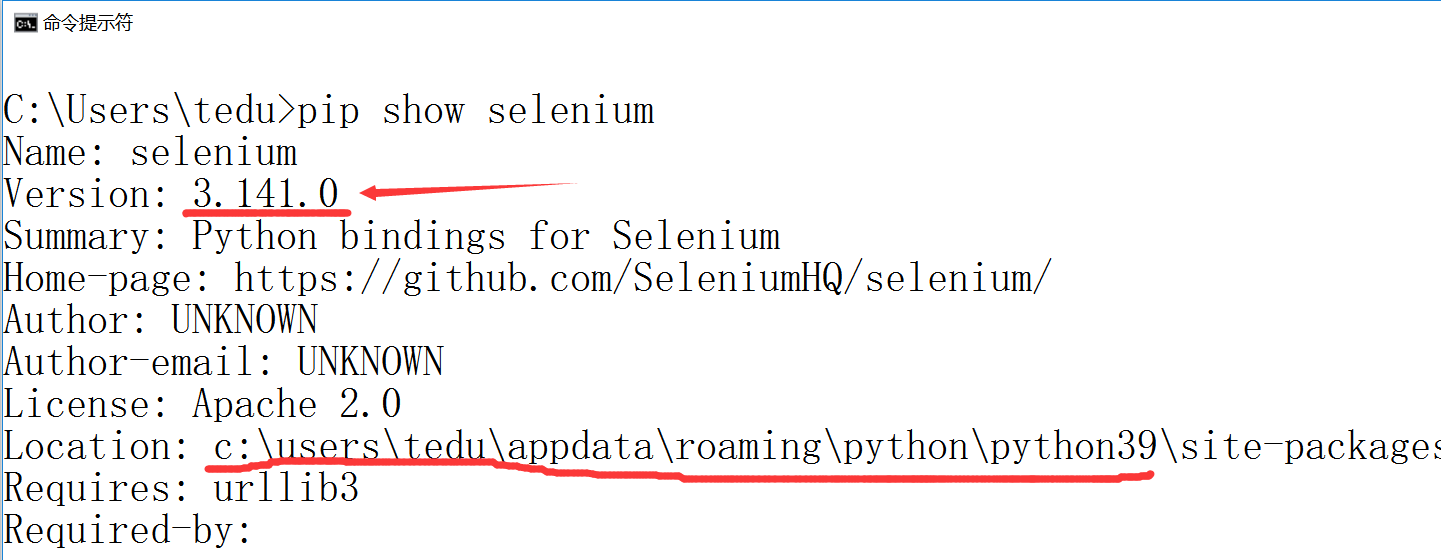
pip install selenium==3.141.0 -i https://pypi.tuna.tsinghua.edu.cn/simple

1. 说明：cmd里出现以下两个提示中任意一个都可以了。

Successfully installed selenium-3.141.0 ---本次安装成功！

Requirement already satisfied: selenium==3.141.0 --以前安装过！

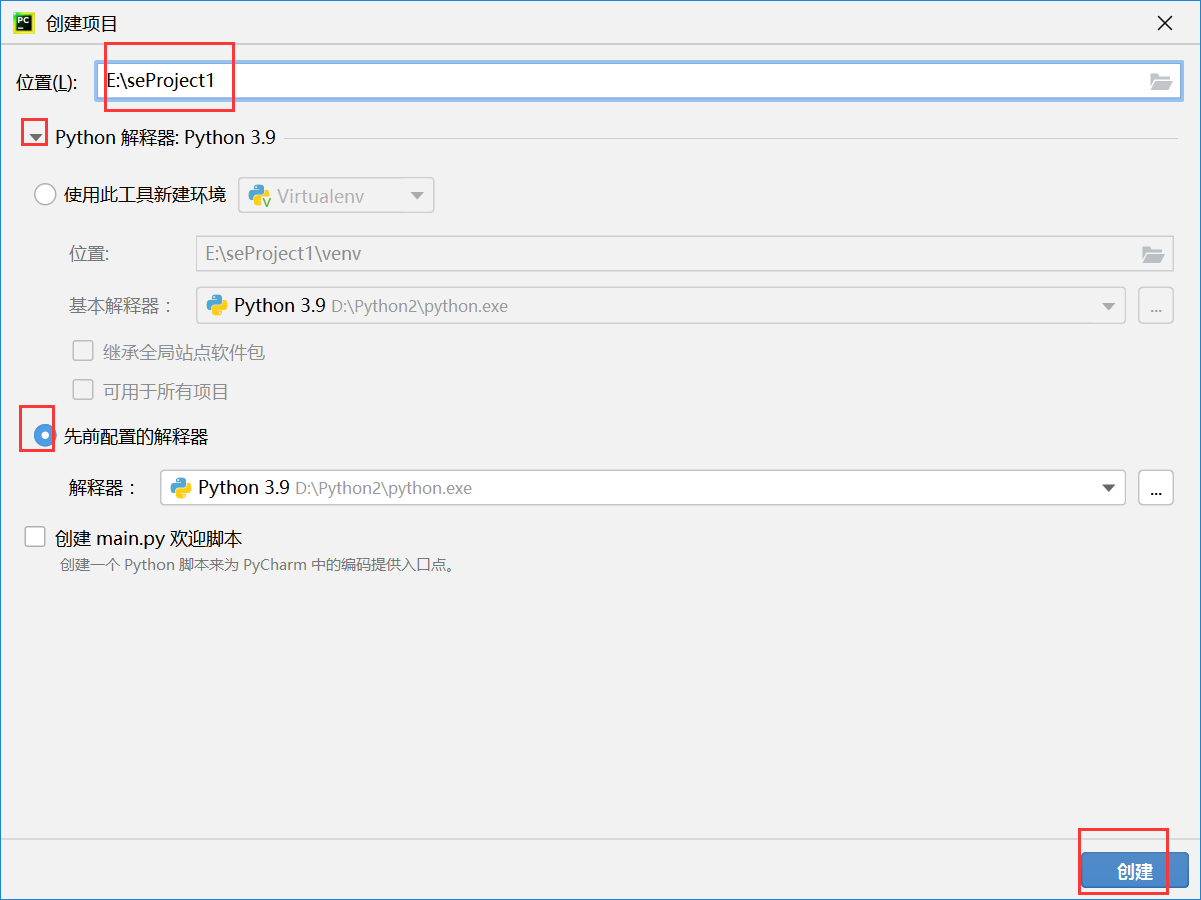
1. 检查：cmd里输入pip show selenium

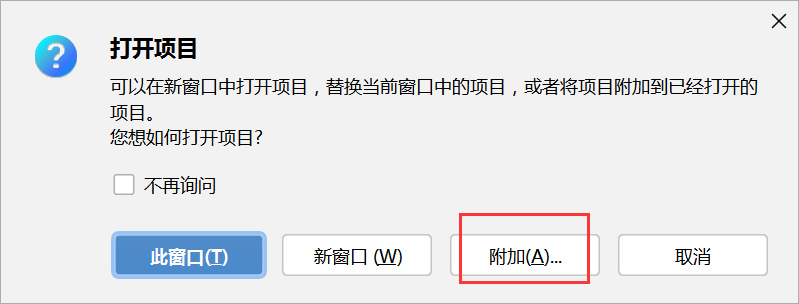


检查能从selenium库里导入模块进行使用，表示安装正常。

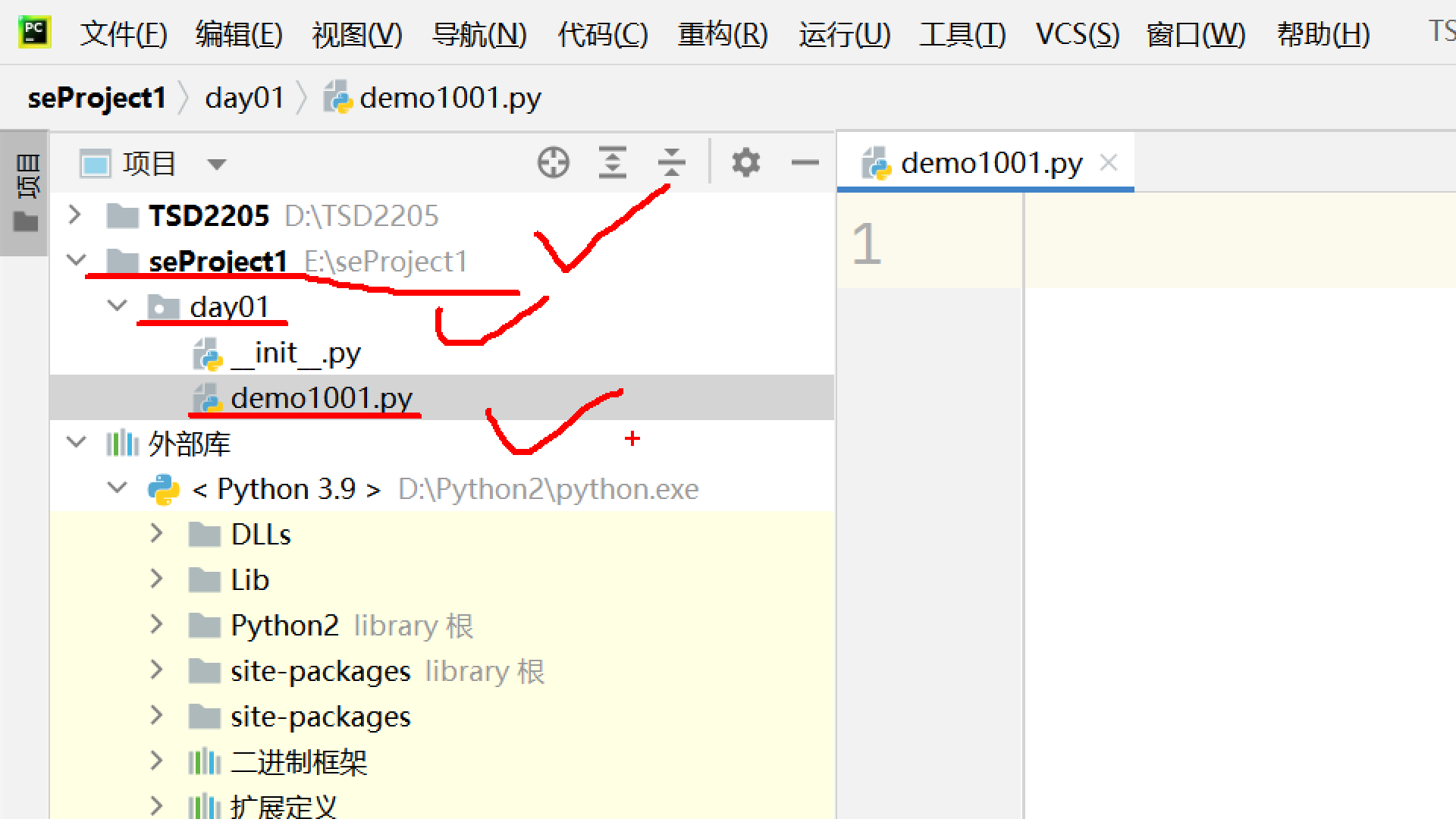


1. 在Pycharm里新建一个项目（工程）：
2. 启动Pycharm：菜单里“文件”下“新建项目”





1. 新建python软件包、新建python文件



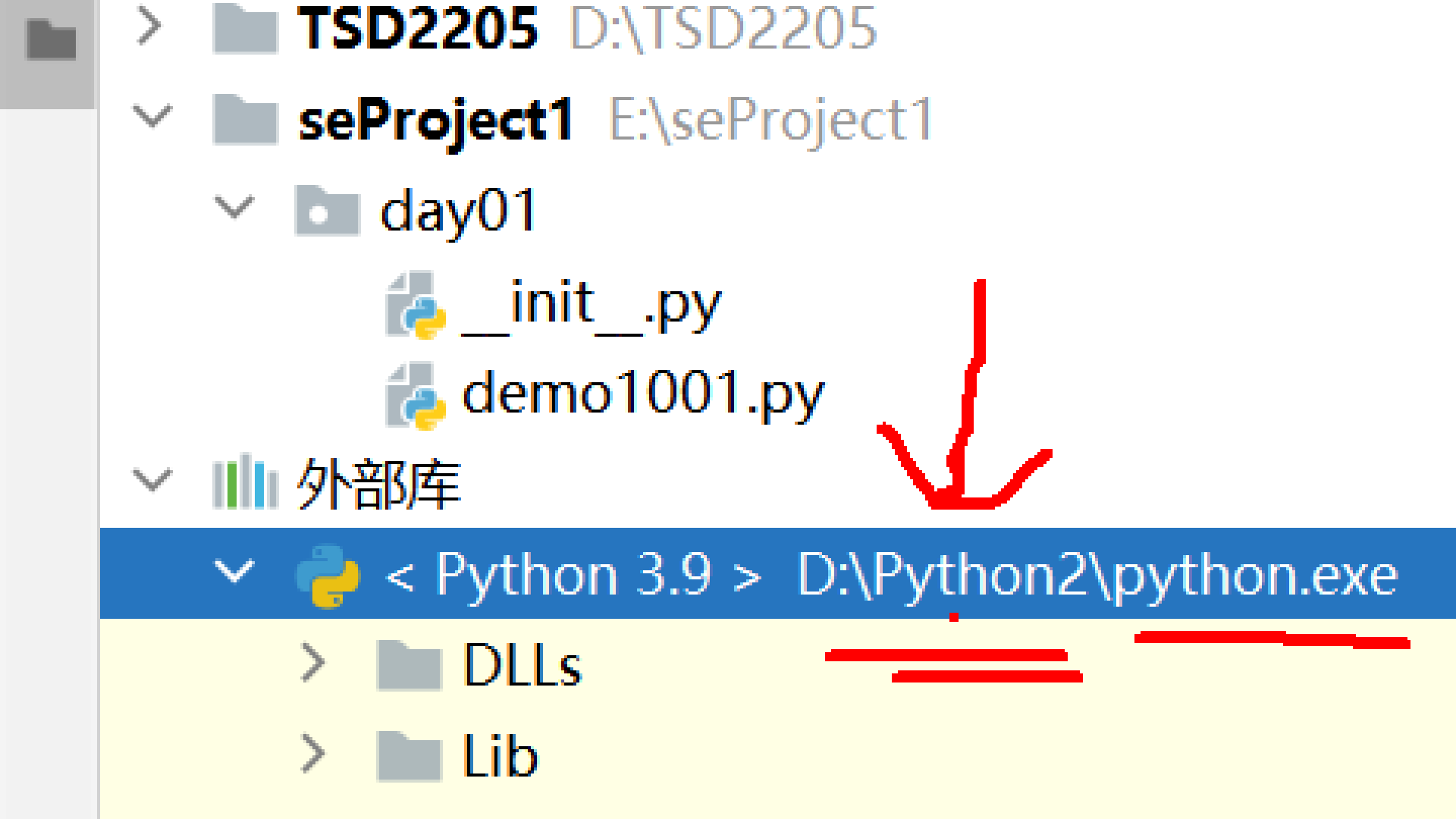
1. 书写一行代码来测试selenium库是否可用

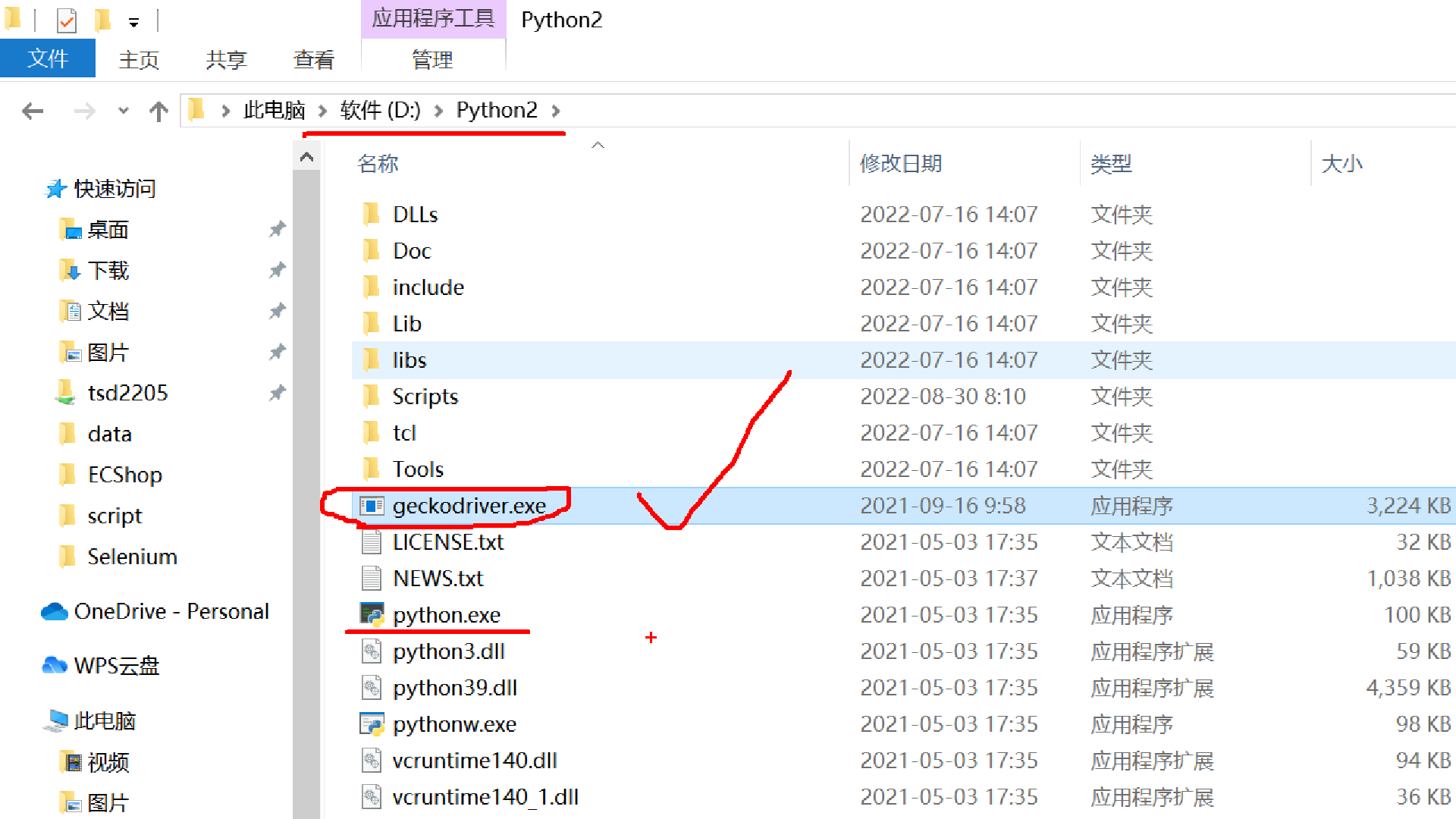
from selenium import webdriver

1. 准备浏览器：Firefox ---略！

安装在默认路径（C:\Program Files (x86)\Mozilla Firefox）下，不要自定义路径

1. 准备浏览器的驱动程序---重要！
2. 按照浏览器的不同，去不同的网站下载驱动程序，解压缩到本地python解释器路径下即可。
3. 说明：一般浏览器使用最新版，驱动程序也用最新版，这样才能兼容。
4. 案例：Firefox版本是100--104（最新，较新），驱动程序是老师下发的资料包里的geckodriver-v0.30.0-win64.zip，解压到当前文件夹，把geckodriver.exe复制到python.exe所在的那个路径里。





说明：各种浏览器的驱动程序的下载地址参考老师下发的资料包里的说明文档<drivers\_download\_url.txt>

1. 测试整体环境：

from selenium import webdriver  
driver=webdriver.Firefox()#启动Firefox

1. 常见错误：
2. ……geckodriver……PATH……

驱动程序文件没有准备好

要么修改代码：

driver=webdriver.Firefox(executable\_path='D:\\Python2\\geckodriver.exe')

1. ……browser binary……

无法自动找到浏览器的启动文件（firefox.exe）

需要卸载Firefox，重新安装在默认路径下

要么修改代码：

driver=webdriver.Firefox(firefox\_binary='C:\\Program Files (x86)\\Mozilla Firefox\\firefox.exe')

1. HTML基础知识
2. HTML是超文本标记语言，是设计网页的主要语言。
3. 为什么测试人员学习HTML？

Selenium自动化测试定位元素的前提是，需要了解HTML语法所实现的网页元素的表示方式，利用元素的表示方式来进行定位方法的调查。---学习强度不用会写，会读就行。

1. html网页基本结构

<html> ---网页开始

<head> ---网页头开始  
……

<title>网页标题</title>

……

</head> ---网页头结束

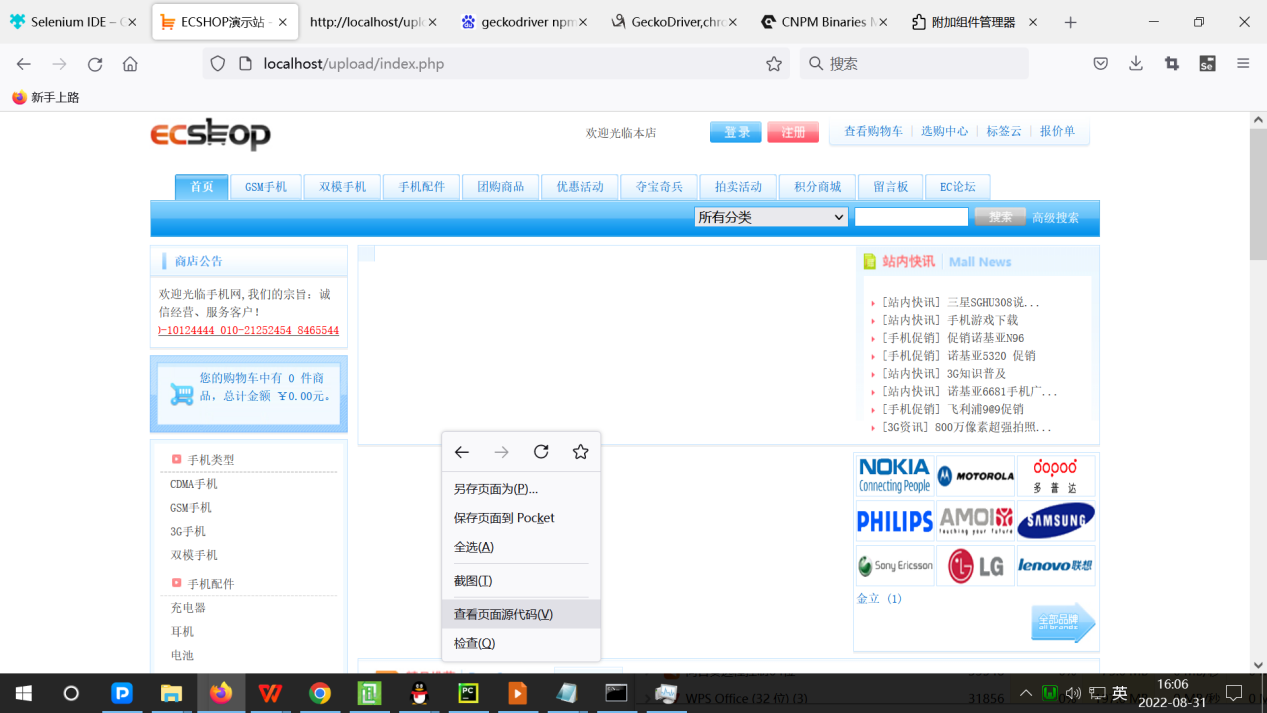
<body> ---网页体开始

…… ---包含网页内的所有元素

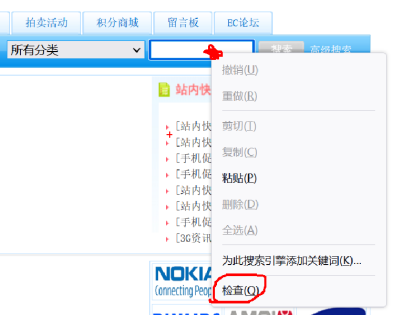
</body> ---网页体结束

</html> ---网页结束

1. 查看网页源代码的两种方式：
2. 方式一：网页任意位置---右击---菜单里选择“查看页面源代码”。---不建议！只看整体。



1. 方式二：网页目标元素---右击---菜单里选择“检查”---建议！看具体的一行代码。





1. HTML标记（或叫“标签”）：tag，用<>表示的基本信息。
2. 单标记

<标记 属性名1=属性值1 属性名2=属性值2 >

1. 双标记

<标记 属性名1=属性值1 属性名2=属性值2>内容</标记>

说明：如果是双标记的内容里，再出现其他标记，称为标记之间的嵌套（上下级、父子）。

1. HTML中常见的元素：
2. 超级链接（简称为“超链接”或“链接”）：用于跳转网页，标记名称是a

<a href="URL"> 高级搜索</a>

1. 图片：标记名称是img

<img src="图片文件的路径">

说明：a和img经常一起搭配使用

<a……><img……></a>

1. 下拉列表：select标记，下级（子）是option标记代表一个选项，用于选择选项。

<select ……>

<option value="值1">选项文本1</option>

<option value="值2">选项文本2</option>

</select>

1. 列表：用于罗列显示多个信息。

<ul>

<li>内容1</li>

<li>内容2</li>

</ul>

说明：ul也可以换为ol（有序列表）

1. 表格：用于把一块网页区域划分为多个行（tr）和列，进行信息的显示。

<table>

<tbody>

<tr>

<th>车次</th>

<th>开车时间</th>

</tr>

<tr>

<td>T123</td>

<td>18:00</td>

</tr>

<tr>

<td>T567</td>

<td>19:00</td>

</tr>

</tbody>

</table>

1. 层（图层）：标记名称是div，用于进行界面布局时，把一块区域划分为多个区域。

<div>内容</div>

1. 表单：表单标记是form，用于收集客户填写的信息，提交给后台服务器。

<form action="xx.php" method="post">

……

</form>

1. 文本框：标记是input，type属性值是text（普通文本框）或password（密码文本框）。文本框有一个隐藏属性value，记录文本框里的当前值。

<input type="text">

<input type="password">



1. 按钮：标记是input，那么type属性值可能是“submit”（提交按钮，最常用！）、“reset”（重置按钮）、“file”（文件按钮，浏览、上传功能使用的按钮）、“button”(少见、与表单无关的按钮)

<input type="submit">

<input type="reset">

<input type="file">

<input type="button> ---少见！

<button></button> ---少见！





1. 复选框：标记是input，type属性值是checkbox

<input type="checkbox" checked="checked"> ---默认被选中

<input type="checkbox"> ---默认没有被选中



1. 单选按钮（单选钮）：一组单选按钮是互斥的，标记是input，type属性值是radio，一组有相同的name属性值的单选按钮是互斥的，互斥的单选按钮有不同的value属性值。



Day02

1. HTML基础知识（续）
2. 其他常见的元素：
3. 文本域（富文本框、多行文本框）：标记是textarea。能输入多行文本信息，数据行数多时，自动出现滚动条。



1. 静态文本（纯文本）：标记名称不限定，但是文本内容都出现在双标记两个尖括号之间。

示例：

<b>评论</b>

<span>留言板</span>

<h1>标题</h1>

1. 通用的属性：不限定标记
2. id属性：唯一标识，好比是人的身份证号
3. name属性：名称，好比是人的姓名
4. value属性：元素里的值（内容）
5. class属性：类名，好比是人的职业
6. style属性：风格/格式
7. 定位元素的方法简介 ---重点！！！！！！
8. 什么是定位元素？

定位元素就是指到网页里找到本次要操作或检查的元素。

1. 定位的依据是什么？

HTML网页源代码里目标元素的表示方法，其中要使用的是元素的标记、属性值、嵌套关系（上下级关系、父子关系）。

1. 定位方法的分析步骤：
2. 先在网页里目标元素右击菜单里选择“检查”，查看到目标元素的那行源代码
3. 分析它的标记、属性信息、父子关系
4. 使用合适的定位方法，书写定位要用到的数据
5. 定位的策略（定位的类型、定位的方法名）：---最常见的面试题！！！
6. ID定位：用id属性值定位
7. NAME定位：用name属性值定位
8. LINK\_TEXT定位：用完整文本来定位超级链接
9. PARTIAL\_LINK\_TEXT定位：用一部分文本来定位超级链接
10. CLASS\_NAME定位：用类名定位
11. TAG\_NAME定位：用标记名称（或叫标签名称）定位
12. XPATH定位：用xpath表达式来定位
13. CSS\_SELECTOR定位：用css的选择器来定位
14. 定位元素的代码的语法：
15. 导入描述定位策略的专用类By

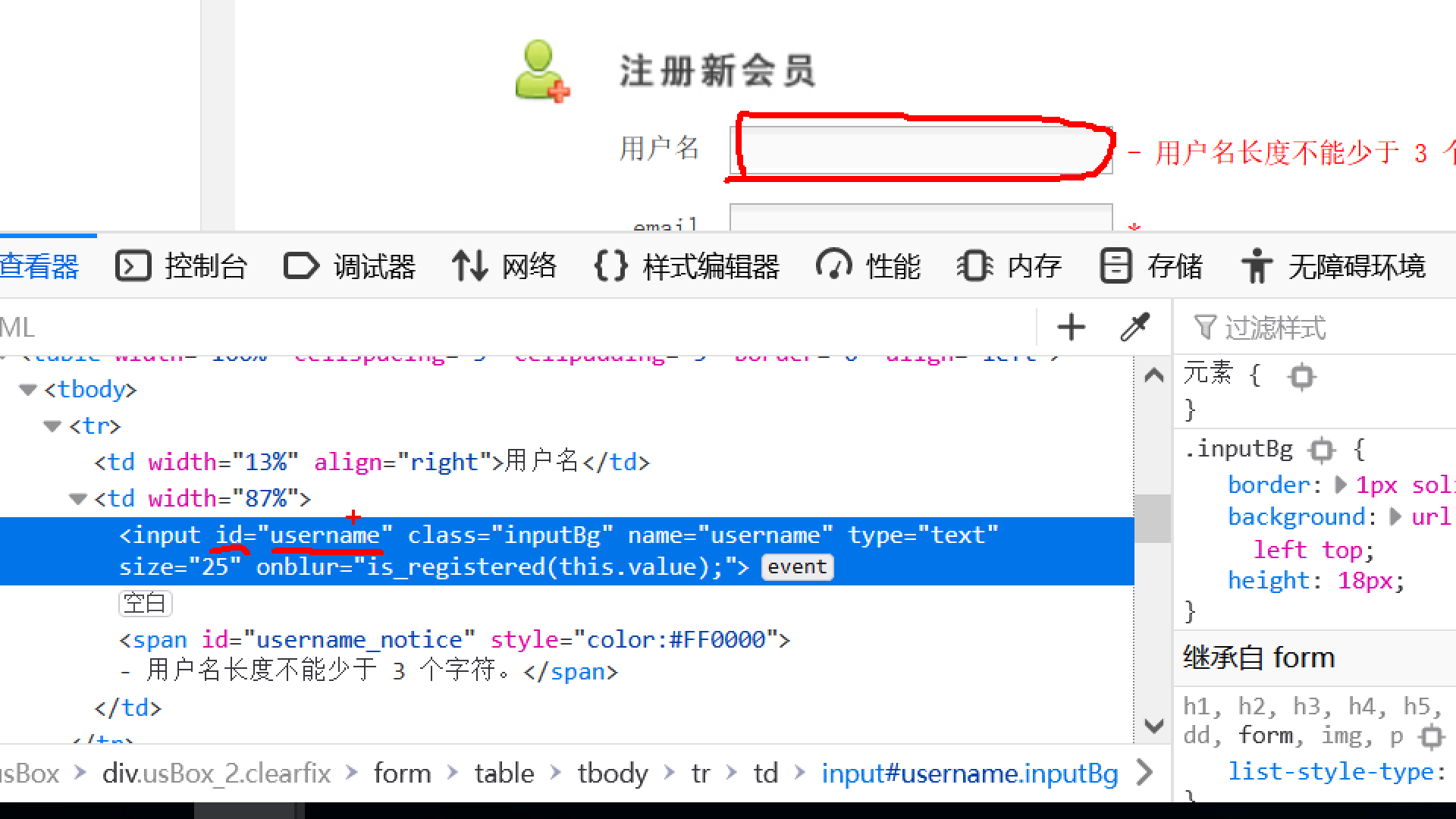
from selenium.webdriver.common.by import By

1. 定位元素的代码

driver.find\_element(By.XXXX,'定位数据')

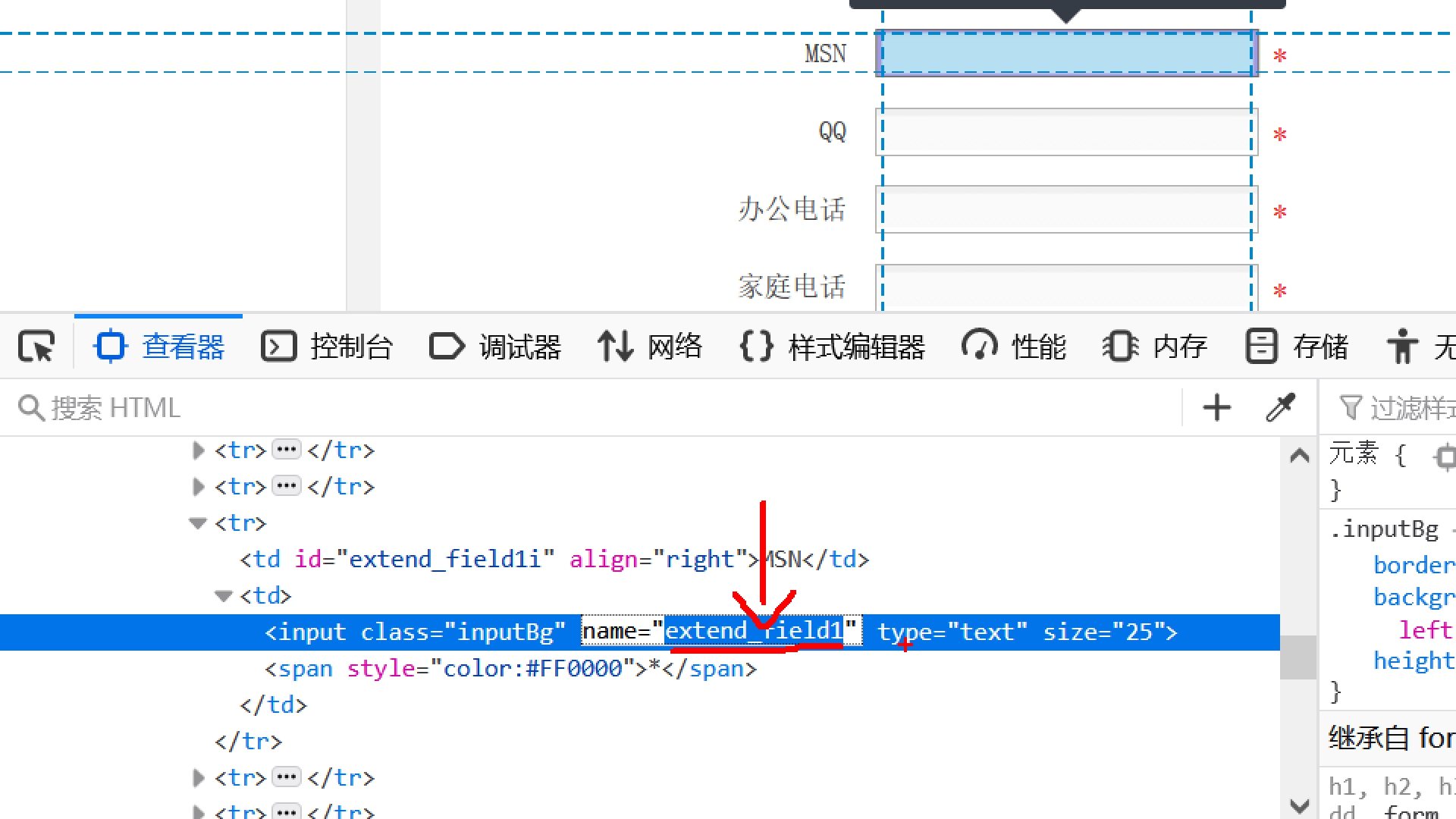
1. 定位到元素后，常见的操作方法：
2. click方法：无参，点击操作。
3. send\_keys方法：有一个参数，输入参数的数据。
4. 定位元素的基础方法（前6个）
5. ID定位：
6. 语法：driver.find\_element(By.ID,'id属性值')
7. 案例：注册页里的用户名文本框

driver.find\_element(By.ID,'username').send\_keys('123')*#向用户名文本框里输入123*



1. 说明：ID定位是通用的定位方法，不限定标记，任何元素都可以使用ID定位，定位之后，可以按照元素的控件类型，调用各种操作方法进行操作。
2. 如果没有id属性，或用id定位失败（少见，id属性值不是固定值引起的），这两种情况下，不能使用ID定位。
3. NAME定位：
4. 语法：driver.find\_element(By.NAME,'name属性值')
5. 案例：注册页里的MSN文本框

driver.find\_element(By.NAME,'extend\_field1').send\_keys('111')*#输入MSN 111*



1. 案例2：

driver.find\_element(By.NAME,'agreement').click()*#点击复选框*



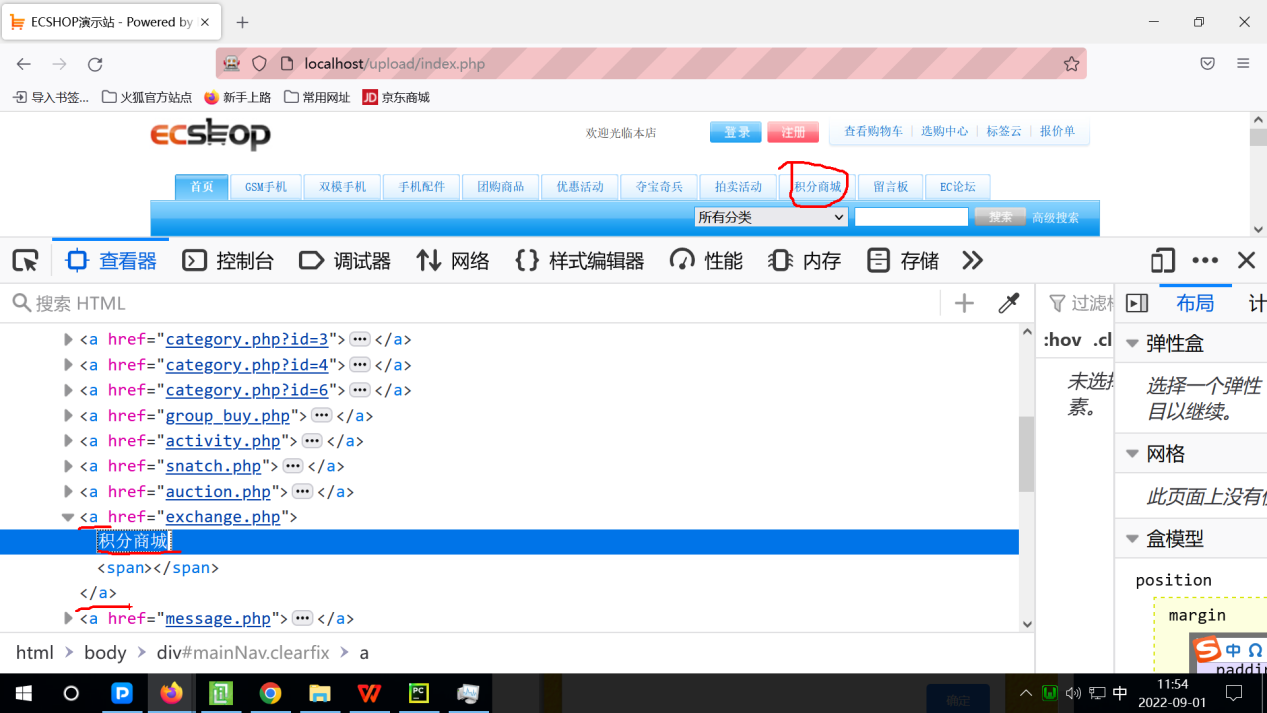
1. NAME定位也是通用的定位方法
2. LINK\_TEXT定位：
3. 语法：driver.find\_element(By.LINK\_TEXT,'完整的链接文本')
4. 说明：完整的链接文本指网页源代码里<a ……>与</a>之间的字符串
5. 注意：只有超级链接才能使用这种定位方法，不是通用的定位方法。
6. 案例：前台首页右上角的“查看购物车”

driver.find\_element(By.LINK\_TEXT,'查看购物车').click()*#点击'查看购物车'这个超级链接*



1. 特殊说明：既有文本、又有子元素的超级链接，也可以使用链接文本进行定位。

driver.find\_element(By.LINK\_TEXT,'积分商城').click()



1. PARTIAL\_LINK\_TEXT定位：模糊匹配，当链接文本较长时使用。
2. 语法：driver.find\_element(By.PARTIAL\_LINK\_TEXT,'一部分链接文本')
3. 选取一部分文本的原则：该链接独有（特有）。

注意：尽量不要选取带有空格或特殊符号的字符串部分，而且避开可能会变化的一些数字等内容。

1. 案例：

driver.find\_element(By.PARTIAL\_LINK\_TEXT,'总计金额').click()



1. CLASS\_NAME定位：建议少用！
2. 语法：driver.find\_element(By.CLASS\_NAME,'类名')
3. 类名：class属性值里如果没有空格，那么class属性值就是唯一的类名，如果有空格，就是用空格所分隔开的字符串中的其中一个类名。

<xx class="类名">

<xx class="类名1 类名2 类名3">

1. 说明：元素的类名不是唯一标识信息，重复概率大，所以建议少用，定位不准确。
2. 案例：首页上左下角的“查询该订单号”的按钮，它既没有id属性，也没有name属性，而且也不是a标记的超级链接，有class属性。

driver.find\_element(By.CLASS\_NAME,'bnt\_blue\_2').click()*#点击“查询该订单号”按钮*



综合练习：新建demo2004.py

<1>启动Firefox

<2>打开ECShop前台首页

<3>点击“留言板”,等待3秒

<4>输入电子邮件地址：[jack@126.com](mailto:jack@126.com)

<5>输入主题：hello

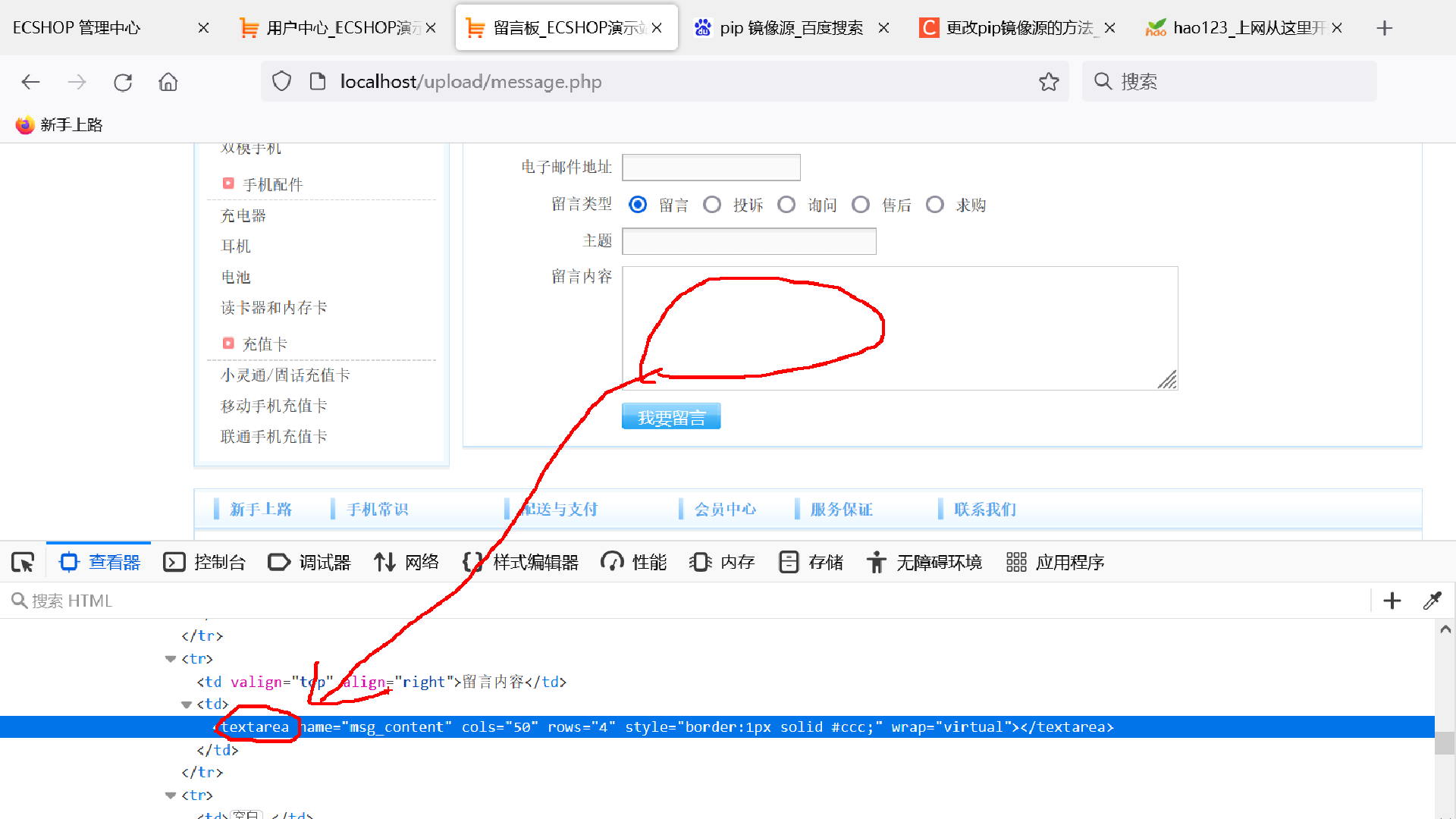
<6>输入留言内容：你好，等待3秒

<7>点击“我要留言”按钮，等待5秒

<8>关闭浏览器

1. TAG\_NAME定位：建议少用！
2. 语法：driver.find\_element(By.TAG\_NAME,'标记名')
3. 案例：

driver.find\_element(By.TAG\_NAME,'textarea').send\_keys('789')



补充：强制等待一段时间（线程休眠），可以使用python语言自带的time模块中的sleep函数来实现，参数是以秒为单位的时间数。

from time import sleep *#导入time模块里的sleep函数*

sleep(5)*#等待5秒*

1. 高级定位方法之XPATH定位
2. 应用场景：针对目标元素，发现用前6种基础定位方法都不能定位，考虑使用高级定位方法。
3. 什么是XPATH？
4. XML：可扩展标记语言，是对HTML的扩展，标记名和属性都可以自定义，通常作为配置文件来使用。

<书店 书店名=“新华书店”>

<图书>

</书店>

1. XPATH：XML PATH（XML路径语言），是在XML文件里描述节点所在的路径来查找一个或多个节点的一种技术。

xpath表达式：/书店/图书

1. 因为HTML完全符合XML的语法格式，所以XPATH也可以用于在HTML网页里查找元素。

<html>

<head>

</head>

<body>

<div> <a……>文本</a></div>

<div></div>

</body>

</html>

xpath表达式：/html/body/div[1]/a

1. 初学者获得xpath表达式：

Firefox---打开网页---操作到目标元素所在的页面---目标元素---右击---检查---在目标源代码的代码行上右击---复制---XPath



/html/body/div[1]/div[2]/ul/li[1]/font/a[1]/img

1. 使用XPath表达式定位元素的语法：

driver.find\_element(By.XPATH,'xpath表达式')

1. Xpath表达式分类：
2. 绝对路径XPath表达式：从根节点开始进行描述，以/开头。 维护性差，表达式长。----建议尽量少用！

示例：

/html/body/div[1]/div[2]/ul/li[2]/a[1]

1. 相对路径XPath表达式：不从根节点开始进行描述，以//开头（或.//开头）。维护性更好，更简短。----建议使用！

示例：

//\*[@id="keyword"]

1. 校验xpath表达式是否正确：利用Firefox的“查看器”

输入表达式，回车，后面显示结果数

n/m n代表当前是第几个,m代表定位到的元素总数。



1. 相对路径Xpath表达式的语法：
2. //标记 ---在整个网页里查找该标记的元素。

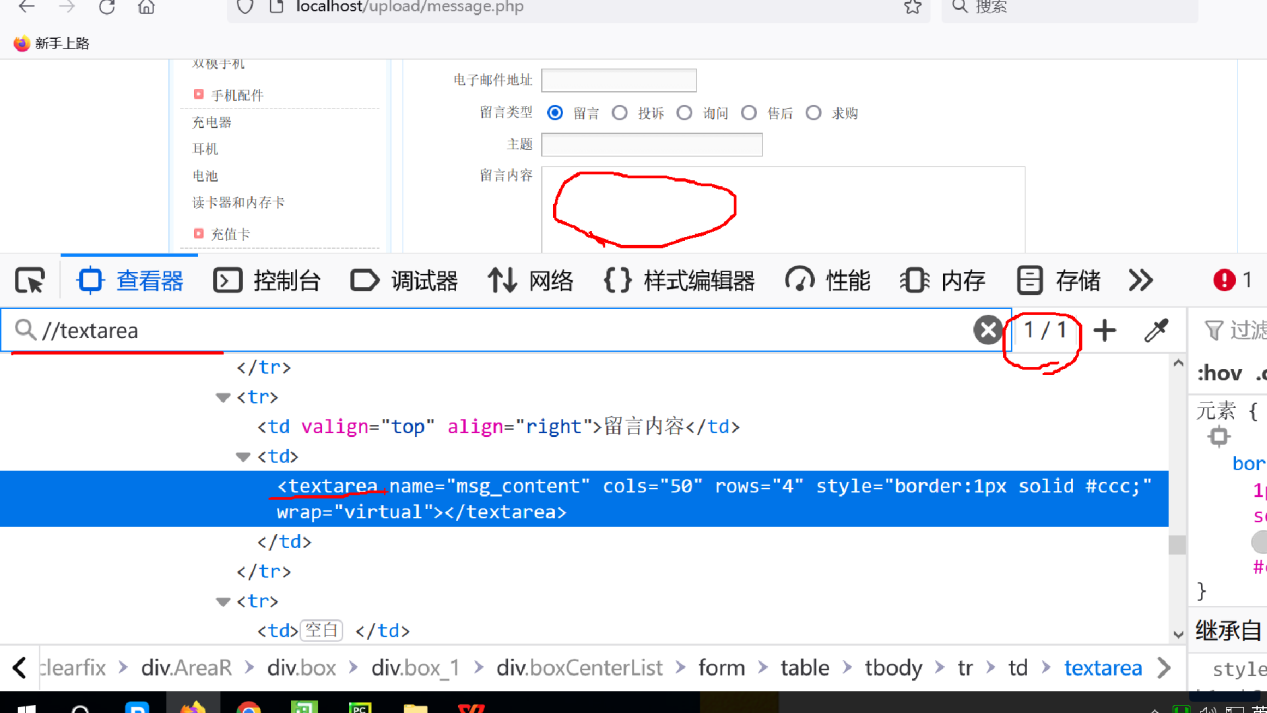
示例：

//a

//input

//img

//textarea --- 1/1 可以用于定位“留言内容”的文本域



1. //父节点标记/子节点标记 ---按照父子关系层级进行定位

说明：可以再在父节点前面继续增加祖父节点、曾祖父、高祖父节点……

//祖父/父/子

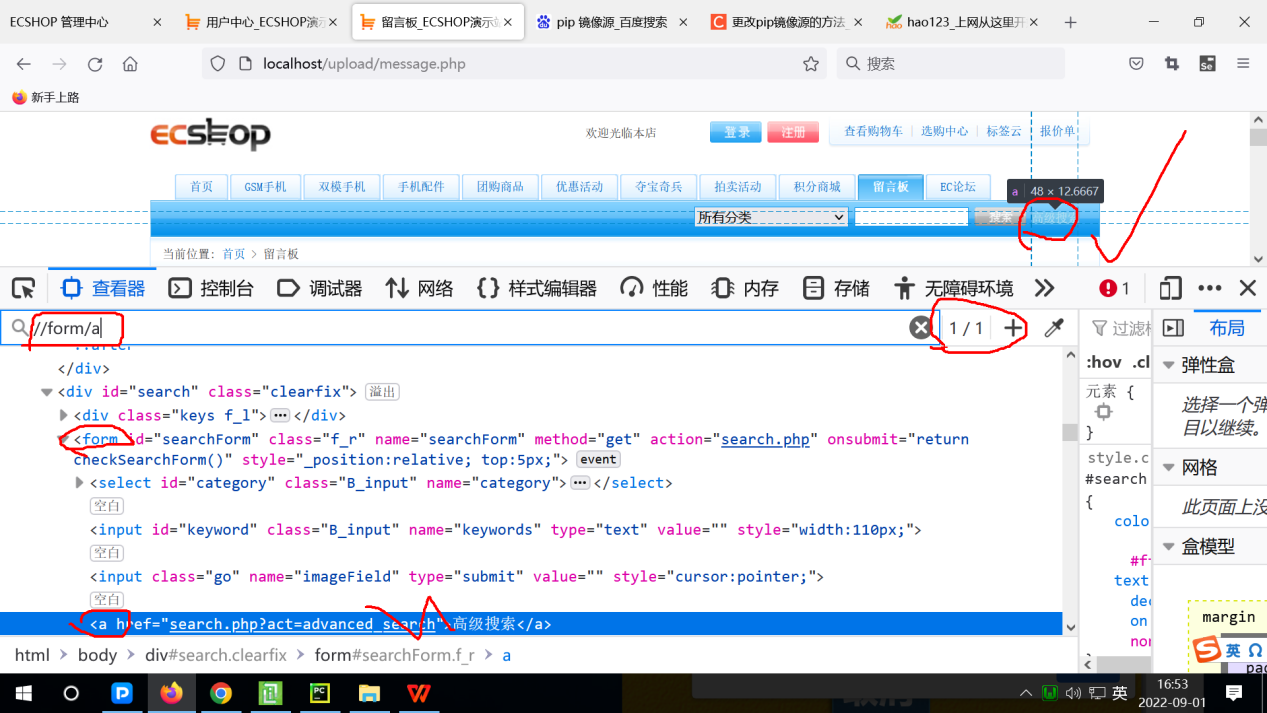
//曾祖父/祖父/父/子

//高祖父/曾祖父/祖父/父/子

……

示例：

//form/a ---在留言板页里的“高级搜索”

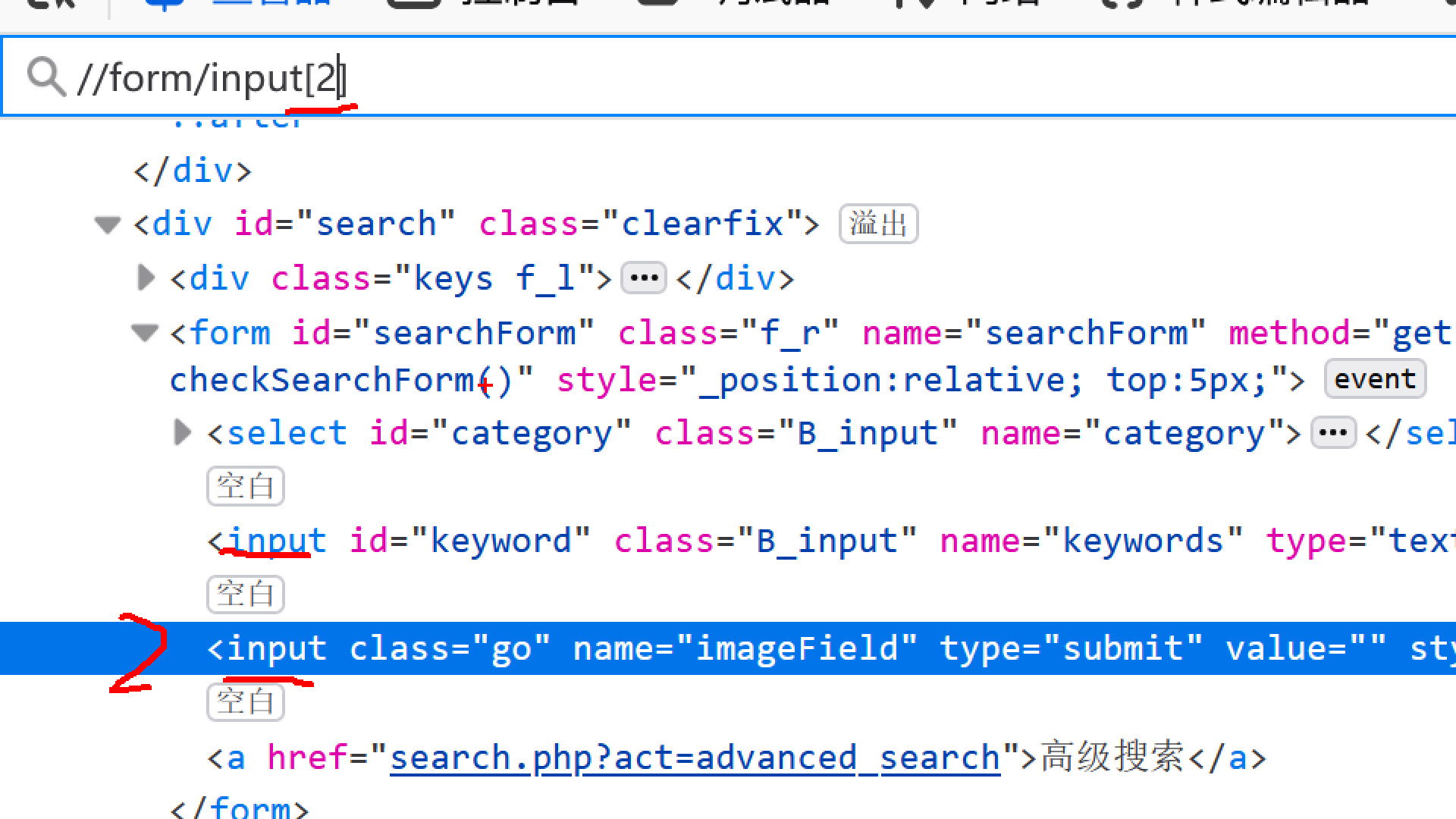


1. 各个层级的标记名称后都可以增加筛选条件：谓词（或叫做谓语），用中括号描述筛选条件。
2. 谓词1： 标记[索引号] 索引号从1开始编号（不是从0开始编号），代表兄弟关系（同父）中同标记的节点的顺序号。

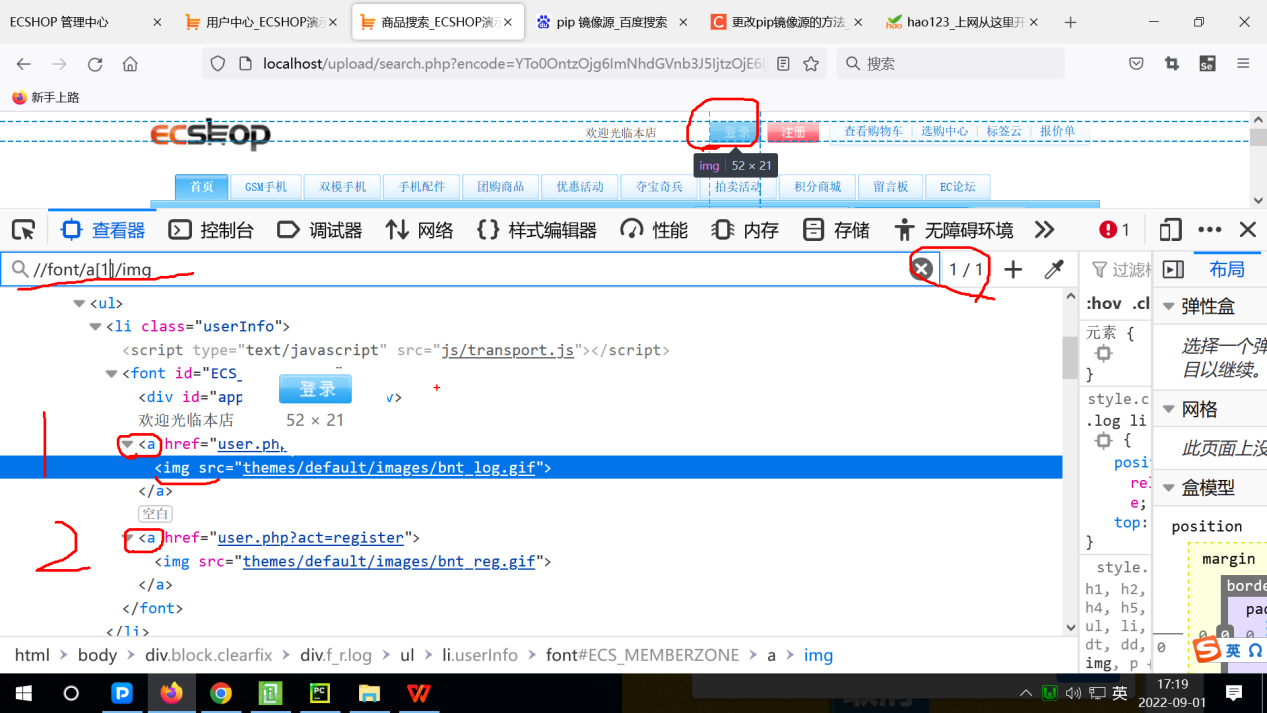
driver.find\_element(By.XPATH,'//form/input[2]').click()*#点击“搜索”*







driver.find\_element(By.XPATH,'//font/a[1]/img').click()*#点击“登录”*



1. 谓词2：标记[@属性名="属性值"] ---用某个属性值来筛选元素。说明：引号支持单引号和双引号。

案例1：

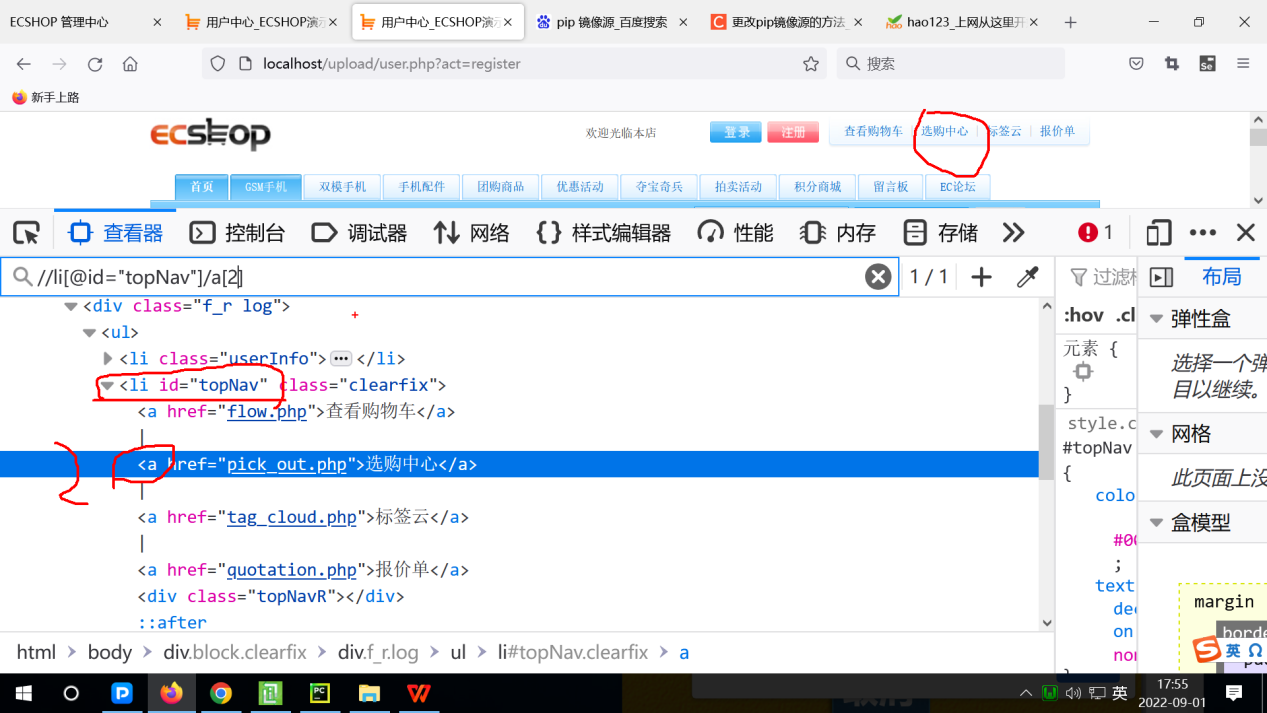
driver.find\_element(By.XPATH,'//a[@name="top"]/img').click()*#点击左上角ECSHOP商标（可以回到首页）*



案例2：

[//a[@href="pick\_out.php"]](mailto:/a[@href=\"pick_out.php\"])

//li[@id="topNav"]/a[2]



练习：demo2007.py ---要求都用自己书写的XPATH定位。

<1>启动Firefox

<2>打开前台登录页，等待2秒

<3>点击“当前位置:”后的“首页”超级链接，等待2秒

<4>点击“积分商城”，等待3秒

<5>点击倒序的下拉列表，等待3秒

<6>点击正序的选项，等待3秒

<7>点击下拉列表后面的GO图片，等待3秒

<8>关闭浏览器