seleniumpc

Introduction

seleniumpc is a selenium wrapper, in the hope of simplifying the use of the original selenium. seleniumpc reduces the number of classes, tries to make definition names more intuitive, and let the program move more like humans. Finally seleniumpc could make it easier for junior test engineers to get started with selenium skills, and write, debug and maintain their own test code.

Develop & Test Environment

system: Windows 7 a64
runtime: Python 2.7 i386
browser: Chrome 29 i386, IE 8 i386, FireFox 20 i386
language: Simplified Chinese

Classes in selenium

Driver

Driver represents the instance of a browser.

Element

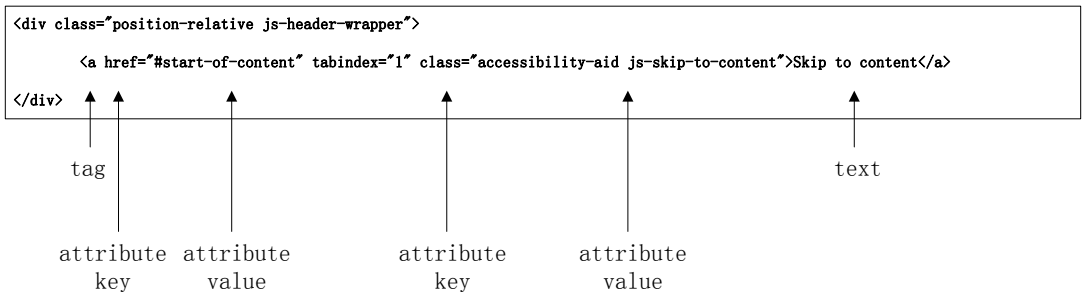
Element represents a DOM instance within a web page.

Attribute

Attribute represents one of the several attributes belonging to a DOM.

Text

Text represents the text field of a DOM.



DOM's tag, attributes & text

Key

Key is a collection of special keys like ENTER/HOME/LEFT.

Properties & Definitions in Driver

Driver.name

Your browser's name, must be a string from 'chrome/ie/ff'.

Driver.executor

Tthe browserdriver.exe's absolute path, must be a string.
Only Chrome & IE require this property.

Driver.browser

The browser's absolute path, must be a string.
Only Chrome & FireFox require this property.

Driver.proxy
http, ssl and ftp proxy to your browser, must be a string in the form of 'address:port'.

Driver.option
Command line option to your browser, must be a string. Pass one option at a time if you would like to set multiple options.
Chrome: http://peter.sh/experiments/chromium-command-line-switches/
FireFox: https://developer.mozilla.org/en-US/docs/Mozilla/Command_Line_Options#Browser
IE: https://msdn.microsoft.com/en-us/library/hh826025(v=vs.85).aspx

Driver.log
The absolute path of where you would like to save the log, must be a string.

Driver.delay
A global delay to slow every step down during your test run, must be an int above 0.
The unit is millisecond, the default value is 700.

Driver.launch()
Launch the browser.

Driver.quit()
Close the browser.

Driver.open(url)
Open the url.
url:
The web page address, must be a string.

Driver.refresh()
Refresh the web page.

Driver.back()
Go back to the previous web page.

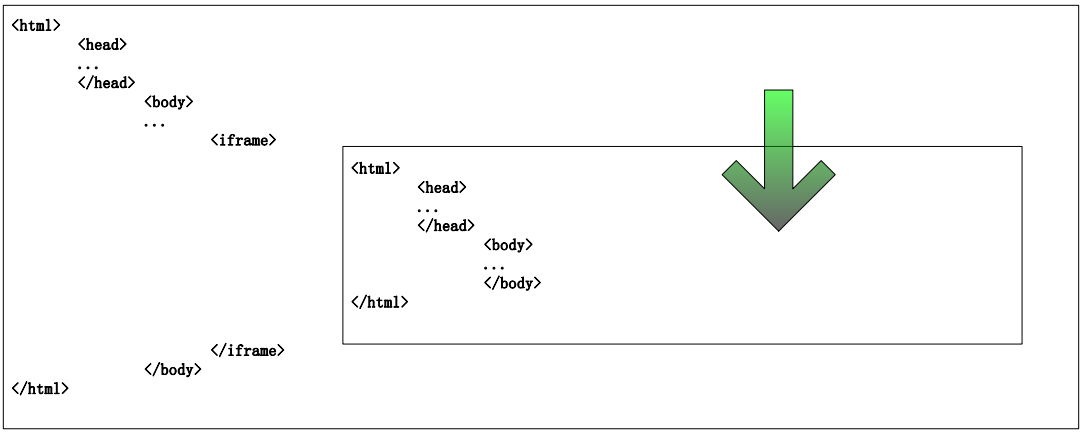
Driver.forward()
Go forward to the next web page.

Driver.close()
Close the web page.

Driver.find(tag = '*', attribute = list(), text = list())
Find all the DOMs within the web page who match the tag, attributes and text you passed.
Return a list of found Element instances.
tag:
The tag name to your desired DOM, must be a string.
The default value is '*', which means to match any tag name.
attribute:

One of the attributes to your desired DOM, must be an Attribute instance, or a list of Attribute instance.
The default value is [], which means to match any attribute.
text:
The text field to your desired DOM node, must be a Text instance, or a list of Text instance.
The default value is [], which means to match ant test field.

Driver. framein (element)
Switch into the frame DOM you passed.
element:
An Element instance who is returned by Driver.find()/Element.find().



Switch into a frame

Driver.frameout()
Switch out of the frame DOM.

Driver.type(key)
Simulate typing on the keyboard.
key:
The characters that could found on your keyboard, must be an int, a string, or a Key instance, even a list nesting ints/strings/Key instances.

Driver.modifierdown(modifier)
Simulate pressing down a modifier key without release.
modifier:
A modifier key who could be found inside the Key class, like CONTROL/SHIFT/ALT. Must be a Key instance, or a list of Key instances.

Driver.modifierdup(modifier)
Simulate releasing a modifier key.
modifier:
A modifier key who could be found inside the Key class, like CONTROL/SHIFT/ALT. Must be a Key instance, or a list of Key instances.

Driver.alert(accept = True, send = None, timeout = 7000)
This definition describes how the browser would deal with a HTML prompt.
Return a string of the prompt message.
accept:

If accept were True, the browser would accept the HTML prompt, otherwise the browser would dismiss it.
send:
The content you would like to send into the HTML prompt, must be a string.
The default value is None, which means to send nothing.
timeout:
The timeout that the browser would wait for the HTML prompt, must be an int above 0.
The unit is millisecond, the default value is 7000.

Driver.upload(path, timeout = 7000)
When you click on a button on a web page, the 'Select File...' dialog pops up. Now this definition would fill-in and close the dialog automatically.
path:
The absolute path to the file that you would like to upload, must be a string.
timeout:
The timeout that the browser would wait for the 'Select File...' dialog, must be an int above 0.
The unit is millisecond, the default value is 7000.

Driver.download(path, timeout = 7000)
When you click on a link on a web page, the 'Save As...' dialog pops up. Now this definition would fill-in and close the dialog automatically.
path:
The absolute path of where you would like to save the downloaded file, must be a string.
timeout:
The timeout that the browser would wait for the 'Save As...' dialog, must be an int above 0.
The unit is millisecond, the default value is 7000.

Driver.title()
Return a string of the web page title.

Driver.url()
Return a string of the web page url.

Definitions in Element
Element.find(tag = '*', attribute = list(), text = list())
Under the current Element instance, find all the DOMs who match the tag, attributes and text you passed.
Return a list of found Element instances.
tag:
The tag name to your desired DOM, must be a string.
The default value is '*', which means to match any tag name.
attribute:
One of the attributes to your desired DOM, must be an Attribute instance, or a list of Attribute instance.
The default value is [], which means to match any attribute.
text:
The text field to your desired DOM node, must be a Text instance, or a list of Text instance.
The default value is [], which means to match ant test field.

Element.parent()
Return an Element instance of the parent node of the current DOM.

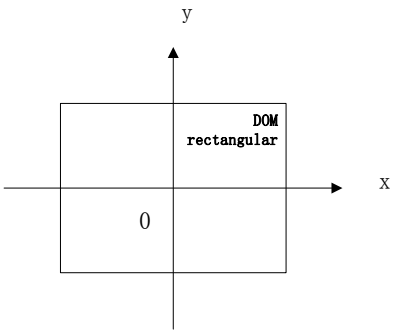
Element.hover(x = 0, y = 0)

Simulate moving mouse cursor to the centre of the current DOM.

x/y:

x/y is the offset to the centre of the current DOM, must be an int above 0.

The unit is pixel, the default value is 0.



Coordinate offsets

Element.click(x = 0, y = 0, count = 1)

Simulate clicking on the current DOM.

x/y:

x/y is the offset to the centre of the current DOM, must be an int above 0.

The unit is pixel, the default value is 0.

count:

How many times you would like to click, the default value is 1.

Element.mousedown(x = 0, y = 0)

Simulate pressing mouse down without release on the current DOM.

x/y:

x/y is the offset to the centre of the current DOM, must be an int above 0.

The unit is pixel, the default value is 0.

Element.mouseup(x = 0, y = 0)

Simulate releasing mouse on the current DOM.

x/y:

x/y is the offset to the centre of the current DOM, must be an int above 0.

The unit is pixel, the default value is 0.

Element.mousepress(duration, x = 0, y = 0)

Simulate pressing mouse down on the current DOM for a period of time, and then releasing.

duration:

How long you would like to hold pressing mouse down.

x/y:

x/y is the offset to the centre of the current DOM, must be an int above 0.

The unit is pixel, the default value is 0.

Element.clear()

Clear the content of the current input DOM.

Element.send(send)

Send content into the current input DOM.

send:

The content you would like to send into the current input DOM, must be a string.

Element. waitexist(tag = '*', attribute = list(), text = list(), timeout = 7000)

Under the current Element instance, wait until a DOM could be found.

tag:

The tag name to your desired DOM, must be a string.

The default value is '*', which means to match any tag name.

attribute:

One of the attributes to your desired DOM, must be an Attribute instance, or a list of Attribute instance.

The default value is [], which means to match any attribute.

text:

The text field to your desired DOM node, must be a Text instance, or a list of Text instance.

The default value is [], which means to match ant test field.

timeout:

The timeout that the browser would wait for the existence of your desired DOM, must be an int above 0.

The unit is millisecond, the default value is 7000.

Element. waitextinct(tag = '*', attribute = list(), text = list(), timeout = 7000)

Under the current Element instance, wait until a DOM could not be found.

tag:

The tag name to your desired DOM, must be a string.

The default value is '*', which means to match any tag name.

attribute:

One of the attributes to your desired DOM, must be an Attribute instance, or a list of Attribute instance.

The default value is [], which means to match any attribute.

text:

The text field to your desired DOM node, must be a Text instance, or a list of Text instance.

The default value is [], which means to match ant test field.

timeout:

The timeout that the browser would wait for the extinction of your desired DOM, must be an int above 0.

The unit is millisecond, the default value is 7000.

Element.tag()

Return a string of the current DOM's tag name.

Element.attribute(key)

Return a string of the current DOM's attribute value to the passed attribute key.

key:

The key of one of the current DOM's several attribute key-value pairs, must be a string.

Element.text()

Return a string of the current DOM's text field.

Element.width()

Return the width of the current DOM.

Element.height()

Return the height of the current DOM.

Element. isdisplay()
Return True if the current DOM were being displayed, otherwise False.

Element. isselect()
Return True if the current DOM were being selected, otherwise False.

Element. isenable()
Return True if the current DOM were enabled, otherwise False.

Definition in Attribute
Attribute.__init__(key, value, strict = True)
Init an Attribute instance.
key:
The key of the attribute key-value pair, must be a string.
value:
The value of the attribute key-value pair, must be a string.
strict:
If strict were True, it means you want to find a DOM whose attribute value of the key should be equal to the string you passed. else, it means the DOM whose attribute value of the key should contain the string you passed.

Definition in Text
Text.__init__(text, strict = True)
Init a Text instance.
text:
The text field of a DOM, must be a string.
strict:
If strict were True, it means you want to find a DOM whose text field should be equal to the string you passed. else, it means the DOM whose text field should contain the string you passed.

Log
The Log instance would write down every definition you called during the test run, the definition's name, the parameters you passed, and the execution result of the definition. If an exception were raised, the Log instance would write down the Exception.msg and traceback, and finally take a screenshot.