Selenium 中文 API

最近研究了下 Selenium，苦于网上中文资料太少，便自己翻译了下 Selenium 官网上的 API，便于大家一起沟通和学习。

由于本人英文水平有限，部分字词句的拿捏可能不太到位，希望各位朋友给出宝贵意见哈

概念

Selenium 通过命令进行驱动。Selenium 可归纳为三种“风格”：**动作**、**辅助**和**断言**。每一个命令调用就是下表中的一行。

命令	目标	值

动作(Actions)命令一般用于操作应用程序的状态。它们通过如”点击链接”和”选择选项”的方式进行工作。如果一个动作执行失败，或是有错误，当前的测试将会停止执行。

许多动作可以被包含后缀”并等待”的方式进行调用，例如,”点击并等待”。这个后缀告知 Selenium，该命令将使浏览器向服务器产生一个请求，并使得 Selenium 等待加载一个新的页面。

辅助(Accessors)用于检查应用程序的状态并将结果存储在变量中。例如”storeTitle”。它们同样可用于自动生成断言。

断言(Assertions)类似于辅助，但它们可以验证应用程序的状态是否同所期望的相一致。例如包括”确认页面标题为 X”和”验证该复选框是否被勾选”。

所有的 Selenium 断言可以被用于三种模式：”assert”, “verify”，和”waitfor”。例如，你可以”assertText”，”verifyText”，及”waitforText”。当”assert”失败时，该测试将终止。当”verify”失败时，该测试将继续执行，并将错误记入日志。这就允许了通过单条”assert”确保应用程序在正确的页面上，而通过一系列的”verify”断言测试表单上的区域值，标签等。

“waitfor”命令用于等待某些条件变为真(可用于 Ajax 应用程序的测试)。如果该条件已经为真，他们将立即成功执行。反之，如果该条件不为真，则将失败并暂停测试，直到超过当前所设定的超时时间(参照后面的 setTimeout 动作)。

元素定位器(Element Locators)告诉 Selenium 是向 HTML 中的哪一个元素发送命令。许多命令需要一个如”target”属性的元素定位器。这其中包括”elementId”和”document. forms[0].element”。在接下来的部分将更详细的描述它们。

式样(Patterns)由于多种因素被使用着，如指定一个输入域的期望值，或识别一个选择选项。Selenium 支持许多类型的式样，其中包括正则表达式，所有这些将在接下来的章节中进行更详细的描述。

定义一个类用于运行 Selenium 命令。

元素定位器(Element Locators)

元素定位器(Element Locators)告诉 Selenium 是向 HTML 中的哪一个元素发送命令。一个定位器的格式如下：

locatorType = argument

我们支持如下写法用于定位元素：

- identifier=*id* :根据指定的@id 属性选择元素。如果没有匹配的值，则选择第一个@name 属性为 *id* 的元素。(参照后面)
- id=*id* :根据指定的@id 属性选择元素。
- name=*name* :选择第一个根据指定的@name 所查找到的元素。
  - username
  - name=username

这里的 name 可以作为可选项跟在一个或多个元素过滤器的后面，通过空格进行分隔。如果没有指定过滤类型，则假定为 value。

  - name=flavour value=chocolate

- dom=*javascriptExpression*: 通过检测指定字符串查找元素。这使得你可以通过 JavaScript 贯穿 HTML 文档对象。注意在这个字符串中返回值不是必须的；仅仅只需要确保这条语句是你块中的最后一条。
  - dom=document.forms['myForm'].myDropdown
  - dom=document.images[56]
  - dom=function foo() { return document.links[1];}; foo();
- xpath=*xpathExpression*: 通过 XPath 表达式定位元素。
  - xpath=//img[@alt='The image alt text']
  - xpath=//table[@id='table1']//tr[4]/td[2]
  - xpath=//a[contains(@href, '#id1')]
  - xpath=//a[contains(@href, '#id1')]/@class
  - xpath=(//table[@class='stylee'])//th[text()='theHeaderText']/../td
  - xpath=//input[@name='name2' and @value='yes']
  - xpath=//*[text()="right"]
- link=*textPattern*: 选择所包含的文字匹配指定式样的链接(锚)。
  - link=The link text
- css=*cssSelectorSyntax*: 通过 css 选择器选择元素。请查询 CSS2 选择器，CSS3 选择器以获得更多信息。在下载下来的 selenium core package 中的 selenium test suite 里的 TestCssLocators test ，你同样可以查看到使用例子。
  - css=a[href="#id3"]

- css=span#firstChid + span

当前 css 选择过滤器支持所有的 css1,css2,css3,除了 css3 中一些虚拟类(:nth-of-type,:nth-last-of-type,:first-of-type,:last-of-type,:only-of-type,:visited,:hover,:active,:focus,:indeterminate)以及虚拟元素(::first-line,::first-letter,::selection,::before,::after)。

如果没有一个显式的前缀，Selenium 使用以下默认写法：

- dom,用于开头为"document."的定位器
- xpath,用于开头为"//"的定位器
- identifier，其他

元素过滤器(Element Filters)

元素过滤器可以同选择器一起使用，从一堆候选元素中进行筛选。它们当前仅使用于'name'元素选择器。

过滤器看起来更像是选择器，也就是：

filterType=argument

所支持的元素过滤器为：

value=*valuePattern*

匹配元素时基于它们的值进行匹配。这在对一堆相似命名的关联按钮的筛选中显得尤其有用。

index=*index*

选择单个元素基于其在列表中的位置(从 0 开始)。

字符串匹配 式样

有各种各样的式样语法可用于匹配字符串值：

- glob:*pattern*: 用"glob"去匹配一个字符串。"Glob"是一种用于命令行 shells 的代表性的有限正则表达式语法。在一个 glob 式样中,"*"代表任意序列字符集,而"?"则代表任意单个字符。Glob 式样匹配整个字符串。
- regexp:*regexp*: 使用正则表达式匹配字符串。可使用所有的 JavaScript 正则表达式。

如果没有指定式样前缀，Selenium 假定其为"glob"式样。

Selenium Actions

addLocationStrategy (strategyName, functionDefinition)

为 selenium 定义一个新的函数用于定位页面上的元素。例如，如果你定义了一个方法"foo",并运行了 click("foo=blah")，我们将运行你的函数，传递给你字

符串"blah"，并点击该函数所返回的元素，如果返回为 null，则抛出一个"Element not found"的错误。我们将给该函数传递三个参数。

- locator: 用户传递过来的字符串
- inWindow: 当前所选中的窗体
- inDocument: 当前所选中的文档

如果未找到相应的元素，则函数必须返回一个 null。

参数：

- strategyName - 定义的方法名；只能使用字母[a-zA-Z],不能包含空格或其他标点符号。
- functionDefinition - 在 JavaScript 函数中的一个定义 body 的字符串。如：return inDocument.getElementById(locator);

addSelection (locator, optioLocator)

为通过使用选择定位器，在一个可多选元素中所选择的集合添加一个 selection。@查看#doSelect 关于选择定位器的细节。

参数：

- locator - 用于指定一个多选框的元素定位器
- optionLocator - 一个选择定位器(默认为标签)

allowNativeXpath (allow)

指定 Selenium 是否使用 XPath 的本地浏览执行(如果有可用的本地版本);如果传递的值为"false"，我们将使用 pure-JavaScript xpath 库。使用 pure-JS xpath 库可以提高 xpath 元素定位器在不同浏览器中的一致性，但其执行速度将大大低于本地执行。

参数：

- allow - Boolean,true 意味着我们更愿意使用本地 XPath;false 则意味着我们将只使用 JS XPath

altKeyDown()

按下 alt 键并保持其按下状态，直到 doAltUp()被调用或一个新的页面被加载。

altKeyUp()

释放 alt 键

answerOnNextPrompt (answer)

通知 Selenium 返回下一次 JavaScript prompt[window.prompt()]所指定的回答字符串。

参数：

- answer - 对弹出的提示所给与的回答

assignId (locator,identifier)

临时为指定元素设定一个"id"属性，使你可以在将来使用其 ID，以代替缓慢且更复杂的 XPath。该 ID 将在页面重载后消失。

参数：
- locator - 指向某个元素的元素定位器
- identifier - 为指定元素作为 ID 使用的字符串

break()
暂停当前正在进行的测试，并等待用户按下继续按钮。这个命令对于调试非常有用，但使用时要特别小心，因为他将强制暂停自动化测试，直到用户手动操作。

check(locator)
勾选一个关联性按钮(checkbox/radio)
参数：
- locator - 一个元素定位器

chooseCancelOnNextConfirmation()
默认情况下， Selenium 的重载 window.confirm()函数将返回 true，等同于用户手动点击 OK;执行该命令后，下一次调用 confirm()将返回 false，等同于用户手动点击了 Cancel。Selenium 对后来的确认动作将继续使用默认行为，自动返回 true(OK),除非/直到你为每个确认动作明确的调用此命令。

chooseOkOnNextConfirmation()
撤销调用 chooseCancelOnNextConfirmation 的效果。注意，Selenium 的重载 window.confirm()函数通常将自动返回 true，等同于用户手动点击 OK，因此你没有必要使用此命令，除非由于某种原因使你在下一次确认动作前不得不改变你先前的想法。在任意确认动作后，Selenium 对后来的确认动作将继续使用默认行为，自动返回 true(OK),除非/直到你为每个确认动作明确的调用 chooseCancelOnNextConfirmation()。

click(locator)
点击一个链接、按钮、多选框或单选框。如果该点击事件导致了新的页面加载(如同链接通常所作的)，将调用 waitForPageToLoad。
参数：
- locator -一个元素定位器

clickAt(locator,coordString)
点击一个链接、按钮、多选框或单选框。如果该点击事件导致了新的页面加载(如同链接通常所作的)，将调用 waitForPageToLoad。
参数：
- locator -一个元素定位器
- coordString - 指定由定位器返回的鼠标事件相关联的元素 x,y 坐标(也就是 - 10,20)

close()
模拟用户点击弹出窗体或表单标题栏上的"关闭"按钮。

controlKeyDown()
按下 control 键并保持其按下状态，直到 doControlUp()被调用或一个新的页面被加载。

controlKeyUp()
释放 control 键

createCookie(nameValuePair,optionsString)
创建一个新的 cookie，除非你清楚的指定该 cookie 的路径，否则其路径和域将与当前测试的页面相同。
参数：
- nameValuePair - 该 cookie 的名称和值，使用如下格式"name=value"
- optionsString - 该 cookie 的选项。当前支持的选项包括'path'和'max_age'。optionsString 的格式为"path=/path/,max_age=60"。选项的顺序无关紧要。

deleteCookie(name,path)
删除指定路径下的该名称 cookie。
参数：
- name - 被删除 cookie 的名称
- path - 被删除 cookie 的路径属性

doubleClick(locator)
双击一个链接、按钮、多选框或单选框。如果该双击事件导致了新的页面加载(如同链接通常所作的)，将调用 waitForPageToLoad。
参数：
- locator - 一个元素定位器

doubleClickAt(locator,coordString)
双击一个链接、按钮、多选框或单选框。如果该双击事件导致了新的页面加载(如同链接通常所作的)，将调用 waitForPageToLoad。
参数：
- locator - 一个元素定位器
- coordString - 指定由定位器返回的鼠标事件相关联的元素 x,y 坐标(也就是 - 10,20)

dragAndDrop(locator,movementsString)
拖动元素一定的距离并放下
参数：
- locator - 一个元素定位器
- movementsString – 从当前位置到指定位置的像素偏移量，如,"+70,-300"

dragAndDropToObject(locatorOfObjectToBeDragged,locatorOfDragDestinationObject)

拖动元素到另一元素

参数：

- locatorOfObjectToBeDragged – 被拖动的元素
- locatorOfDragDestinationObject – 被拖动的元素将拖向的元素的坐标 (如，其最中心像素)

Dragdrop(locator,movementsString)

不建议 – 用 dragAndDrop 代替

参数：

- locator - 一个元素定位器
- movementsString – 从当前位置到指定位置的像素偏移量，如,"+70,-300"

Echo(message)

打印指定消息到你的 Selenese 表的第三个表单元。有利于调试。

参数：

- message - 要打印的消息

fireEvent(locator,eventName)

明确地模拟一个事件，触发"onevent"响应句柄。

参数：

- locator - 一个元素定位器
- eventName – 事件名，如"focus" 或"blur"

getSpeed()

获取执行速度(也就是，获取接下来的每一个 selenium 操作的延迟豪秒长度)。默认情况下，是不会有延迟的。也就是延迟为 0 毫秒。参照 setSpeed。

goBack()

模拟用户点击其浏览器上的"back"按钮

highlight(locator)

暂时将指定元素的背景色改变为黄色。有利于调试。

参数：

- locator - 一个元素定位器

keyDown(locator,keySequence)

模拟用户按下一个键(除了还没释放的)

参数：

- locator - 一个元素定位器
- keySequence - 可以是个字符串("\"后跟随要被按下键的数字键码，通常是该键的 ASCII 值)，或是个单字符，如"w", "\119"。

keyPress(locator,keySequence)

模拟用户按下和释放一个键。

参数：

- locator - 一个元素定位器
- keySequence - 可以是个字符串(“\”后跟随要被按下键的数字键码，通常是该键的 ASCII 值)，或是个单字符，如“w“, “\119“。

keyUp(locator,keySequence)

模拟用户释放一个键。

参数：

- locator - 一个元素定位器
- keySequence - 可以是个字符串(“\”后跟随要被按下键的数字键码，通常是该键的 ASCII 值)，或是个单字符，如“w“, “\119“。

metaKeyDown()

按下 meta 键并保持其按下状态，直到 doMetaUp()被调用或一个新的页面被加载。

metaKeyUp()

释放 meta 键

mouseDown(locator)

模拟用户在指定元素上按下鼠标按钮(除了还没释放的)。

参数：

- locator - 一个元素定位器

mouseDownAt(locator,coordString)

模拟用户在指定位置上按下鼠标按钮(除了还没释放的)。

参数：

- locator - 一个元素定位器
- coordString -指定由定位器返回的鼠标事件相关联的元素 x,y 坐标(也就是 - 10,20)

mouseMove(locator)

模拟用户在指定元素上按下鼠标按钮(除了还没释放的)。

参数：

- locator - 一个元素定位器

mouseMoveAt(locator,coordString)

模拟用户在指定位置上按下鼠标按钮(除了还没释放的)。

参数：

- locator - 一个元素定位器
- coordString -指定由定位器返回的鼠标事件相关联的元素 x,y 坐标(也就是 - 10,20)

mouseOut(locator)

模拟用户从指定元素上移开鼠标指针。

参数：

- locator - 一个元素定位器

mouseOver(locator)

模拟用户鼠标滑过指定元素。

参数：

- locator - 一个元素定位器

mouseUp(locator)

模拟用户在指定元素上释放鼠标按钮时发生的事件(也就是，停止保持按钮按下)。

参数：

- locator - 一个元素定位器

mouseUpAt(locator,coordString)

模拟用户在指定元素上释放鼠标按钮时发生的事件(也就是，停止保持按钮按下)。

参数：

- locator - 一个元素定位器
- coordString -指定由定位器返回的鼠标事件相关联的元素 x,y 坐标(也就是 - 10,20)

open(url)

在测试框架中打开一个 URL，可以为相对和绝对 URLs。”open”命令将等待页面加载完成才继续进行，也就是明确的指名”并等待”后缀。注意：由于浏览器安全策略(相同来源方针)这个 URL 必须和当前运行的 HTML 在相同的域。如果你不得不在另一个域打开一个 URL，则需要用 Selenium 服务在另一个域去打开一个新的浏览器会话。

参数：

- url - 要打开的 URL,可以为空
- windowID - 要选择窗体的 JavaScript window ID

pause(waitTime)

等待指定时间(以毫秒为单位)

参数：

- waitTime - 要睡眠的时间(以毫秒为单位)

refresh()

模拟用户点击浏览器上的”Refresh”按钮。

removeAllSelections(locator)

取消所有可多选元素的选择状态。
参数：
- locator – 一个用于识别多选框的元素定位器

removeSelection(locator,optionLocator)
从用选项定位器进行筛选的多选元素的筛选集合中移除一个集合。@在#doSelect 中查看选项定位器的详细信息。
参数：
- locator – 一个用于识别多选框的元素定位器
- optionLocator – 一个选项定位器(默认为一个标签)

runScript(script)
在当前测试窗体的 body 中创建一个新的”script”标签，并在 body 中添加指定的命令文本。用这种方式执行脚本，通常可以比使用 Selenium 的”getEval”方式更简易的进行调试。要注意的是，由这种脚本标签所抛出的异常不受 Selenium 管理，因此当该脚本有可能会抛出异常时，你需要用 try/catch 块将其包含起来。
- script - 需要执行的 JavaScript 片段

select(selectLocator,optionLocator)
用选项选择器从一个下拉框中选择一个选项。
选项选择器提供不同的方法从一个 HTML 选择元素中识别选项。(例如：选择一个指定选项，或断言一个满足某种规范的选项)有许多种形式的选择选项定位器。

- label=*labelPattern*: 基于其标签匹配选项，如其有效文本。(默认)
  - label=regexp:^[Oo]ther
- value=*valuePattern*: 基于其值匹配选项。
  - value=other
- id=*id*: 基于其 id 匹配选项。
  - id=option1
- index=*index*: 基于其索引匹配选项(从 0 开始)。
  - index=2

如果没有为选项定位器提供前缀，则默认匹配为标签行为。
参数：
- selectLocator – 一个用于识别下拉菜单的元素定位器
- optionLocator – 一个选项选择器(默认为标签)

selectFrame(locator)
在当前窗体中选择一个框架(你可以多次调用这个命令用于选择嵌套框架)。要选择父框架，用”relative=parent”作为定位器；要选择顶级框架，用”relative=top”。你同样可以通过基于 0 的索引号选择框架；用”index=0”选择第一个框架，或者用”index=2”选择第三个框架。
你同样可以直接使用一个 DOM 表达式来识别你要的框架。像这样：
dom=frames[“main”].frames[“subframe”]

参数：

- locator - 一个用于识别框架或子框架的元素定位器

selectWindow(windowID)

选择一个弹出窗体；一旦一个弹出窗体被选中，所有的命令将指向该窗体。要再次选择主窗体，将对象设定为 null。

注意：window 的内在 JavaScript 的”name”属性和被给与的 window 文档(通常是你实际看到的，作为最终用户，在窗体的标题栏上)的”title”之间有一个很大的不同。”name”对于最终用户通常是不可见的；它是作为第二个参数”windowName”传递给 JavaScript 函数 window.open(url,windowName,windowFeatures,replaceFlag)(被 Selenium 截取)。

Selenium 有许多方法用于查找被”windowID”参数所提及的窗体对象。

1.) 如果 windowID 为 null，(或是字符串”null”)，则假定为用户是提交给由浏览器最初实例化的窗体。
2.) 如果”windowID”参数的值是当前应用窗体的一个 JavaScript 变量名，则假定该变量包含一个由调用 JavaScript window.open()函数所产生的返回值。
3.) 另外，selenium looks in a hash it maintains that maps string names to window “names”.
4.) 如果失败了，我们将循环遍历所有已知的窗体以便试图找出适合的”title”。由于”title”不是必须唯一，因此可能会产生一些非期望的行为。

如果很难判定你所要操作的窗体的名称，你可以查看为识别通过 window.open(被 Selenium 截取)所打开窗体的名称时所产生的 selenium 日志消息。在每个窗体被打开时，你通常可以看到如下信息：

debug: window.open call intercepted; window ID (你可以用于 selectWindow()) is “myNewWindow”

在某些情况，Selenium 会无法截取 window.open 的调用(例如，如果该调用发生在”onLoad”事件之间或之前)。(该 BUG 标记为 SEL-339)。在这些情况，你可以使用 Selenium 的 openWindow 命令强制 Selenium 去通告打开窗体的名称，使用一个空(blank) url，像这样：openWindow(“”,”myFunnyWindow”)。

参数：

- windowID – 要选择窗体的 JavaScript 窗体 ID

setBrowserLogLevel(logLevel)

设定浏览器方日志信息级别；在此级别之下的日志信息将被丢弃。有效的日志级别字符串有：”debug”，”info”，”warn”，”error”，或”off”。要查看浏览器日志，在 GUI 模式下打开日志窗口，或在 Selenium RC 中将浏览器端记入日志设定为 enable。

参数：

- logLevel – 以下之一：”debug”，”info”，”warn”，”error”，或”off”

setCursorPosition(locator,position)

将文本光标移动到被给与的输入元素或文本域的指定位置。若指定元素不是一个可输入元素或文本域，该方法将失败。
参数：
- locator – 一个指向输入元素或文本域的元素定位器
- position – 该范围的光标数字位置；position 如果设定为 0，则为该范围域的最开始位置，你同样可以将光标设定为-1 以移动到该范围域的最末端。

setMouseSpeed(pixels)
配置在 dragAndDrop 命令执行期间，”mousemove”事件时的像素数字(默认为 10)。
将这个值设定为 0，意味着我们将向从开始位置到结束位置的每一个像素发送一个”mousemove”事件；那将会非常缓慢，且可能导致某些浏览器将该 JavaScript 强制设定为超时。

如果该鼠标速度大于两个拖动对象间的距离，我们将只向开始位置和结束位置发送一个”mousemove”事件。
参数：
- pixels – 两个“mousemove”事件间的像素间隔

setSpeed(value)
设定执行速度(也就是说，设定将要执行的每条 selenium 操作间的毫秒延迟间隔长度)。默认情况下，没有延迟，也就是延迟为 0 毫秒。
参数：
- value – 在操作后的暂停毫秒数

setTimeout(timeout)
指定 Selenium 等待动作完成的等待时间。
需要等待的动作包括”open”和”waitFor*”。
默认超时为 30 秒。
参数：
- timeout – 以毫秒为单位，超过后该命令将返回错误。

shiftKeyDown()
按下 shift 键，并保持按下状态，直到 doShiftUp()被调用或一个新的页面被加载。

shiftKeyUp()
释放 shift 键。

store(expression,variableName)
该命令是存储表达式的同义词。
参数：
- expression – 要存储的值
- variableName – 用于存储结果的变量名

submit(formLocator)

提交给指定表单。这对于没有提交按钮的表单特别有用，如，简单输入的”search”表单。

参数：

- formLocator – 一个指向你要提交的表单的元素定位器

type(locator,value)

设定一个输入域的值，如同你输入进去一样。

其同样可用于单选框，多选框等。在这些情况，value 应为选项选择时的值，而不是有效文本。

参数：

- locator – 一个元素定位器
- value - 要录入的值

typeKeys(locator,value)

模拟在指定元素上的按键事件，如同是你一个键一个键敲上去一样。

比起为指定字符串的每个字符调用 keyDown,keyUp,keyPress 方法，这个函数要方便的多；其对于需要明确按键事件的动态 UI 组件(如自动完成的 combo box)同样有用。

不同于简单的”敲打”命令——将指定值直接强制赋给页面，该指令可能有，也可能没有任何效果，即时在敲打按钮通常会有效的情况下。例如，如果你在一个表单元素上使用”typeKeys”，你可能可以，也可能不可以看到看到你在该区域录入的效果。

在有些时候，你可能不得不使用简单的”type”命令去设定域的值，然后用”typeKeys”命令去发送按键事件以告知你所录入的值。

参数：

- locator – 一个元素定位器
- value - 要录入的值

uncheck(locator)

取消选中一个关联性按钮(checkbox/radio)

参数：

- locator – 一个元素定位器

waitForCondition(script,timeout)

重复执行指定 JavaScript 片段直到其值为”true”。

该片段可以有多行，但只考虑其最后一行的结果。

要注意：默认情况下，该片段会在运行者的测试窗体运行，而不是在你的应用程序窗体。要得到你的应用程序窗体，你可以使用 JavaScript 片段 selenium.browserbot.getCurrentWindow()，然后让你的 JavaScript 在那运行。

参数：

- script - 要运行的 JavaScript 片段

- timeout - 以毫秒为单位，超过后该命令将返回错误。

waitForFrameToLoad(frameAddress,timeout)

等待一个新的框架加载。

Selenium 通常会持续跟踪新页面和框架的加载状态，当其第一次注意到页面加载完成，将会设定一个”newPageLoaded”标志。

查看 waitForPageToLoad 获得更多信息。

参数：

- frameAddress – 服务端的框架地址
- timeout - 以毫秒为单位，超过后该命令将返回错误。

waitForPageToLoad(timeout)

等待一个新的页面加载。

你可以使用此命令以代替”AndWait”后缀，”clickAndWait”，”selectAndWait”，”typeAndWait”等(仅在 JS API 中有效)。

Selenium 通常会持续跟踪新页面的加载状态，当其第一次注意到页面加载完成，将会设定一个”newPageLoaded”标志。当此标志变为 false 后再运行其他 Selenium 命令。因此，如果你要等待一个页面加载完成，当一个 Selenium 命令导致一个页面加载后就需立即开始等待。

参数：

- timeout - 以毫秒为单位，超过后该命令将返回错误。

waitForPopUp(windowID,timeout)

等待一个弹出窗体出现和加载。

参数：

- windowID – 将出现窗体的 JavaScript 窗体 ID
- timeout - 以毫秒为单位，超过后该命令将返回错误。

windowFocus()

将焦点赋给当前选择窗体

windowMaximize()

重新设定当前窗体大小为全屏

Selenium Accessors

assertErrorOnNext(message)

告诉 Selenium 在下一个命令执行时期待有错误。

参数：

- message – 我们所期望的错误信息。如果出现不正确的错误信息，该命令将失败。

同断言相关联，自动生成：

- assertNotErrorOnNext(message)
- verifyErrorOnNext(message)
- verifyNotErrorOnNext(message)
- waitForErrorOnNext(message)
- waitForNotErrorOnNext(message)

assertFailureOnNext(message)

告诉 Selenium 在下一个命令执行时期待有失败。

参数：

- message – 我们所期望的失败信息。如果出现不正确的失败信息，该命令将失败。

同断言相关联，自动生成：

- assertNotFailureOnNext(message)
- verifyFailureOnNext(message)
- verifyNotFailureOnNext(message)
- waitForFailureOnNext(message)
- waitForNotFailureOnNext(message)

assertSelected(selectLocator,optionLocator)

验证从下拉框中选择的选项满足选项指定器。

注意，不赞成使用该命令；你应该使用 assertSelectedLabel, assertSelectedValue, assertSelectedIndex, 或 assertSelectedId 进行代替。

查看选择命令获取更多关于选择定位器的信息。

参数：

- selectLocator - 一个用于识别下拉菜单的元素定位器
- optionLocator – 一个选项定位器，代表性的就是一个选项标签(如”John Smith”)

同断言相关联，自动生成：

- assertNotSelected(selectLocator,optionLocator)
- verifySelected(selectLocator,optionLocator)
- verifyNotSelected(selectLocator,optionLocator)
- waitForSelected(selectLocator,optionLocator)

- waitForNotSelected(selectLocator,optionLocator)

storeAlert(variableName)

返回在之前动作所产生的 JavaScript 警告消息，如果没有警告将失败。
得到一个警告同手动点击 OK 有着相同的效果。如果产生了一个警告，而你并不去得到/验证它，那么下一个 Selenium 动作将失败。
注意：在 Selenium 中，JavaScript 警告将不会弹出一个可见的警告对话框。
注意：Selenium 不支持在页面的 onload()事件句柄中所产生的 JavaScript 警告。在这种情况下，将会生成一个可见的对话框，Selenium 将被悬停直到手动点击 OK。

Returns:

最近 JavaScript 的警告消息

同断言相关联，自动生成：

- assertAlert(pattern)
- assertNotAlert(pattern)
- verifyAlert(pattern)
- verifyNotAlert(pattern)
- waitForAlert(pattern)
- waitForNotAlert(pattern)

storeAllButtons(variableName)

返回页面上所有按钮的 ID 集。
如果被给与的按钮没有 ID，则将在结果数组中显示为""。

Returns:

页面上所有按钮的 ID 集。

同断言相关联，自动生成：

- assertAllButtons(pattern)
- assertNotAllButtons(pattern)
- verifyAllButtons(pattern)
- verifyNotAllButtons(pattern)
- waitForAllButtons(pattern)
- waitForNotAllButtons(pattern)

storeAllFields(variableName)

返回页面上所有可输入域的 ID 集。
如果被给与的域没有 ID，则将在结果数组中显示为""

Returns:

页面上所有域的 ID 集。

同断言相关联，自动生成：

- assertAllFields(pattern)
- assertNotAllFields (pattern)
- verifyAllFields (pattern)
- verifyNotAllFields (pattern)
- waitForAllFields (pattern)
- waitForNotAllFields (pattern)

storeAllLinks(variableName)

返回页面上所有链接的 ID 集。

如果被给与的链接没有 ID，则将在结果数组中显示为""

Returns:

页面上所有链接的 ID 集。

同断言相关联，自动生成：

- assertAllLinks(pattern)
- assertNotAllLinks (pattern)
- verifyAllLinks (pattern)
- verifyNotAllLinks (pattern)
- waitForAllLinks (pattern)
- waitForNotAllLinks (pattern)

storeAllWindowIds(variableName)

返回所有浏览器已知的窗体 ID 集。

Returns:

所有浏览器已知的窗体 ID 集。

同断言相关联，自动生成：

- assertAllWindowIds (pattern)
- assertNotAllWindowIds (pattern)
- verifyAllWindowIds (pattern)
- verifyNotAllWindowIds (pattern)
- waitForAllWindowIds (pattern)
- waitForNotAllWindowIds (pattern)

storeAllWindowNames(variableName)

返回所有浏览器已知的窗体名称集。

Returns:

所有浏览器已知的窗体名称集。

同断言相关联，自动生成：

- assertAllWindowNames (pattern)
- assertNotAllWindowNames (pattern)

- verifyAllWindowNames (pattern)
- verifyNotAllWindowNames (pattern)
- waitForAllWindowNames (pattern)
- waitForNotAllWindowNames (pattern)

storeAllWindowTitles(variableName)

返回所有浏览器已知的窗体标题集。

Returns:

所有浏览器已知的窗体标题集。

同断言相关联，自动生成：

- assertAllWindowTitles (pattern)
- assertNotAllWindowTitles (pattern)
- verifyAllWindowTitles (pattern)
- verifyNotAllWindowTitles (pattern)
- waitForAllWindowTitles (pattern)
- waitForNotAllWindowTitles (pattern)

storeAttribute(attributeLocator,variableName)

获得一个元素属性值。

参数：

- attributeLocator – 由@符号开头，后跟随属性名，如”foo@bar”
- variableName – 用于存储结果的变量名。

Returns:

指定属性的值

同断言相关联，自动生成：

- assertAttribute (attributeLocator, pattern)
- assertNotAttribute (attributeLocator, pattern)
- verifyAttribute (attributeLocator, pattern)
- verifyNotAttribute (attributeLocator, pattern)
- waitForAttribute (attributeLocator, pattern)
- waitForNotAttribute (attributeLocator, pattern)

storeAttributeFromAllWindows(attributeName,variableName)

返回所有已知窗体的某些属性的每一个实例。

参数：

- attributeName – 窗体某属性的名称
- variableName - 用于存储结果的变量名。

Returns:

从所有已知窗体获得的该属性的数值集。

同断言相关联，自动生成：

- assertAttributeFromAllWindows (attributeName, pattern)

- assertNotAttributeFromAllWindows (attributeName, pattern)
- verifyAttributeFromAllWindows (attributeName, pattern)
- verifyNotAttributeFromAllWindows (attributeName, pattern)
- waitForAttributeFromAllWindows (attributeName, pattern)
- waitForNotAttributeFromAllWindows (attributeName, pattern)

storeBodyText(variableName)

获取页面上所有文本。

Returns:

页面上所有文本

同断言相关联，自动生成：

- assertBodyText (pattern)
- assertNotBodyText (pattern)
- verifyBodyText (pattern)
- verifyNotBodyText (pattern)
- waitForBodyText (pattern)
- waitForNotBodyText (pattern)

storeConfirmation(variableName)

返回在之前动作所产生的 JavaScript 确认消息。
默认情况下，confirm 函数将返回 true，同手动点击 OK 有着相同的效果。这可以通过之前执行 chooseCancelOnNextConfirmation 命令改变。如果产生了一个确认，而你并不去得到/验证它，那么下一个 Selenium 动作将失败。
注意：在 Selenium 中，JavaScript 确认将不会弹出一个可见的对话框。
注意：Selenium 不支持在页面的 onload()事件句柄中所产生的 JavaScript 警告。在这种情况下，将会生成一个可见的对话框，Selenium 将被悬停直到手动点击 OK。

Returns:

最近 JavaScript 的确认消息

同断言相关联，自动生成：

- assertConfirmation (pattern)
- assertNotConfirmation (pattern)
- verifyConfirmation (pattern)
- verifyNotConfirmation (pattern)
- waitForConfirmation (pattern)
- waitForNotConfirmation (pattern)

storeCookie(variableName)

返回当前测试下当前页面的所有 cookies

Returns:

当前测试下当前页面的所有 cookies

同断言相关联，自动生成：

- assertCookie (pattern)
- assertNotCookie (pattern)
- verifyCookie (pattern)
- verifyNotCookie (pattern)
- waitForCookie (pattern)
- waitForNotCookie (pattern)

storeCursorPosition(locator,variableName)

返回所给与的输入元素或文本域的文本光标位置。
要注意，这并不在所有的浏览器中有效。
特别指出，如果光标/选择已经被 JavaScript 所清除，该命令将尝试返回光标所在的最后位置，即使光标已经不在该页面。这被归档为 SEL-243。

如果指定元素不是一个可输入元素或文本域，或没有光标在此元素上，该方法将失败。
参数：

- locator – 一个指向输入元素或文本域的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

在该域中的光标数字位置

同断言相关联，自动生成：

- assertCursorPosition (locator,pattern)
- assertNotCursorPosition (locator,pattern)
- verifyCursorPosition (locator,pattern)
- verifyNotCursorPosition (locator,pattern)
- waitForCursorPosition (locator,pattern)
- waitForNotCursorPosition (locator,pattern)

storeElementHeight(locator,variableName)

返回元素的高度
参数：

- locator – 一个指向元素的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

元素的高度

同断言相关联，自动生成：

- assertElementHeight (locator,pattern)
- assertNotElementHeight (locator,pattern)
- verifyElementHeight (locator,pattern)
- verifyNotElementHeight (locator,pattern)

- waitForElementHeight (locator,pattern)
- waitForNotElementHeight (locator,pattern)

storeElementIndex(locator,variableName)

获取元素相对于其父元素的索引(从 0 开始)。注释节点和空文本节点将被忽略。

参数:

- locator – 一个指向元素的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

元素相对于其父元素的索引(从 0 开始)

同断言相关联，自动生成:

- assertElementIndex (locator,pattern)
- assertNotElementIndex (locator,pattern)
- verifyElementIndex (locator,pattern)
- verifyNotElementIndex (locator,pattern)
- waitForElementIndex (locator,pattern)
- waitForNotElementIndex (locator,pattern)

storeElementPositionLeft(locator,variableName)

返回元素的水平位置

参数:

- locator – 一个指向元素的元素定位器或元素本身
- variableName - 用于存储结果的变量名。

Returns:

到框架边缘的像素。

同断言相关联，自动生成:

- assertElementPositionLeft (locator,pattern)
- assertNotElementPositionLeft (locator,pattern)
- verifyElementPositionLeft (locator,pattern)
- verifyNotElementPositionLeft (locator,pattern)
- waitForElementPositionLeft (locator,pattern)
- waitForNotElementPositionLeft (locator,pattern)

storeElementPositionTop(locator,variableName)

返回元素的纵向位置

参数:

- locator – 一个指向元素的元素定位器或元素本身
- variableName - 用于存储结果的变量名。

Returns:

到框架边缘的像素。

同断言相关联，自动生成:

- assertElementPositionTop (locator,pattern)
- assertNotElementPositionTop (locator,pattern)
- verifyElementPositionTop (locator,pattern)
- verifyNotElementPositionTop (locator,pattern)
- waitForElementPositionTop (locator,pattern)
- waitForNotElementPositionTop (locator,pattern)

storeElementWidth(locator,variableName)

返回元素的宽度

参数：

- locator – 一个指向元素的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

元素的宽度(以像素为单位)

同断言相关联，自动生成：

- assertElementWidth (locator,pattern)
- assertNotElementWidth (locator,pattern)
- verifyElementWidth (locator,pattern)
- verifyNotElementWidth (locator,pattern)
- waitForElementWidth (locator,pattern)
- waitForNotElementWidth (locator,pattern)

storeEval(script,variableName)

获得指定 JavaScript 片段执行后的值。该片段可以有多行，但只返回最后一行的值。要注意到，默认情况下，该片段将在”selenium”对象本身的上下文中运行，因此其将提交给 Selenium 对象。用 window 将窗体提交给你的应用程序，如：

window.document.getElementById(‘foo’)

如果你不得不在你的应用程序页面使用一个定位器提交一个单元素，你可以用this.browserbot.findElement(“id=foo”)，这里”id=foo”就是你的定位器。

参数：

- script – 要运行的 JavaScript
- variableName - 用于存储结果的变量名。

Returns:

片段执行后的值

同断言相关联，自动生成：

- assertEval (script,pattern)
- assertNotEval (script,pattern)
- verifyEval (script,pattern)
- verifyNotEval (script,pattern)
- waitForEval (script,pattern)

- waitForNotEval (script,pattern)

storeExpression(expression,variableName)

返回指定表达式。

由于 JavaScript 的预处理机制使其显得非常有用。它可以用于生成如 assertExpression 和 waitForExpression 命令。

参数：

- expression – 要返回的值
- variableName - 用于存储结果的变量名。

Returns:

通过的值

同断言相关联，自动生成：

- assertExpression (expression,pattern)
- assertNotExpression (expression,pattern)
- verifyExpression (expression,pattern)
- verifyNotExpression (expression,pattern)
- waitForExpression (expression,pattern)
- waitForNotExpression (expression,pattern)

storeHtmlSource(variableName)

返回”html”标签间的整个 HTML 源代码。

Returns:

整个 HTML 源代码

同断言相关联，自动生成：

- assertHtmlSource (pattern)
- assertNotHtmlSource (pattern)
- verifyHtmlSource (pattern)
- verifyNotHtmlSource (pattern)
- waitForHtmlSource (pattern)
- waitForNotHtmlSource (pattern)

storeLocation(variableName)

取得当前页面的绝对路径

Returns:

当前页面的绝对路径

同断言相关联，自动生成：

- assertLocation (pattern)
- assertNotLocation (pattern)
- verifyLocation (pattern)
- verifyNotLocation (pattern)
- waitForLocation (pattern)

- waitForNotLocation (pattern)

storeMouseSpeed(variableName)

返回在 dragAndDrop 命令执行期间，”mousemove”事件时的像素数字(默认为 10)

Returns:

dragAndDrop 命令执行期间，”mousemove”事件时的像素数字(默认为 10)

同断言相关联，自动生成：

- assertMouseSpeed (pattern)
- assertNotMouseSpeed (pattern)
- verifyMouseSpeed (pattern)
- verifyNotMouseSpeed (pattern)
- waitForMouseSpeed (pattern)
- waitForNotMouseSpeed (pattern)

storePrompt(variableName)

返回在之前动作所产生的 JavaScript 问题提示消息。

要成功挂起问题提示需要先运行 answerOnNextPrompt 命令。如果产生了一个问题提示，而你并不去得到/验证它，那么下一个 Selenium 动作将失败。

注意：在 Selenium 中，JavaScript 确认将不会弹出一个可见的对话框。

注意：Selenium 不支持在页面的 onload()事件句柄中所产生的 JavaScript 问题提示。在这种情况下，将会生成一个可见的对话框，Selenium 将被悬停直到手动点击 OK。

Returns:

最近 JavaScript 的问题提示消息

同断言相关联，自动生成：

- assertPrompt (pattern)
- assertNotPrompt (pattern)
- verifyPrompt (pattern)
- verifyNotPrompt (pattern)
- waitForPrompt (pattern)
- waitForNotPrompt (pattern)

storeSelectedId(selectLocator,variableName)

获取从指定选择元素中选择的选项元素 ID。

参数：

- selectLocator - 一个用于识别下拉菜单的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

从指定选择元素中选择的选项元素 ID

同断言相关联，自动生成：

- assertSelectedId (selectLocator,pattern)
- assertNotSelectedId (selectLocator,pattern)
- verifySelectedId (selectLocator,pattern)
- verifyNotSelectedId (selectLocator,pattern)
- waitForSelectedId (selectLocator,pattern)
- waitForNotSelectedId (selectLocator,pattern)

storeSelectedIds(selectLocator,variableName)

从指定选择或多选元素中获取选择的选项元素 ID 集。

参数：

- selectLocator - 一个用于识别下拉菜单的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

从指定选择或多选元素中选择的选项元素 ID 集合数组

同断言相关联，自动生成：

- assertSelectedIds (selectLocator,pattern)
- assertNotSelectedIds (selectLocator,pattern)
- verifySelectedIds (selectLocator,pattern)
- verifyNotSelectedIds (selectLocator,pattern)
- waitForSelectedIds (selectLocator,pattern)
- waitForNotSelectedIds (selectLocator,pattern)

storeSelectedIndex(selectLocator,variableName)

从指定的选择元素中获取被选项索引(从 0 开始)。

参数：

- selectLocator - 一个用于识别下拉菜单的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

从指定选择元素中选择的选项元素索引

同断言相关联，自动生成：

- assertSelectedIndex (selectLocator,pattern)
- assertNotSelectedIndex (selectLocator,pattern)
- verifySelectedIndex (selectLocator,pattern)
- verifyNotSelectedIndex (selectLocator,pattern)
- waitForSelectedIndex (selectLocator,pattern)
- waitForNotSelectedIndex (selectLocator,pattern)

storeSelectedIndexs(selectLocator,variableName)

从指定的选择或多选元素中获取被选项索引(从 0 开始)集。

参数：

- selectLocator - 一个用于识别下拉菜单的元素定位器

- variableName - 用于存储结果的变量名。

Returns:
从指定选择或多选元素中选择的选项元素索引集合数组

同断言相关联，自动生成：
- assertSelectedIndexs (selectLocator,pattern)
- assertNotSelectedIndexs (selectLocator,pattern)
- verifySelectedIndexs (selectLocator,pattern)
- verifyNotSelectedIndexs (selectLocator,pattern)
- waitForSelectedIndexs (selectLocator,pattern)
- waitForNotSelectedIndexs (selectLocator,pattern)

storeSelectedLabel(selectLocator,variableName)
从指定的选择元素中获取所选择的选项标签(可见文本)。
参数：
- selectLocator - 一个用于识别下拉菜单的元素定位器
- variableName - 用于存储结果的变量名。

Returns:
所选择的选项标签

同断言相关联，自动生成：
- assertSelectedLabel (selectLocator,pattern)
- assertNotSelectedLabel (selectLocator,pattern)
- verifySelectedLabel (selectLocator,pattern)
- verifyNotSelectedLabel (selectLocator,pattern)
- waitForSelectedLabel (selectLocator,pattern)
- waitForNotSelectedLabel (selectLocator,pattern)

storeSelectedLabels(selectLocator,variableName)
从指定的选择或多选元素中获取所选择的选项标签(可见文本)。
参数：
- selectLocator - 一个用于识别下拉菜单的元素定位器
- variableName - 用于存储结果的变量名。

Returns:
所选择的选项标签

同断言相关联，自动生成：
- assertSelectedLabel (selectLocator,pattern)
- assertNotSelectedLabel (selectLocator,pattern)
- verifySelectedLabel (selectLocator,pattern)
- verifyNotSelectedLabel (selectLocator,pattern)
- waitForSelectedLabel (selectLocator,pattern)
- waitForNotSelectedLabel (selectLocator,pattern)

storeSelectedValue(selectLocator,variableName)

从指定的选择元素中获取所选择的选项值(值属性)。

参数：

- selectLocator - 一个用于识别下拉菜单的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

所选择的选项值

同断言相关联，自动生成：

- assertSelectedValue (selectLocator,pattern)
- assertNotSelectedValue (selectLocator,pattern)
- verifySelectedValue (selectLocator,pattern)
- verifyNotSelectedValue (selectLocator,pattern)
- waitForSelectedValue (selectLocator,pattern)
- waitForNotSelectedValue (selectLocator,pattern)

storeSelectedValues(selectLocator,variableName)

从指定的选择或多选元素中获取所有所选择的选项值(值属性)。

参数：

- selectLocator - 一个用于识别下拉菜单的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

所有所选择的选项值数组

同断言相关联，自动生成：

- assertSelectedValues (selectLocator,pattern)
- assertNotSelectedValues (selectLocator,pattern)
- verifySelectedValues (selectLocator,pattern)
- verifyNotSelectedValues (selectLocator,pattern)
- waitForSelectedValues (selectLocator,pattern)
- waitForNotSelectedValues (selectLocator,pattern)

storeSelectOptions(selectLocator,variableName)

获取指定选择下拉框的选项标签。

参数：

- selectLocator - 一个用于识别下拉菜单的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

指定选择下拉框的选项标签数组

同断言相关联，自动生成：

- assertSelectedOptions (selectLocator,pattern)
- assertNotSelectedOptions (selectLocator,pattern)
- verifySelectedOptions (selectLocator,pattern)

- verifyNotSelectedOptions (selectLocator,pattern)
- waitForSelectedOptions (selectLocator,pattern)
- waitForNotSelectedOptions (selectLocator,pattern)

storeTable(tableCellAddress,variableName)

从某个表的单元格中获取文本。单元格地址语法如 tablelocator.row.column，这里的 row 和 column 从 0 开始。

参数：

- tableCellAddress – 一个单元格地址，如”foo.1.4”
- variableName -用于存储结果的变量名。

Returns:

从指定单元格取出的文本

同断言相关联，自动生成：

- assertTable (tableCellAddress, pattern)
- assertNotTable (tableCellAddress, pattern)
- verifyTable (tableCellAddress, pattern)
- verifyNotTable (tableCellAddress, pattern)
- waitForTable (tableCellAddress, pattern)
- waitForNotTable (tableCellAddress, pattern)

storeText(locator,variableName)

获取元素的文本。这对任何包含文本的元素都有效。该命令即可以用于如火狐浏览器的 textContent，也可以用于如 IE 浏览器的 innerText——显示给用户的。

参数：

- locator - 一个元素定位器
- variableName - 用于存储结果的变量名。

Returns:

元素的文本

同断言相关联，自动生成：

- assertText (locator,pattern)
- assertNotText (locator,pattern)
- verifyText (locator,pattern)
- verifyNotText (locator,pattern)
- waitForText (locator,pattern)
- waitForNotText (locator,pattern)

storeTitle(variableName)

获取当前页面的标题。

Returns:

当前页面的标题。

同断言相关联，自动生成：

- assertTitle (pattern)
- assertNotTitle (pattern)
- verifyTitle (pattern)
- verifyNotTitle (pattern)
- waitForTitle (pattern)
- waitForNotTitle (pattern)

storeValue(locator,variableName)

获得一个输入域(或任何包含 value 参数的元素)的值(已去除空格)。对于 checkbox/radio 元素，其值为”on”还是”off”依赖于该元素是否被选中。

参数：

- locator - 一个元素定位器
- variableName - 用于存储结果的变量名。

Returns:

元素值，对于 checkbox/radio 元素则为”on/off”

同断言相关联，自动生成：

- assertValue (locator,pattern)
- assertNotValue (locator,pattern)
- verifyValue (locator,pattern)
- verifyNotValue (locator,pattern)
- waitForValue (locator,pattern)
- waitForNotValue (locator,pattern)

storeWhetherThisFrameMatchFrameExpression(currenFrameString,target,variableName)

确定当前框架是否包含该运行代码。

这对于代理注入代码模式非常有用，这些代码在每个浏览器框架和窗体中运行，有时 selenium server 需要识别当前是哪个框架。这种情况下，当测试调用 selectFrame，该程序将被每个框架调用以指出哪个框架被选择。被选择的框架将返回 true，而其他将返回 false。

参数：

- currenFrameString – 开始框架
- target – 新框架(也许与当前框架相关联)
- variableName -用于存储结果的变量名。

Returns:

如果新框架为该代码的窗体，返回 true

同断言相关联，自动生成：

- assertWhetherThisFrameMatchFrameExpression (currenFrameString, target)
- assertNotWhetherThisFrameMatchFrameExpression (currenFrameString, target)

- verifyWhetherThisFrameMatchFrameExpression (currenFrameString, target)
- verifyNotWhetherThisFrameMatchFrameExpression (currenFrameString, target)
- waitForWhetherThisFrameMatchFrameExpression (currenFrameString, target)
- waitForNotWhetherThisFrameMatchFrameExpression (currenFrameString, target)

storeWhetherThisWindowMatchWindowExpression(currenWindowString,target,variableName)

确定当前窗体是否包含该运行代码。

这对于代理注入代码模式非常有用，这些代码在每个浏览器框架和窗体中运行，有时 selenium server 需要识别当前是哪个窗体。这种情况下，当测试调用 selectWindow，该程序将被每个窗体调用以指出哪个窗体被选择。被选择的窗体将返回 true，而其他将返回 false。

参数：

- currenFrameString – 开始窗体
- target – 新窗体(也许与当前窗体相关联，如，”_parent”)
- variableName - 用于存储结果的变量名。

Returns:

如果新窗体为该代码的窗体，返回 true

同断言相关联，自动生成：

- assertWhetherThisWindowMatchWindowExpression (currenWindowString, target)
- assertNotWhetherThisWindowMatchWindowExpression (currenWindowString, target)
- verifyWhetherThisWindowMatchWindowExpression (currenWindowString, target)
- verifyNotWhetherThisWindowMatchWindowExpression (currenWindowString, target)
- waitForWhetherThisWindowMatchWindowExpression (currenWindowString, target)
- waitForNotWhetherThisWindowMatchWindowExpression (currenWindowString, target)

storeXpathCount(xpath,variableName)

返回匹配指定 xpath 的节点数，如”//table”将给出表的个数。

参数：

- xpath – 要计算的 xpath 表达式。不要用’count()’函数将该表达式包含起来，我们将自动帮你作这件事。
- variableName - 用于存储结果的变量名。

Returns:
匹配指定 xpath 的节点数

同断言相关联，自动生成：
- assertXpathCount (xpath, pattern)
- assertNotXpathCount (xpath, pattern)
- verifyXpathCount (xpath, pattern)
- verifyNotXpathCount (xpath, pattern)
- waitForXpathCount (xpath, pattern)
- waitForNotXpathCount (xpath, pattern)

storeAlertPresent(variableName)
发生警告了?
该函数永远不会抛出异常
Returns:
如果有警告返回 true

同断言相关联，自动生成：
- assertAlertPresent ()
- assertNotAlertPresent ()
- verifyAlertPresent ()
- verifyNotAlertPresent ()
- waitForAlertPresent ()
- waitForNotAlertPresent ()

storeChecked(locator,variableName)
获取一个关联性按钮(checkbox/radio)是否被勾选。如果指定元素不存在或不是一个关联性按钮，将失败。
参数：
- locator– 一个执行 checkbox 或 radio 按钮的元素定位器
- variableName - 用于存储结果的变量名。

Returns:
如果该 checkbox 被勾选，返回 true,否则返回 false

同断言相关联，自动生成：
- assertChecked (locator)
- assertNotChecked (locator)
- verifyChecked (locator)
- verifyNotChecked (locator)
- waitForChecked (locator)
- waitForNotChecked (locator)

storeConfirmationPresent(variableName)
confirm()被调用了？

该函数永远不会抛出异常

Returns:

如果有一个未决的确认返回 true

同断言相关联，自动生成：

- assertConfirmationPresent ()
- assertNotConfirmationPresent ()
- verifyConfirmationPresent ()
- verifyNotConfirmationPresent ()
- waitForConfirmationPresent ()
- waitForNotConfirmationPresent ()

storeEditable(locator,variableName)

判定指定的输入元素是否为可编辑，且 ie 没有被禁用。如果指定元素不为一个可输入元素，该函数将失败。

参数：

- locator – 一个元素定位器
- variableName - 用于存储结果的变量名。

Returns:

如果输入元素可编辑返回 true，否则返回 false

同断言相关联，自动生成：

- assertEditable (locator)
- assertNotEditable (locator)
- verifyEditable (locator)
- verifyNotEditable (locator)
- waitForEditable (locator)
- waitForNotEditable (locator)

storeElementPresent(locator,variableName)

验证指定元素在页面上。

参数：

- locator – 一个元素定位器
- variableName - 用于存储结果的变量名。

Returns:

如果该元素出现返回 true，否则返回 false

同断言相关联，自动生成：

- assertElementPresent (locator)
- assertNotElementPresent (locator)
- verifyElementPresent (locator)
- verifyNotElementPresent (locator)
- waitForElementPresent (locator)
- waitForNotElementPresent (locator)

storeOrdered(locator1,locator2,variableName)

检查这两个元素是否有相同的父级，且在 DOM 中为顺序亲属。两个相同元素将不考虑顺序。

参数：

- locator1 – 指向第一个元素的元素定位器
- locator2 – 指向第二个元素的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

如果元素 1 是元素 2 的兄长，返回 true，否则返回 false

同断言相关联，自动生成：

- assertOrdered (locator1,locator2)
- assertNotOrdered (locator1,locator2)
- verifyOrdered (locator1,locator2)
- verifyNotOrdered (locator1,locator2)
- waitForOrdered (locator1,locator2)
- waitForNotOrdered (locator1,locator2)

storePromptPresent(variableName)

发生提示了？

该函数永远不会抛出异常

Returns:

如果有一个未决的提示返回 true

同断言相关联，自动生成：

- assertPromptPresent ()
- assertNotPromptPresent ()
- verifyPromptPresent ()
- verifyNotPromptPresent ()
- waitForPromptPresent ()
- waitForNotPromptPresent ()

storeSomethingSelected(selectLocator,variableName)

判定一个下拉菜单是否选择了某个选项。

参数：

- selectLocator – 一个用于识别下拉菜单的元素定位器
- variableName - 用于存储结果的变量名。

Returns:

如果选择某选项返回 true，否则返回 false

同断言相关联，自动生成：

- assertSomethingSelected (selectLocator)
- assertNotSomethingSelected (selectLocator)

- verifySomethingSelected（selectLocator）
- verifyNotSomethingSelected（selectLocator）
- waitForSomethingSelected（selectLocator）
- waitForNotSomethingSelected（selectLocator）

storeTextPresent(pattern,variableName)

验证指定文本出现在提交给用户的页面上。

参数：

- pattern – 用于匹配页面文本的范式
- variableName - 用于存储结果的变量名。

Returns:

如果该范式匹配文本返回 true，否则返回 false

同断言相关联，自动生成：

- assertTextPresent（pattern）
- assertNotTextPresent（pattern）
- verifyTextPresent（pattern）
- verifyNotTextPresent（pattern）
- waitForTextPresent（pattern）
- waitForNotTextPresent（pattern）

storeVisible(locator,variableName)

判定指定元素是否可见。一个元素可以通过将其本身或其父级的CSS”visibility”属性设定为”hidden”，或将”dispaly”属性设定为”none”，以使其不可见。如果该元素不存在，此方法将失败。

参数：

- locator – 一个元素定位器
- variableName - 用于存储结果的变量名。

Returns:

如果指定元素为可见返回 true，否则返回 false

同断言相关联，自动生成：

- assertVisible（locator）
- assertNotVisible（locator）
- verifyVisible（locator）
- verifyNotVisible（locator）
- waitForVisible（locator）
- waitForNotVisible（locator）