

Software
Testing

软件测试丛书

Selenium 自动化测试指南

Automated Web Testing
with Selenium

赵卓 著

- 51Testing 软件测试网鼎力推荐
- Web 自动化测试的初学者指南
- 全面剖析各语言、各版本的 Selenium 测试工具

The background of the cover is a grid of six panels, each containing a different abstract pattern of light spots and lines on a dark background. The top-left panel is partially obscured by the 'Software Testing' text.

Software
Testing

软件测试丛书

Selenium 自动化测试指南

Automated Web Testing
with Selenium

赵卓 著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Selenium 自动化测试指南 / 赵卓著. -- 北京: 人民邮电出版社, 2013. 9
ISBN 978-7-115-31534-2

I. ①S… II. ①赵… III. ①软件—测试—自动化—指南 IV. ①TP311.5-62

中国版本图书馆CIP数据核字(2013)第068047号

内 容 提 要

Selenium 是 ThoughtWorks 公司开发的 Web 自动化测试工具。Selenium 可以直接在浏览器中运行, 支持 Windows、Linux 和 Macintosh 平台上的 Internet Explorer、Mozilla 和 Firefox 等浏览器, 得到了广大 Web 开发和测试人员的应用。

本书是使用 Selenium 实现 Web 自动化测试的指南。本书共分为 9 章。第 1 章~第 2 章, 介绍了 Selenium 自动化测试相关的基础知识和辅助工具; 第 3 章~第 5 章, 分别介绍了 Selenium IDE、Selenium1 (Remote Control) 以及 Selenium2 (WebDriver) 的用法。第 6 章~第 7 章, 主要介绍实际工作中如何使用 Selenium 来进行测试, 介绍了自动化测试的流程和框架, 并通过实例来讲解自动化测试用例和测试代码的实际编写。第 8 章~第 9 章介绍了 Selenium 测试难点和常见问题的解决。

本书兼顾 Selenium 当前流行和应用的不同版本, 包括丰富的示例和图解。本书适合于测试人员、测试组长、测试经理、质量保证工程师、软件过程改进人员以及相关专业学生阅读, 以快速掌握并在实际工作中使用 Selenium。

-
- ◆ 著 赵 卓
责任编辑 陈冀康
责任印制 程彦红 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 19.5
字数: 390 千字 2013 年 9 月第 1 版
印数: 1-3 000 册 2013 年 9 月河北第 1 次印刷
-

定价: 42.00 元

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

软件测试领域总是在不断地发展：从最开始没有专门测试人员，到终于认可了测试人员的价值。开始的时候测试人员仅执行繁琐的手工测试，逐渐发展到使用自动化测试。而对于自动化测试，也有各种分类，各种自动化测试工具也如雨后春笋般地涌现。

在 Selenium 测试工具出现之前，对于 Web 的自动化功能测试一直没有较好的解决方案。即使是当年很火的 QTP，也很难应对以下复杂的 Web 自动化功能测试的问题。

Web 测试时如何应对不同的浏览器？是否支持高级编程语言？对于不同的平台，例如 Windows，IOS 又如何处理？即使这些问题都能解决，那该工具的价格是否不菲？

在这种情况下，ThoughtWorks 公司发布了 Selenium 测试工具。该工具拥有如下特性。

- (1) 可对多浏览器进行测试，例如 IE、Firefox、Safari、Chrome、Android 手机浏览器等。
- (2) 支持各种语言，例如 Java、C#、Python、Ruby、PHP 等。
- (3) 跨平台，例如 Windows、Linux、iOS、Android 等。
- (4) 开源免费。

使用 Selenium 测试工具，终于解决了 Web 自动化功能测试的难题，而且它使用起来非常便捷。

写作本书的目的

我已经在自动化测试领域工作好几年了，测试过大大小小的项目，接触了各种各样的测试工具，Selenium 是其中的一种，它是 Web 自动化功能测试最好用的一款工具。

记得第一次使用 Selenium 大约是在两年前，客户要求对公司网站进行测试，具体要求是支持多浏览器，越多越好；最好是 C#；最好是免费的。

我使用搜索引擎在互联网上查找，惊讶地发现还真有工具能满足如此苛刻的要求，这个工具就是 Selenium。通过 Selenium，我顺利完成了公司网站自动化测试项目第一期的任务。

到了第二期，公司招募了更多的人来完成该测试项目，我则负责指导测试人员完成该项目的实施。然而问题出现了：由于 Selenium 本身要求测试人员拥有较好的测试基础，同时 Selenium 相关的资料相当匮乏，导致测试人员的学习进度和工作效果都不尽如人意，其中有些测试人员购买了一些早期的 Selenium 相关的书籍，却也发现它们写的不够详尽，缺乏对于 Selenium 2 的描述，而对于 Selenium 1 的描述则不够清晰，看了之后不知道该怎么用。

这时候，我开始有了写本书的想法，一则是希望在今后的项目中，大家都能有比较详细的资料能够参考；二则是希望能帮助所有正在学习或使用 Selenium 的读者，希望通过分享自己在使用 Selenium 的心得体会，达到共同学习和共同提高的目的。

适用读者

本书主要适用于测试人员、测试经理、质量保证工程师、软件过程改进人员以及相关专业的在校学生和实习生，同时也适用于对自动化测试感兴趣的开发人员、项目主管和经理等。

如何阅读本书

本书共分为 9 章，由浅入深介绍了使用 Selenium 测试工具进行软件测试的各个方面，即使读者不具备任何开发和测试功底，仍可以阅读。

第 1 章至第 2 章介绍了自动化测试相关的基础、HTML/XML/XPath 等语言基础以及 Web 测试时常用到的辅助工具，适用于很少接触应用软件测试和 Web 网页测试，也没有自动化测试基础的读者阅读。

第 3 章至第 5 章分别对 Selenium IDE、Selenium 1 (Remote Control) 以及 Selenium 2 (WebDriver) 的用法进行介绍。这几章适合拥有自动化测试基础但不了解 Selenium 的读者阅读。对于比较了解 Selenium 的读者，也可以从中温故而知新。建议大家多花些精力和时间进行研究。

第 6 章至第 7 章为 Selenium Web 测试实战，主要介绍在实际工作中如何使用 Selenium 进行测试，并介绍了自动化测试的流程和框架。这部分还以 www.360buy.com 为例，介绍自动化测试用例的编写以及测试代码的实际编写。建议大家对这部分的实例进行仔细研究。

第 8 章至第 9 章为 Selenium 测试难点，分别介绍了如何对 Opera/iPhone/Android 进行测试、如何切换 Selenium 1 和 Selenium 2、常见问题的解决以及如何对 Selenium 进行扩展。如

果在使用 Selenium 进行测试的过程中遇到了疑难问题，可以参考本章。

读者可以根据自己的需求选择阅读侧重点，不过最好按照顺序来阅读，这样不仅仅可以循序渐进，还可以从整体上对 Selenium 测试有一个清晰的认识。

致谢

首先，感谢全体 Selenium 的制作人员，正是他们敢于创新、乐于分享的精神才造就了如此强大易用的工具。

同时，非常感谢人民邮电出版社的各位编辑对我的指导和帮助，以及认真细致的工作，才使得书稿得以完善和出版。尤其感谢陈冀康先生，在本书写作过程中给予我的信任、支持和鼓励。正是有了你们对我的帮助，本书才有机会为广大的读者知晓。

当然，感谢我的家人，正是由于他们默默的支持，我才能静下心来写作。同时感谢我的伙伴和同事们，因为大家共同的努力才顺利完成了本书的编写。

关于勘误

虽然花了很多时间和精力去核对书中的文字、代码和图片，但因为时间仓促和水平有限，书中仍难免会有一些错误和纰漏，如果大家发现什么问题，恳请反馈给我，相关信息可发到我的邮箱 realdigit@163.com。敬请广大读者及同行批评指正。

目 录

第 1 章 Selenium 自动化测试基础	1	3.5.6 DOM 定位	76
1.1 自动化测试基础	2	3.5.7 CSS 定位	77
1.1.1 软件测试概述	2	3.5.8 隐式定位	78
1.1.2 自动化测试简介	4	3.6 Value	78
1.1.3 Selenium 简介	5	3.6.1 带变量的字符串	78
1.2 HTML/XML/XPath 基础	7	3.6.2 带 JavaScript 的字符串	79
1.2.1 HTML 简介	7	3.7 日志与引用	79
1.2.2 XML 简介	9	3.8 将命令复制或导出为代码	80
1.2.3 使用 XPath 进行 元素定位	10	3.8.1 将命令复制为代码	80
第 2 章 必备辅助工具	17	3.8.2 将命令导出为代码	82
2.1 FireBug	18	第 4 章 Selenium 1 (Remote Control)	84
2.2 FirePath	22	4.1 Selenium 1 的工作原理	85
第 3 章 Selenium IDE	25	4.1.1 Selenium 1 的组件	85
3.1 Selenium IDE 的安装	26	4.1.2 Selenium 服务器	86
3.2 录制测试用例	27	4.1.3 Selenium 客户端库文件	86
3.3 编写测试用例	30	4.2 安装并使用 Selenium	86
3.4 Command	33	4.2.1 在 C# IDE 中使用 Selenium	87
3.4.1 Action	34	4.2.2 在 Java IDE 中使用 Selenium	92
3.4.2 Accessor	46	4.3 下载并启动 Selenium 服务器	99
3.4.3 Assertion	59	4.4 选择浏览器开始测试	100
3.5 Target	71	4.5 浏览器导航操作	102
3.5.1 identifier 定位	71	4.5.1 Open(url)	102
3.5.2 id 定位	72	4.5.2 GoBack(url)	103
3.5.3 name 定位	73	4.5.3 Refresh(), WindowFocus(), WindowMaximize()和	
3.5.4 XPath 定位	74		
3.5.5 链接文字定位	75		

- Close() 104
- 4.6 操作页面元素 105
 - 4.6.1 Type (locator, value) 105
 - 4.6.2 TypeKeys (locator, value) ... 106
 - 4.6.3 Click (locator) 107
 - 4.6.4 ClickAt (locator, coordString) 107
 - 4.6.5 doubleClick (locator) 108
 - 4.6.6 doubleClickAt (locator, coordString) 108
 - 4.6.7 Select (selectLocator, optionLocator) 108
 - 4.6.8 check (locator)/unCheck (locator) 110
 - 4.6.9 focus (locator) 110
- 4.7 键盘鼠标模拟操作 110
- 4.8 获取页面元素的内容 112
 - 4.8.1 getTitle () 112
 - 4.8.2 getLocation() 114
 - 4.8.3 getValue (locator) 114
 - 4.8.4 IsEditable (locator) 115
 - 4.8.5 getText (locator) 116
 - 4.8.6 isChecked (locator) 118
 - 4.8.7 getSelectedIndex (selectLocator) 119
 - 4.8.8 getSelectedLabel (selectLocator) 121
 - 4.8.9 getSelectedValue (selectLocator) 122
 - 4.8.10 getSelectOptions (selectLocator) 123
 - 4.8.11 getTable(tableCell Address) 124
- 4.8.12 getAttribute (attributeLocator) 126
- 4.8.13 isTextPresent (pattern) ... 127
- 4.8.14 isElementPresent (locator) 128
- 4.8.15 isVisible (locator) 130
- 4.8.16 getXpathCount (locator) 131
- 4.9 设置等待 132
 - 4.9.1 WaitForPageToLoad (timeout) 132
 - 4.9.2 setTimeout (timeout) 133
 - 4.9.3 setSpeed (value) 133
- 4.10 测试控制和调试类操作 134
 - 4.10.1 captureEntirePage Screenshot (filename, kwargs) 134
 - 4.10.2 captureEntirePage Screenshot (filename) ... 135
 - 4.10.3 highlight (locator) 136
- 4.11 JavaScript 弹出对话框的处理 136
 - 4.11.1 IsAlertPresent() 138
 - 4.11.2 GetAlert() 139
 - 4.11.3 IsConfirmationPresent() ... 140
 - 4.11.4 GetConfirmation() 141
 - 4.11.5 ChooseOkOnNext Confirmation()和 Choose CancelOnNext Confirmation() 142
 - 4.11.6 IsPromptPresent() 142

- 4.11.7 GetPrompt()..... 143
- 4.11.8 AnswerOnNextPrompt
(answer)..... 144
- 4.12 浏览器多窗口处理..... 144
 - 4.12.1 GetAllWindowIds()、
GetAllWindowsNames()和
GetAllWindows
Titles()..... 145
 - 4.12.2 WaitForPopUp (windowID,
timeout)..... 148
 - 4.12.3 SelectPopUp(windowID)和
SelectWindow
(windowID)..... 149
 - 4.12.4 OpenWindow (url,
windowID)..... 150
- 4.13 结束测试..... 151
 - 4.13.1 Close()和 Stop()..... 151
 - 4.13.2 shutDownSelenium
Server()..... 152
- 第 5 章 Selenium 2 (WebDriver) 153**
 - 5.1 Selenium 2——基于对象的
测试..... 154
 - 5.2 安装并引用 Selenium 2..... 155
 - 5.2.1 在 C# IDE 中使用
Selenium..... 156
 - 5.2.2 在 Java IDE 中使用
Selenium..... 160
 - 5.3 选择浏览器开始测试..... 166
 - 5.4 浏览器导航对象 Navigation..... 169
 - 5.4.1 GoToUrl()/to()..... 169
 - 5.4.2 Back()/Forward()..... 170
 - 5.4.3 Refresh()..... 172
 - 5.5 查找条件对象 By..... 173
 - 5.5.1 Id(idToFind)..... 173
 - 5.5.2 Name(nameToFind)..... 175
 - 5.5.3 LinkText(linkText
ToFind)..... 175
 - 5.5.4 PartialLinkText(partial
LinkTextToFind)..... 176
 - 5.5.5 ClassName(className
ToFind)..... 177
 - 5.5.6 TagName(TagName
ToFind)..... 178
 - 5.5.7 XPath(xpathToFind)..... 179
 - 5.6 操作页面元素 WebElement..... 179
 - 5.6.1 Click()..... 180
 - 5.6.2 SendKeys(text)..... 183
 - 5.6.3 Clear()..... 184
 - 5.6.4 Submit()..... 185
 - 5.7 获取页面及页面元素的内容..... 187
 - 5.7.1 Title/getTitle()..... 187
 - 5.7.2 Url/getCurrentUrl()..... 188
 - 5.7.3 Text/getText ()..... 189
 - 5.7.4 Selected/isSelected()..... 190
 - 5.7.5 TagName/getTagName()..... 191
 - 5.7.6 Enabled/isEnabled()..... 192
 - 5.7.7 Displayed/isDisplayed()..... 193
 - 5.7.8 GetAttribute(attribute
Name)..... 194
 - 5.8 弹出对话框的处理..... 195
 - 5.8.1 Accept()..... 197
 - 5.8.2 Dismiss()..... 198
 - 5.8.3 SendKeys(keysToSend)..... 200
 - 5.8.4 Text/getText()..... 201

5.9 浏览器多窗口处理.....	202	7.2.3 购买商品功能的测试 代码.....	262
5.9.1 WindowHandles/get WindowHandles().....	203	7.3 设计自动化测试数据.....	268
5.9.2 Window(windowName).....	204	7.3.1 登录功能的数据.....	268
5.10 设置管理.....	205	7.3.2 搜索商品功能的数据.....	269
5.10.1 Cookies/getCookies().....	206	7.3.3 购买商品功能的数据.....	271
5.10.2 Window/window().....	210	第 8 章 Selenium 测试进阶	272
5.10.3 Timeouts().....	212	8.1 Opera/iPhone/Android 测试.....	273
5.11 为测试操作添加事件.....	213	8.1.1 Opera 测试.....	273
5.11.1 在 C#中使用 Event FiringWebDriver.....	214	8.1.2 iPhone 测试.....	276
5.11.2 在 Java 中使用 Event FiringWebDriver.....	219	8.1.3 Android 测试.....	279
5.12 结束测试.....	225	8.2 Selenium 1 与 Selenium 2 的 切换.....	282
第 6 章 自动化测试的流程和框架	229	8.2.1 从 Selenium 1 切换到 Selenium 2.....	282
6.1 自动化测试的流程.....	230	8.2.2 从 Selenium 2 切换到 Selenium 1.....	287
6.2 自动化测试框架.....	232	8.3 对 Selenium 进行扩展.....	287
第 7 章 自动化测试的实施	236	8.3.1 对 Selenium IDE 应用 扩展.....	290
7.1 设计自动化测试用例.....	237	8.3.2 对 Selenium 1 应用扩展.....	290
7.1.1 登录功能的用例设计.....	237	8.3.3 对 Selenium 2 应用扩展.....	292
7.1.2 搜索商品功能的用例 设计.....	238	第 9 章 使用 Selenium 常见的问题	294
7.1.3 购买商品功能的用例 设计.....	242	9.1 使用 Selenium IDE 常见的 问题.....	295
7.2 编写自动化测试代码.....	245	9.2 使用 Selenium 1 常见的问题.....	296
7.2.1 登录功能的测试代码.....	245	9.3 使用 Selenium 2 常见的问题.....	301
7.2.2 搜索商品功能的测试 代码.....	252		

Chapter

1

第 1 章

Selenium 自动化测试基础

Selenium 是一系列基于 Web 的自动化测试工具。它提供了许多测试函数，用于支持 Web 自动化测试。这些函数非常灵活，它们能够通过多种方式定位界面元素，并可以将预期结果与系统实际表现进行比较。

在使用 Selenium 进行测试前，必须具备自动化测试相关基础、HTML/XML/XPath 等语言基础，并对 Web 测试经常用到的辅助工具有基本的了解。下面将分别介绍这些知识。

1.1 自动化测试基础

1.1.1 软件测试概述

即使是经验非常丰富的程序员，在编写代码时也很容易出现错误，这些错误也许是由于需求不明确，也许是由于设计问题，也许是编码中出现了失误等。但无论是怎样的错误，若不及时处理，都会降低软件的可靠性，严重时甚至会导致整个软件的失败。

为了排除这些错误，人们引入了软件测试的概念。通俗地说，软件测试就是为了发现程序中的错误而分析或执行程序的过程。

据研究机构统计分析表明，国外软件开发机构 40% 的工作量都花在软件测试上，软件测试费用占软件开发总费用的 30%~50%。对于一些要求高可靠、高安全的软件，测试费用所占的比例更高。由此可见，要成功开发出高质量的软件产品，软件测试必不可少。

软件测试的主要工作是验证（Verification）和确认（Validation）。

验证是保证软件正确地实现了一些特定功能的一系列活动，即保证软件以正确的方式做了该做的事。具体地讲，验证主要完成以下任务。

- (1) 确定软件生存周期中一个给定阶段的产品是否达到当前阶段确立的需求。
- (2) 程序正确性的形式证明，即采用形式理论证明程序符合设计规约的规定。

(3) 评审、审查、测试、检查、审计等，或对某些项处理、服务或文件等是否和规定的需求相一致进行判断并进行报告。

确认（Validation）的目的是想证实在一个给定的外部环境中软件的逻辑正确性，即保证软件做了所期望的事情。

- (1) 静态确认，不在计算机上实际执行程序，通过人工或程序分析来证明软件的正确性。
- (2) 动态确认，通过执行程序进行分析，测试程序的动态行为，以证实软件是否存在问题。

测试和改正活动可以在软件生命周期的任何阶段进行。然而，随着开发的不断进行，找出并修正错误的代价也会急剧增加。在需求阶段就对其进行修改，付出的代价会很少。如果代码已经编写完毕，再进行更改，则将付出代价会大许多。

软件测试的分类

从是否关心软件内部结构和具体实现的角度来看，软件测试可以划分为以下几类。

- 白盒测试：需要了解内部结构和代码。
- 黑盒测试：不关心内部结构和代码。
- 灰盒测试：介于白盒测试和黑盒测试之间。

从是否执行程序的角度来看，软件测试可以划分为以下几类。

- 静态测试：测试时不执行被测试软件。
- 动态测试：测试时执行被测试软件。

按软件开发过程的阶段划分，软件测试可以划分为以下几类。

- 单元测试：测试软件的单元模块（单元模块指某个功能、某个类等）。
- 集成测试：将各个“单元”集成到一起测试是否能正确运行。
- 系统测试：测试软件是否符合系统中的各项需求。
- 验收测试：类似系统测试，但由用户执行。

按测试的具体目标进行划分，软件测试可以划分为以下几类。

- 功能测试：测试软件是否符合功能性需求，通常采用黑盒测试方法。
- 性能测试：测试软件在各种状态下的性能，找出性能瓶颈。
- 安全测试：测试该软件防止非法入侵的能力。
- 回归测试：在软件被修正或运行环境发生变化后进行重新测试。
- 兼容性测试：测试该软件与其他软件、硬件的兼容能力。
- 安装测试：测试软件的安装、卸载、升级是否正常。

各种软件测试书籍对前面 3 种划分方式介绍得很多，但在实际工作中，更习惯将测试按具体目标进行划分。

1.1.2 自动化测试简介

软件测试一般是由测试人员执行的，如果由机器来代替人执行软件测试，那么这种测试就叫自动化测试。例如，由计算机代替人来单击被测试软件的界面，执行一系列操作并进行验证。

在各种测试中，都可以引入自动化测试。

通过执行自动化测试，可以节省大量的人力、时间或硬件等资源，同时也能大大提高测试效率。

当然，这听起来很好，但如果要由机器代替人来执行软件测试，就得开发相应的自动化测试代码。开发自动化测试代码也是需要人力、时间或硬件等资源的。因此，在资源有限的情况下，必须对软件开发过程进行分析，观察哪些地方适合使用自动化测试。使用自动化测试通常需要同时满足以下条件。

(1) 软件需求变动不频繁。

测试脚本的稳定性决定了自动化测试的维护成本。如果软件需求变动过于频繁，测试人员需要根据变动的需求来更新测试用例以及相关的测试脚本，而脚本的维护本身就是一个代码开发的过程，需要修改和调试，必要的时候还要修改自动化测试的框架，如果所花费的成本不低于利用其节省的测试成本，那么自动化测试便是失败的。

项目中的某些模块相对稳定，而某些模块需求变动性很大。可以对相对稳定的模块进行自动化测试，而变动较大的仍采用手工测试。

(2) 项目周期足够长。

自动化测试的过程包括自动化测试需求的确定、自动化测试框架的设计、测试脚本的编写和调试等，这个过程本身就是一个测试软件的开发过程，需要较长的时间来完成。如果项目的周期比较短，没有足够的时间去支持这样一个过程，那么自动化测试便也无从谈起。

(3) 自动化测试脚本可重复使用。

如果费尽心思开发了一套近乎完美的自动化测试脚本，但是脚本的重复使用率很低，致

使其间所耗费的成本大于所创造的经济价值，自动化测试便成为了测试人员的练手之作，而并非真正可产生效益的测试手段了。

另外，在手工测试无法完成，需要投入大量时间与人力时也需要考虑引入自动化测试，例如性能测试、配置测试、大数据量输入测试等。

至于自动化测试的开发过程，与软件开发过程从本质上来讲是一样的，这将在第 6 章中介绍。

1.1.3 Selenium 简介

Selenium 是一系列基于 Web 的自动化测试工具。它提供了一系列测试函数，用于支持 Web 自动化测试。这些函数非常灵活，它们能够通过多种方式定位界面元素，并可以将预期结果与系统实际表现进行比较。

作为一款强大的测试工具，Selenium 具有以下几个特性。

- (1) 可对多浏览器进行测试，如 IE、Firefox、Safari、Chrome、Android 手机浏览器等。
- (2) 支持多种语言，例如 Java、C#、Python、Ruby、PHP 等。
- (3) 跨平台，例如 Windows、Linux、iOS、Android 等。
- (4) 开源免费。

下面举个例子来说明 Selenium 的使用，假设将要开始对京东商城的用户登录进行手动测试，测试步骤如下。

- (1) 进入京东用户登录页面 <https://passport.360buy.com/new/login.aspx>，如图 1-1 所示。



用户登录

用户名:

密码: [忘记密码?](#)

记住用户名 自动登录

图 1-1 京东用户登录页面

- (2) 输入用户名和密码，单击“登录”按钮。输入过程中分别按不输入用户名，输

入错误的用户名，不输入密码，输入错误的密码，输入正确的用户名和密码的顺序进行测试。

(3) 检查登录后的状态是否符合预期结果。

这看上去是一个很简单的操作，但下面进行一个极端的假设：假设在测试时，需要在 6 个浏览器上都测试通过（如 IE、Firefox、Safari、Chrome、Android、iPhone），每天都有 8 个项目上线，而登录功能属于冒烟测试检查点，那么登录功能也同样得检查 8 次。那么算上各个浏览器的测试，每天就得将登录执行 $6 \times 8 = 48$ 遍，而且每遍都要包含以下几种检查：不输入用户名，输入错误的用户名，不输入密码，输入错误的密码，最后输入正确的用户名和密码。一个看似简单的操作一天要执行这么多次（还没有算上其他的功能），会让测试人员感觉很烦。

如果使用 Selenium，一切就好办了，只需编写少量代码，就可以实现以下功能。

(1) 计算机自动打开浏览器并进入京东用户登录页面 <https://passport.360buy.com/new/login.aspx>。

(2) 计算机自动输入用户名和密码，并单击“登录”按钮。分别按不输入用户名，输入错误的用户名，不输入密码，输入错误的密码，输入正确的用户名和密码的顺序进行测试。

(3) 计算机自动检查登录后的提示文字是否为预期结果，然后自动输出测试报告。

(4) 计算机自动切换到另一个浏览器，重复执行(1)~(3)步，直到每个浏览器都执行完毕。

以后执行的时候只需要执行 Selenium 代码即可，无需测试人员费神，看上去是不是要轻松多了？

1. Selenium 工具组

Selenium 由以下几个工具组成，每一种工具都扮演着独特的角色。

- **Selenium IDE**。Selenium IDE 是一个用于构建脚本的初级工具。它是一个 Firefox 插件，拥有一个易于使用的界面。Selenium IDE 拥有录制功能，能够记录用户执行的操作，并将其导出为可重复使用的脚本（支持多种编程语言），然后用于执行测试。

- **Selenium 1**。Selenium 1 (Selenium-RC) 是 Selenium 最主要的测试工具之一，它所具有的某些功能即使是新版的 Selenium 2 也无法支持。它能够通过多种语言（Java、JavaScript、Ruby、PHP、Python、Perl 和 C#）编写测试代码，同时能支持几乎所有浏览器的测试。

- **Selenium 2。** Selenium 2 (WebDriver) 作为最新版的 Selenium 工具，代表未来 Selenium 的发展方向。这套全新的自动化测试工具提供了许多功能，包括一套组织性更好、面向对象的 API，并克服了在之前 Selenium 1 版本中测试的局限性。

可以通过很少的修改就将 Selenium 1 的代码移植到 Selenium 2。同时，Selenium 2 也提供了向前兼容 Selenium 1 的接口。

- **Selenium Grid。** Selenium Grid 能够让 Selenium 1 的测试在多个不同的环境中运行，也能让测试并行执行。也就是说，各个测试能够在同一时间、不同机器上运行。这有两个好处。首先，如果拥有一套大规模的测试或执行缓慢的测试，可以通过 Selenium Grid 将测试在同一时间、不同机器上运行，从而大幅提高性能；其次，如果测试必须在多个环境中运行，那么 Selenium Grid 具有的“同一时间、不同机器”的特性也能够轻松做到这一点。不管怎么样，Selenium Grid 都能够大幅提高测试的效率。

2. 选择合适的 Selenium 工具

大多数用户都是从 Selenium IDE 开始的。如果没有编程经验，可以通过 Selenium IDE 来快速熟悉 Selenium 的命令。使用 IDE，可以快速创建简单的测试，有时甚至只需花几分钟的时间。

然而，并不建议所有的自动化测试都使用 Selenium IDE。为了有效地使用 Selenium，需要使用 Selenium 1 或 Selenium 2，并配合使用其中一种编程语言，自己创建并运行测试。

虽然，Selenium 2 是 Selenium 未来的发展方向，但 Selenium 1 和 Selenium 2 各有优劣，这需要用户进行判断。另外，Selenium 1 和 Selenium 2 可以互相转换。

Selenium Grid 一般用于分布式测试和集群测试，需要在多台机器同时执行测试时，可以选择使用该工具。

1.2 HTML/XML/XPath 基础

1.2.1 HTML 简介

Selenium 毕竟是 Web 测试工具，在编写 Selenium 测试时，大部分时间都要与 HTML 打交道，因此，能读懂 HTML 对于使用 Selenium 测试来说至关重要。