**Webdriver概述**

Webdriver (Selenium2）是一种用于Web应用程序的自动测试工具，它提供了一套友好的API，与Selenium 1（Selenium-RC）相比，Webdriver 的API更容易理解和使用，其可读性和可维护性也大大提高。Webdriver完全就是一套类库，不依赖于任何测试框架，除了必要的浏览器驱动，不需要启动其他进程或安装其他程序，也不必像Selenium 1那样需要先启动服务。

**支持浏览器**

* Firefox （FirefoxDriver）
* IE（InternetExplorerDriver）
* Opera（OperaDriver）
* Chrome （ChromeDriver）
* safari（SafariDriver）

**支持语言**

* Java
* C#
* PHP
* Python
* Perl
* Ruby

**安装，卸载、查看**

安装：pip install selenium==XXXX

卸载 pip uninstall selenium

查看版本号：

pip show selenium

**多浏览器运行**

**启动Firefox**

* 1.FireFox 48以上版本
* Selenium 3.X +FireFox驱动——geckodriver
* 2.Firefox 48 以下版本
* Selenium2.X 内置驱动
* 驱动下载地址<https://github.com/mozilla/geckodriver/releases>

**启动IE浏览器**

* IE 9以上版本
* Selenium3.X +IE驱动
* IE 9以下版本
* Selenium 2.X +IE驱动

**启动Chrome浏览器**

selenium2.x/3.x +Chrome驱动

**注意！**

各个驱动下载地址： <http://www.seleniumhq.org/download/>

浏览器位数的版本和驱动版本要一致！ 如果是32bit浏览器而Driver是64bit则会导致脚本运行失败！

第一个自动化测试脚本

案例：

* 启动火狐浏览器，
* 首先打开我要自学网页面，打印网页标题，等待3秒
* 打开百度首页，打印网页标题，再等待2秒
* 关闭浏览器。

from selenium import webdriver

from time import sleep

#加载浏览器驱动

driver=webdriver.**Firefox**()

#打开自学网页面

driver.**get**("http://www.51zxw.net")

**print**(driver.title)

**sleep**(3)

#打开百度首页

driver.**get**("http://www.baidu.com")

**print**(driver.title)

**sleep**(3)

#关闭浏览器

driver.**quit**()

### 浏览器操作

* 浏览器窗口大小设置
* 页面前进后退
* 页面刷新

from selenium import webdriver

from time import sleep

driver=webdriver.**Firefox**()

driver.**get**("http://www.51zxw.net")

driver.**maximize\_window**()

**sleep**(2)

driver.**get**("http://www.51zxw.net/list.aspx?cid=615")

driver.**set\_window\_size**(400,800)

driver.**refresh**()

**sleep**(2)

driver.**back**()

**sleep**(2)

driver.**forward**()

**sleep**(2)

driver.**quit**()

**元素定位**

元素的定位应该是自动化测试的核心，要想操作一个元素，首先应该识别这个元素。

webdriver提供了一系列的元素定位方法，常用的有以下几种

* id
* name
* class name
* link text
* partial link text
* tag name
* xpath
* css selector

案例：打开百度首页，在搜索框自动输入“Selenium我要自学网”关键词，然后点击搜索按钮，查看搜索页面。

### id与name 定位

from selenium import webdriver

from time import sleep

driver=webdriver.**Firefox**()

driver.**get**("http://www.baidu.com")

driver.**find\_element\_by\_id**("kw").**send\_keys**("Selenium我要自学网")

driver.**find\_element\_by\_name**("wd").**send\_keys**("Selenium我要自学网")

**sleep**(2)

driver.**find\_element\_by\_id**("su").**click**()

### tag\_name定位

案例：打开我要自学网页面，在用户名输入框输入用户名“selenium”

from selenium import webdriver

from time import sleep

driver=webdriver.**Firefox**()

driver.**get**("http://www.51zxw.com")

#定位标签名为input的元素

driver.**find\_element\_by\_tag\_name**("input").**send\_keys**("selenium")

#获取页面所有标签名称为“input”的标签。

driver.**find\_elements\_by\_tag\_name**("input")[0].**send\_keys**("selenium")

**sleep**(3)

driver.**quit**()

### class\_name定位

根据标签中属性class来进行定位的一种方法

from selenium import webdriver

from time import sleep

driver=webdriver.**Firefox**()

driver.**get**("http://www.baidu.com")

driver.**find\_element\_by\_class\_name**("s\_ipt").**send\_keys**("Selenium 我要自学网")

**sleep**(2)

driver.**find\_element\_by\_id**("su").**click**()

**sleep**(3)

driver.**quit**()

### link\_text定位

link\_text定位就是根据超链接文字进行定位。

**from** selenium import webdriver

**from** time import sleep

driver=webdriver.Firefox()

driver.get("http://www.51zxw.net/")

driver.find\_element\_by\_link\_text('程序开发').click()

sleep(3)

driver.find\_element\_by\_partial\_link\_text('神秘面纱').click()

### XPath定位

XPath即为XML路径语言，它是一种用来确定XML文档中某部分位置的语言。XPath基于XML的树状结构，提供在数据结构树中找寻节点的能力。

##### xpath绝对与相对定位

from selenium import webdriver

from **time** import **sleep**

driver=webdriver.Firefox()

driver.get("http://www.baidu.com")

*# 绝对路径定位*

driver.find\_element\_by\_xpath("/html/body/div[2]/div[1]/div/div[1]/div/form/span[1]/input").send\_keys("51zxw")

*# 利用元素熟悉定位--定位到input标签中为kw的元素*

driver.find\_element\_by\_xpath("//input[@id='kw']").send\_keys("Selenium")

*# 定位input标签中name属性为wd的元素*

driver.find\_element\_by\_xpath("//input[@name='wd']").send\_keys("Selenium")

*# 定位所有标签元素中，class属性为s\_ipt的元素*

driver.find\_element\_by\_xpath("//\*[@class='s\_ipt']").send\_keys("Python3")

driver.find\_element\_by\_id('su').click()

**sleep**(3)

driver.quit()

#### Xpath层级与逻辑定位

from selenium import webdriver

from time import sleep

driver=webdriver.**Firefox**()

driver.**get**("http://www.51zxw.net/")

#层级和属性结合定位--自学网首页输入用户和名密码

driver.**find\_element\_by\_xpath**("//form[@id='loginForm']/ul/input[1]").**send\_keys**("51zxw")

driver.**find\_element\_by\_xpath**("//form[@id='loginForm']/ul/input[2]").**send\_keys**("66666")

#逻辑运算组合定位

driver.**find\_element\_by\_xpath**("//input[@class='loinp' and @name='username']").**send\_keys**("51zxw")

**sleep**(3)

driver.**quit**()

**Css定位**

Selenium极力推荐使用CSS 定位，而不是XPath来定位元素，原因是CSS 定位比XPath 定速度快，语法也更加简洁。

**CSS常用定位方法**

1. find\_element\_by\_css\_selector（）
2. #id id选择器根据id属性来定位元素
3. .class class选择器，根据class属性值来定位元素
4. [attribute='value'] 根据属性来定位元素
5. element>element 根据元素层级来定位 父元素>子元素

from selenium import webdriver

from time import sleep

driver=webdriver.**Firefox**()

driver.**get**("http://www.baidu.com")

#根据id来定位

driver.**find\_element\_by\_css\_selector**('#kw').**send\_keys**("Selenium 我要自学网")

#根据class定位

driver.**find\_element\_by\_css\_selector**('.s\_ipt').**send\_keys**('python')

#通过属性来定位

driver.**find\_element\_by\_css\_selector**("[autocomplete='off']").**send\_keys**("selenium")

**sleep**(2)

driver.**find\_element\_by\_id**('su').**click**()

driver.**get**("http://www.51zxw.net")

#通过元素层级来定位

driver.**find\_element\_by\_css\_selector**("form#loginForm>ul>input").**send\_keys**("51zxw")

**sleep**(2)

driver.**quit**()