

## 02 Selenium环境搭建

# 本章大纲

2.1 预备知识和技能

2.2 前端工具介绍

2.4 Selenium环境配置步骤

2.5 编写第一个自动化脚本

# 预备知识和技能

- 软件测试理论
- HTML
- CSS
- Xpath
- HTML DOM

# 本章大纲

2.1 预备知识和技能

2.2 前端工具介绍

2.4 Selenium环境配置步骤

2.5 编写第一个自动化脚本

# 前端工具介绍

- 前端工具

- Google Chrome

- View → Developer → Developer Tools

- Firefox-Firebug组件

- FireBug安装方式：首先在FireFox浏览器的菜单栏中单击tools（工具）—>add-ons Manager（添加组件），搜索FireBug。然后对搜索到的插件进行安装，安装完成后重启浏览器，即可在工具栏看到FireBug的按钮。

- Internet Explorer

- 工具 → 开发人员工具

# 本章大纲

2.1 预备知识和技能

2.2 前端工具介绍

2.4 Selenium环境配置步骤

2.5 编写第一个自动化脚本

# 环境配置的步骤

1. 浏览器Firefox的安装
2. JDK8的安装
3. Eclipse的安装
4. Selenium Server的安装
5. Selenium IDE的安装
6. FireBug的安装
7. Firepath的安装
8. 总结安装xpi文件的方法

# Firefox的安装

- 下载对应版本版Firefox

<http://firefox.com.cn/download/>

- 下载成功后默认安装即可



# JDK的安装

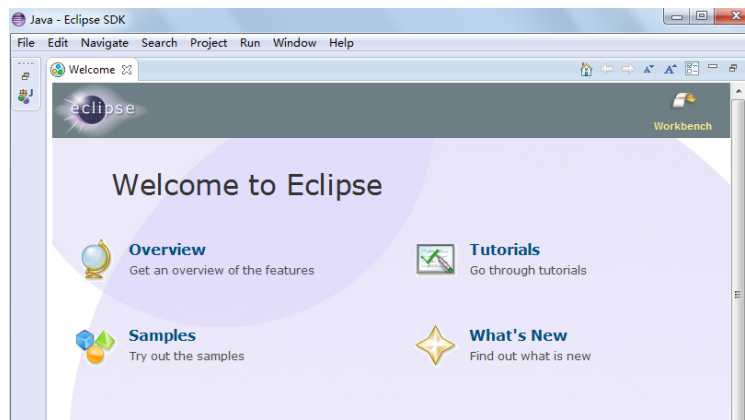
- JDK (Java Development Kit) 是Sun Microsystems针对Java开发员的产品。自从Java推出以来, JDK已经成为使用最广泛的Java SDK。JDK 是整个Java的核心, 包括了Java运行环境、Java工具和Java基础类库
- 下载JDK1.8, 下载地址为  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- 点击页面中的 DownLoad 下载JDK的最新版本, 下载成功后默认安装即可, 安装成功后, 在cmd窗口中输入:  
`java -version` 会查看到已经安装的java版本信息

# Eclipse的安装

- Eclipse 是一个开放源代码的、基于Java的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境  
下载地址为：

<https://www.eclipse.org/downloads/eclipse-packages/>

- 下载的包为eclipse-jee-neon-R-win32-x86\_64.zip，解压后点击eclipse.exe(首次使用会弹出如下页面)，关于Eclipse的基础操作介绍请看相关演示（包括，建立工程，创建包，创建类，代码的执行）



# Selenium Server的安装

<http://docs.seleniumhq.org/download/>

## Selenium Standalone Server

---

The Selenium Server is needed in order to run Remote Selenium WebDriver. Selenium 3.X is no longer capable of running Selenium RC directly, rather it does it through emulation and the WebDriverBackedSelenium interface.

Download version 3.6.0

selenium-server-standalone-3.6.0.jar

# Selenium IDE的安装

## Selenium IDE

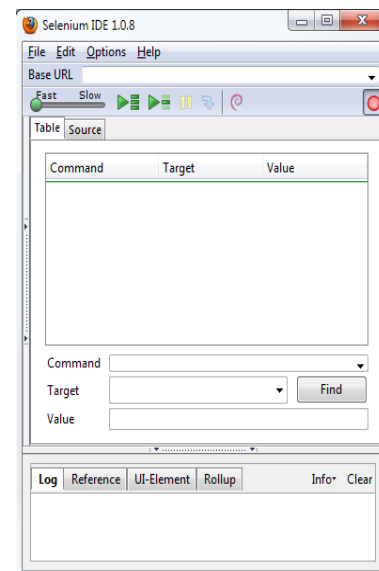
Selenium IDE is a Firefox plugin which records and plays back user interactions with the browser. Use this to either create simple scripts or assist in exploratory testing. It can also export Remote Control or WebDriver scripts, though they tend to be somewhat brittle and should be overhauled into some sort of Page Object-y structure for any kind of resiliency.

Download latest released version [from addons.mozilla.org](https://addons.mozilla.org) or view the [Release Notes](#) and then [install some plugins](#).

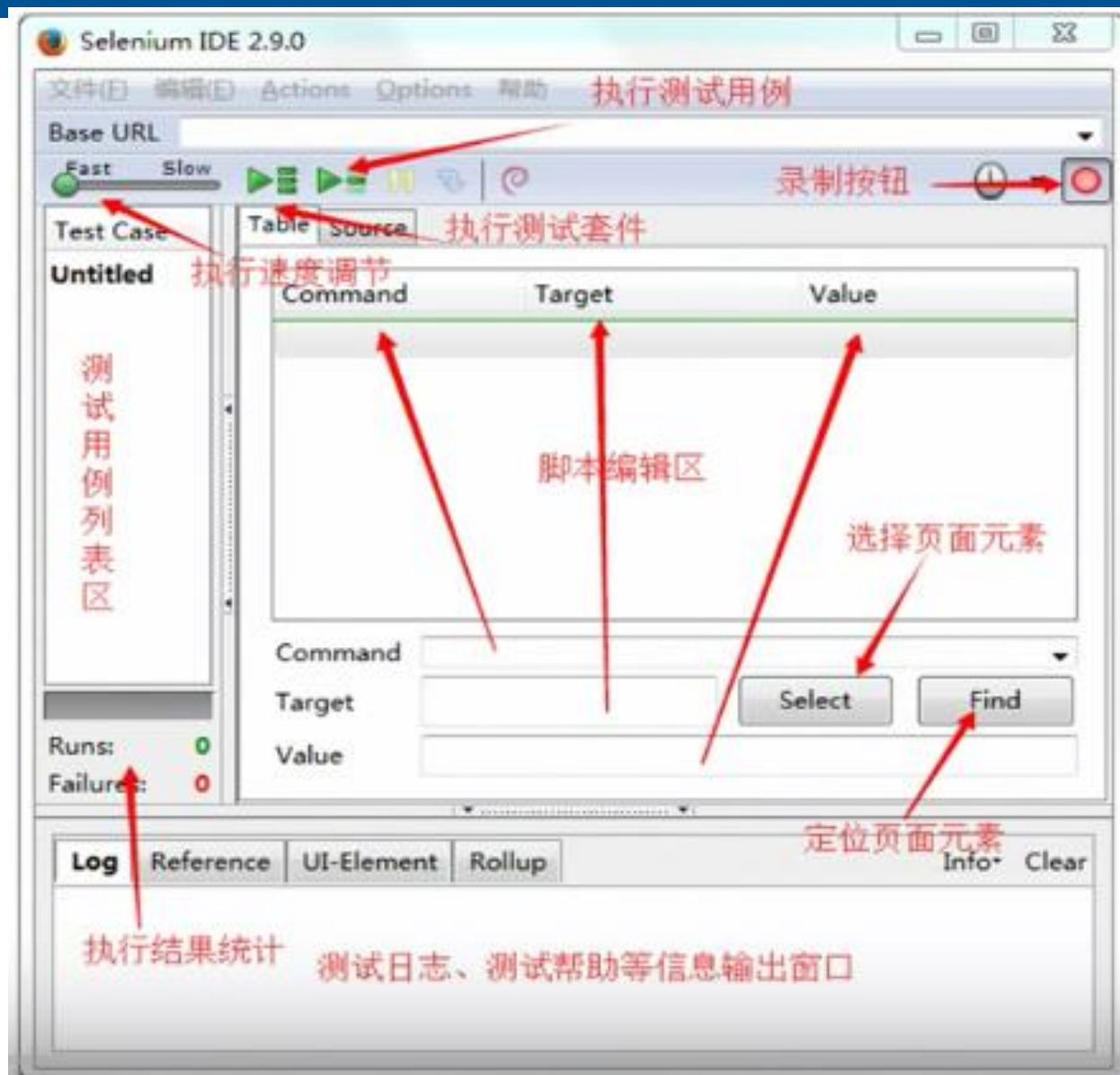
Download [previous versions here](#).

## Selenium IDE 2.9.1.1

- 如果是通过Firefox下载IDE，下载成功后会提示是否安装IDE，点击安装后，重启FireFox，
- 弹出窗口如下：

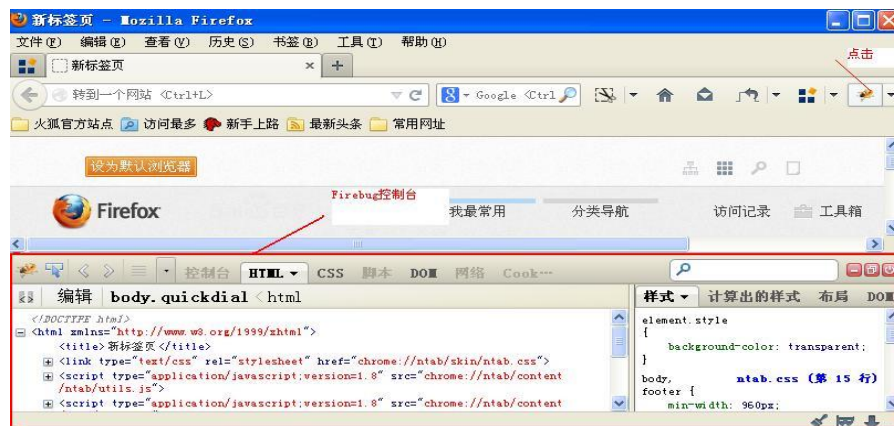


# Selenium IDE界面功能



# FireBug安装

- 首先在Firefox浏览器的菜单栏中单击tools（工具）—>add-ons Manager（添加组件），搜索FireBug。然后对搜索到的插件进行安装，安装完成后重启浏览器，即可在工具栏看到FireBug的按钮。



# Firepath的安装

- Firepath: 安装FirePath以后, firebug窗口里面有FirePath Tab, 进而查询网页中xpath的信息, 如下图
- 安装步骤: 请参考安装FireBug, 在Firefox的组件搜索中输入“Firepath”进行安装



# 总结安装xpi文件的方法

- 1.直接拖拽到Firefox浏览器的窗口里或扩展窗口里
- 2.点击'菜单"--"工具"--"扩展",把.xpi文件拖进弹出窗口里,稍后就可以看到Firefox会询问你是否要安装这个插件,点击是,并重启Firefox浏览器
- 3.鼠标右键点xpi文件,选打开方式,在打开方式里选firefox,设置关联。以后只要双击xpi文件就自动安装了



# 本章大纲

2.1 预备知识和技能

2.2 前端工具介绍








2.4 Selenium环境配置步骤

2.5 编写第一个自动化脚本

# 自动化用例的步骤

1. 声明driver对象（将要启动什么浏览器）
2. driver去打开浏览器并输入你要测试的网页地址（使用get方法打开测试站点）
3. 找到需要操作元素（利用WebElement声明元素对象）
4. 对元素进行输入、点击、断言操作
5. 关闭浏览器，释放资源

- <https://github.com/mozilla/geckodriver/releases>

Downloads		
 <a href="#">geckodriver-v0.19.0-arm7hf.tar.gz</a>		2.15 MB
 <a href="#">geckodriver-v0.19.0-linux32.tar.gz</a>		2.19 MB
 <a href="#">geckodriver-v0.19.0-linux64.tar.gz</a>		2.15 MB
 <a href="#">geckodriver-v0.19.0-macos.tar.gz</a>		1.28 MB
 <a href="#">geckodriver-v0.19.0-win64.zip</a>		2.08 MB
 <a href="#">Source code (zip)</a>		
 <a href="#">Source code (tar.gz)</a>		

# 编写第一个自动化脚本

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
public class BaiduSearch {
    public static void main(String[] args) {
        System.setProperty("webdriver.gecko.driver","D:\\demo\\geckodriver
        WebDriver driver = new FirefoxDriver();
        String baseUrl = "https://www.baidu.com";
        driver.get(baseUrl);
        System.out.println(driver.getTitle());
        driver.quit();
    }
}
```

# 注意

如果启动ff浏览器失败:

可以使用System的setProperty()方法指定浏览器的路径, 用法如下

```
System.setProperty("webdriver.firefox.bin", "D:\\Program  
Files (x86)\\Mozilla Firefox\\firefox.exe");
```

# 总结

在不同的编程语言中会有语法的差异，我们抛去语法的差异性，在不同的语言中实现百度搜索的自动化实例都完成了下面几个操作。

- (1) 首先导入Selenium (webdriver) 相关模块。
- (2) 调用Selenium的浏览器驱动，获取浏览器句柄 (driver) 并启动浏览器。
- (3) 通过句柄访问百度URL。
- (4) 通过句柄操作页面元素 (百度输入框和按钮)。
- (5) 通过句柄关闭浏览器。