

KIỂM THỬ PHẦN MỀM

KIỂM THỬ TỰ ĐỘNG

ThS. Dương Hữu Thành
Khoa CNTT, Đại học Mở Tp.HCM
thanh.dh@ou.edu.vn



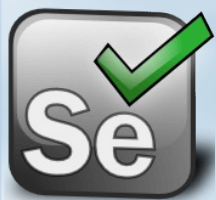
Software
testing





Nội dung chính

- 1. Giới thiệu kiểm thử tự động.**
- 2. Giới thiệu Selenium.**
- 3. Selenium IDE.**
- 4. Selenium Web Driver.**
- 5. IWebDriver & WebElement.**
- 6. Wait.**



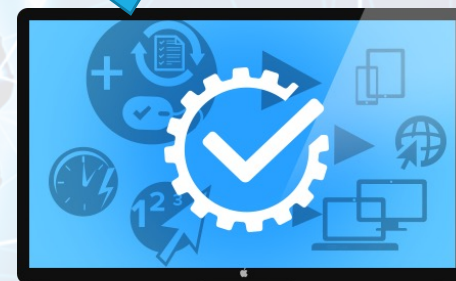
Giới thiệu kiểm thử tự động

Kiểm thử tự động (Automation Testing) là kỹ thuật **sử dụng công cụ** nào đó **để thực thi các test case một cách tự động** và đưa ra kết quả kiểm thử một cách nhanh chóng.

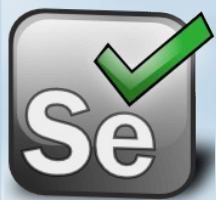
Kiểm thử tự động đặc biệt có hiệu quả trong **kiểm thử hồi quy** (regression testing).



Kiểm thử thủ công
(Manual Testing)

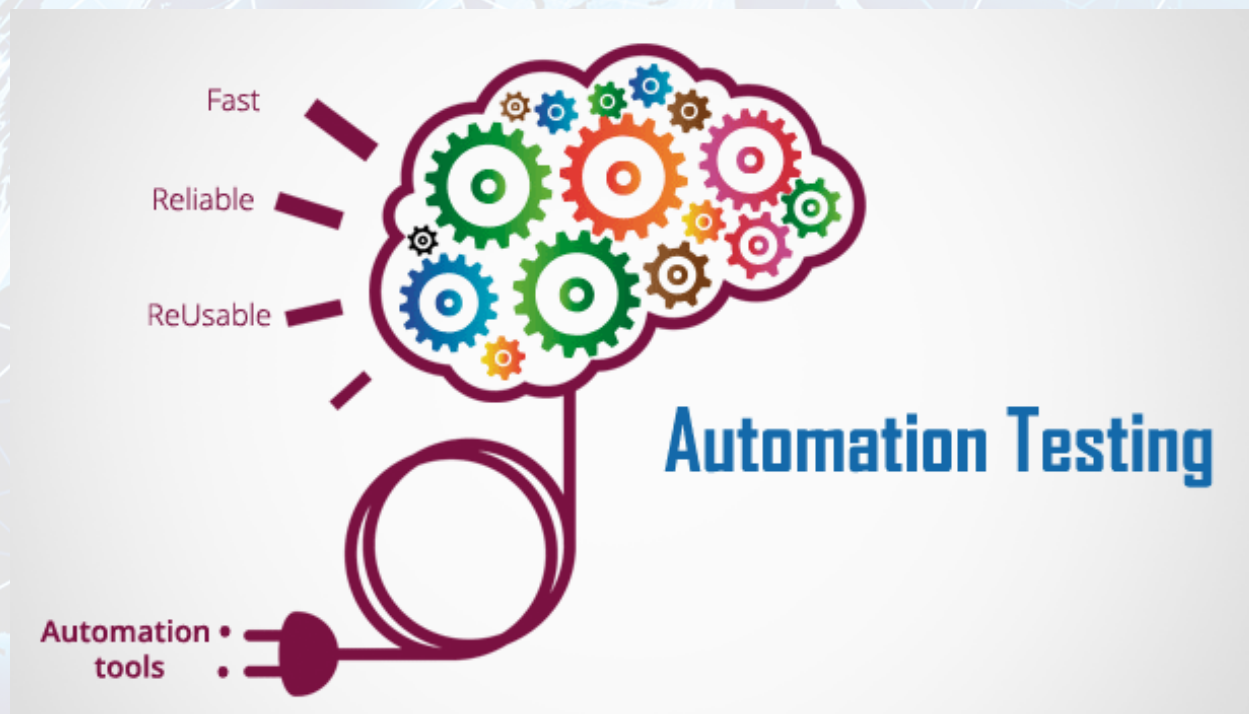


Kiểm thử tự động
(Automation Testing)

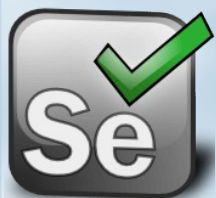


Giới thiệu kiểm thử tự động

Mục đích của kiểm thử tự động giúp **giảm thời gian, chi phí, tăng độ tin cậy, nâng cao hiệu suất công việc**, đồng thời **giảm sự nhầm lẫn** cho kiểm thử viên.



Nguồn ảnh: Viblo

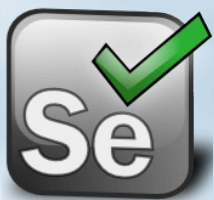


Giới thiệu kiểm thử tự động

Làm sao để
trở thành
kiểm thử viên
tự động

- **Hiểu các kỹ thuật kiểm thử.**
- **Có khả năng lập trình**
 - Thành thạo ít nhất một ngôn ngữ lập trình: Java, C#, Python, Ruby, Javascript, v.v.
 - Thành thạo ít nhất một framework kiểm thử: Selenium, NUnit, JUnit, TestNG, v.v.
 - Nắm rõ HTML, CSS Selector, XPath để tương tác các thành phần giao diện.
 - Hiểu về design pattern.





Giới thiệu kiểm thử tự động

Một số công cụ kiểm thử tự động

Selenium

LoadRunner

jMeter

jUnit/ NUnit

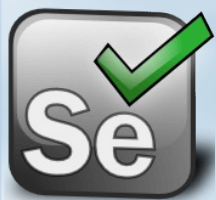
Quick Test Professional (HP)

Katalon Studio



JUnit





Giới thiệu Selenium

Selenium là công cụ **mã nguồn mở** dùng cho kiểm thử tự động các **ứng dụng web** được phát triển bởi ThoughtWorks năm 2004.

Selenium có thể hoạt động trên nhiều trình duyệt và hệ điều hành khác nhau.

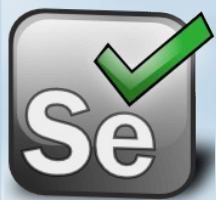
Các công cụ của Selenium

- Selenium IDE.

- Selenium RC.

- Selenium WebDriver.

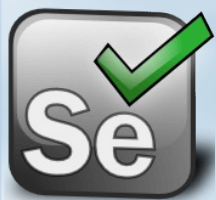
- Selenium Grid.



Giới thiệu Selenium

Selenium IDE: là một plugin trình duyệt Firefox và Chrome. Nó có thể ghi nhận những hành động cần kiểm thử.

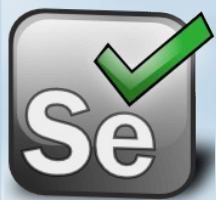
Selenium RC (Remote Control): là một framework cho phép thực hiện nhiều hành động trên trình duyệt và thực thi tuyến tính, nó có thể dùng trong nhiều ngôn ngữ lập trình như Java, C#, Python, PHP.



Giới thiệu Selenium

Selenium WebDriver: nối tiếp của Selenium RC là Selenium WebDriver cho phép gửi các lệnh trực tiếp lên trình duyệt và tìm kiếm kết quả.

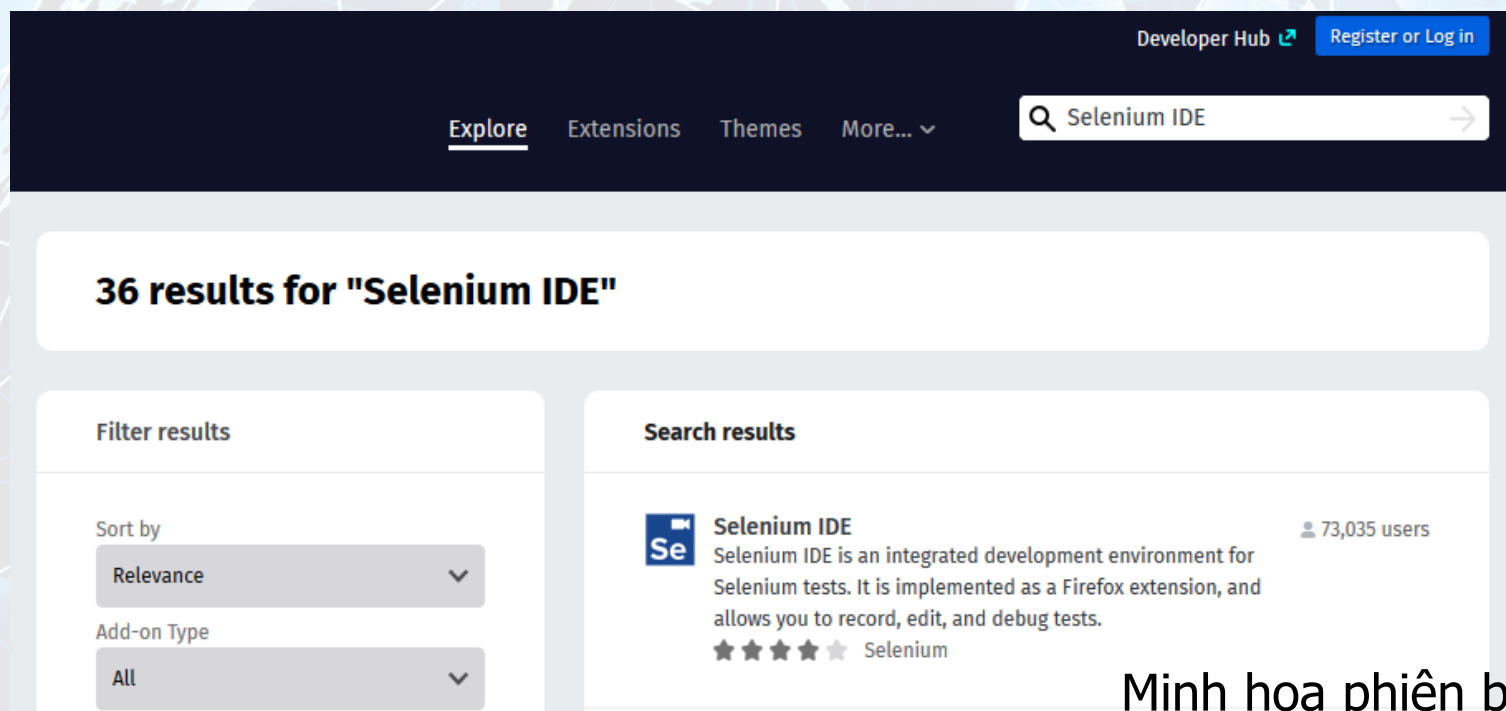
Selenium Grid: là công cụ dùng chạy các test song song trên nhiều máy và nhiều trình duyệt khác nhau một cách đồng thời để tối ưu thời gian thực thi.



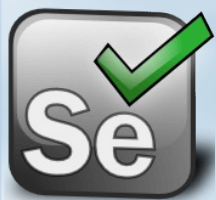
Selenium IDE

Mở trình duyệt Mozilla FireFox. Vào menu **Tools** > **Add-ons** hoặc dùng tổ hợp phím Ctrl+Shift+A.

Trong cửa sổ Add-ons Manager, tìm từ khóa Selenium IDE.

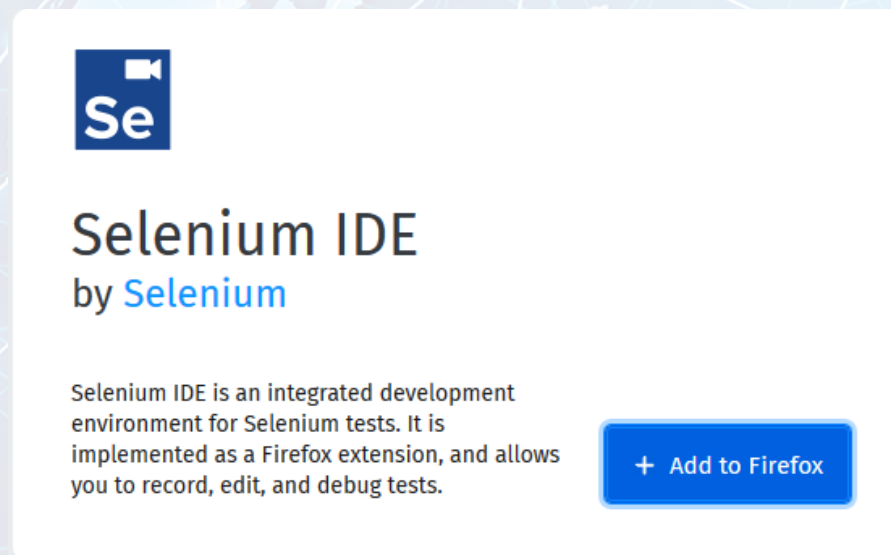


Minh họa phiên bản 61.0.2

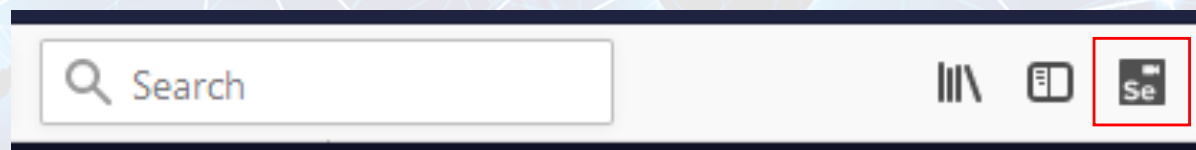


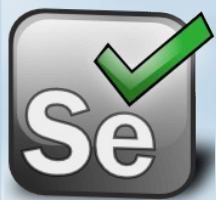
Selenium IDE

Click chọn Selenium IDE tìm thấy và click nút **Add to Firefox.**



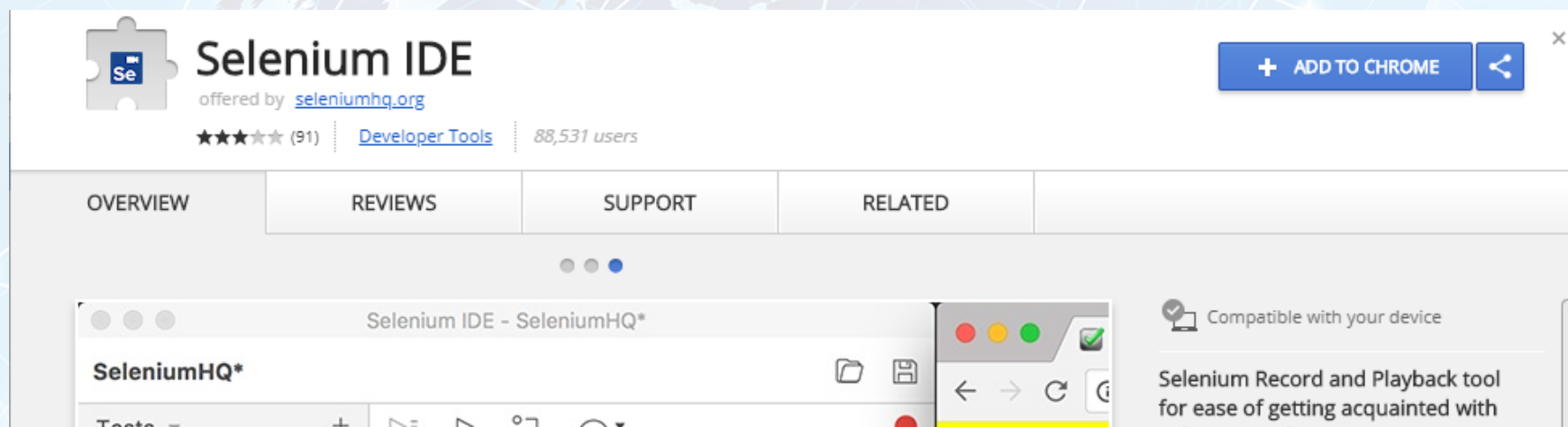
Một icon được thêm vào trên thanh tìm kiếm.

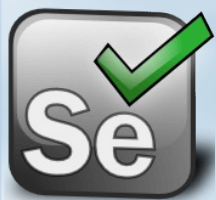




Selenium IDE

Ngoài ra, ta vẫn có thể cài đặt Selenium IDE trên Chrome từ **Chrome Store**.





Selenium IDE

Selenium
IDE

Các test
case

moz-extension://64eb0a41-3e90-4dca-8c80-ce71384df4c5 - Selenium IDE - Untitled Project - Mozilla ...

Untitled Project

Tests ▾ +

Search tests... 🔍

Untitled

Playback base URL ▾

Command	Target	Value

Command ▾

Target 🔍

Value

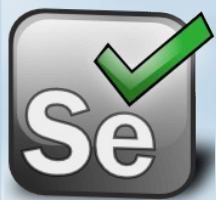
Comment

Log Reference

Các công
cụ điều
 khiển

Các bước
 trong test
 case

Thiết lập
 thông tin
 của test
 case

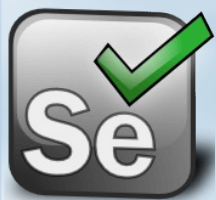


Selenium IDE

Command: chứa các lệnh để tạo test case.

Target: chỉ định vị trí thành phần cần tương tác.

Value: đặt giá trị cho thành phần chỉ định trong target nếu cần.



Selenium IDE



Tự động nhập liệu
tìm kiếm với Google

Thực thi test

Mở trang tìm kiếm
Google

Tìm ô nhập liệu và
nhập giá trị
"Software testing".

Click vô nút tìm kiếm

moz-extension://64eb0a41-3e90-4dca-8c80-ce71384df4c5 - Selenium IDE - Google Test * ...

Google Test *

Executing ▾

Input data*

Playback base URL

	Command	Target	Value
1	open	https://www.google.com.vn/	
2	send keys	id=lst-ib	Software testing
3	click	name=btnK	

Command: click

Target: name=btnK

Value:

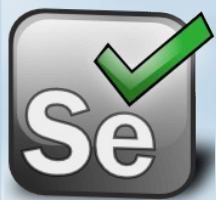
Comment:

Runs: 1 Failures: 0

Log Reference

3. click on name=btnK... OK

'Input data' completed successfully

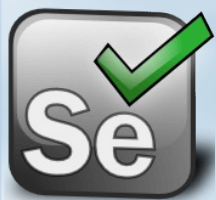


Selenium WebDriver

Đặc trưng nổi bật trong Selenium 2.0 là bổ sung WebDriver:

Selenium 2.0 = Selenium 1.0 + WebDriver

Selenium WebDriver cung cấp các API cho nhiều ngôn ngữ lập trình C#, Java, Python, PHP, Ruby, v.v. để tự động hóa việc kiểm thử.



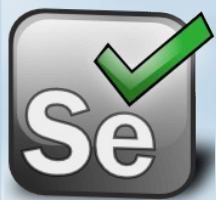
Selenium WebDriver

Developer hoặc tester viết các **kịch bản** kiểm thử sử dụng các **API của selenium** trên các ngôn ngữ lập trình mong muốn.

Kịch bản này chỉ định **tương tác với trình duyệt cụ thể** của WebDriver như Chrome, FireFox, IE, v.v. để tương tác với ứng dụng web.

Sau khi **thực thi**, WebDriver sẽ trả về **kết quả xử lý**.





Cài đặt Python Selenium

Download Python và double-click tập tin cài đặt

<https://www.python.org/downloads/>

Mở Command Prompt (cmd) gõ lệnh

```
pip install selenium
```

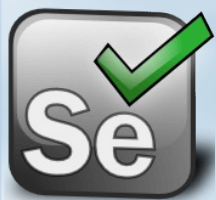
Cài IDE PyCharm Community

<https://www.jetbrains.com/pycharm/download>

Download driver cho các trình duyệt kiểm thử

<https://chromedriver.chromium.org/downloads>

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



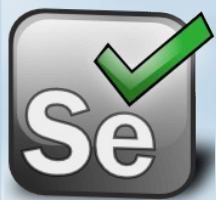
Viết chương trình đầu tiên

Truy cập trang web google hiển thị tiêu đề trang và mã nguồn HTML của trang.

```
from selenium import webdriver

driver = webdriver.Chrome(executable_path="...")
driver.get("https://www.google.com.vn/")
print(driver.title)
print(driver.page_source)

driver.close()
```



Viết chương trình đầu tiên

Các thuộc tính của webdriver

url: thiết lập/hoặc lấy url của trang web.

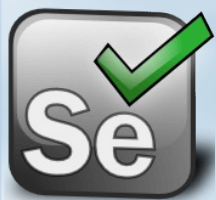
title: lấy chủ đề trang web.

page_source: lấy mã nguồn HTML của trang web.

Các phương thức của webdriver

close(): đóng cửa sổ webdriver đang điều khiển.

quit(): đóng tất cả các cửa sổ mở bởi IWebDriver.



Tương tác HTML DOM

`find_element_by_id`: dùng id

`find_element(s)_by_name`: dùng thuộc tính name

`find_element(s)_by_tag_name`: dùng tên thẻ

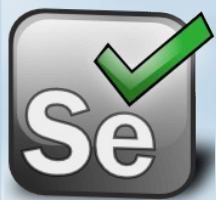
`find_element(s)_by_class_name`: dùng thuộc tính class

`find_element(s)_by_link_text`: tìm thẻ a dựa trên text.

`find_element_by_partial_link_text`

`find_element(s)_by_css_selector`: dùng css selector

`find_element(s)_by_xpath`: dùng xpath



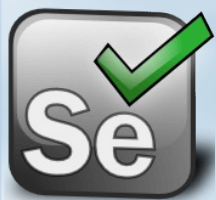
Tương tác HTML DOM

Truyền nội dung và thực hiện tìm kiếm google

```
driver = webdriver.Chrome(executable_path="...")
driver.get("https://www.google.com.vn/")

ele = driver.find_element_by_name("q")
ele.send_keys("software testing")
ele.submit()

driver.close()
```

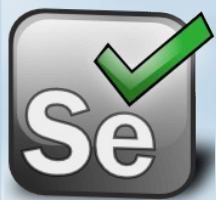
Tương tác với HTML DOM



Webdriver cung cấp 2 phương thức để tìm thành phần HTML để tương tác.

find_element trả về thành phần đầu tiên tìm thấy trong cây HTML DOM. Nếu không tìm thấy, nó sẽ ném ngoại lệ NoSuchElementException.

find_elements trả về một danh sách các thành phần tìm thấy trong cây HTML DOM, và sẽ trả về danh sách rỗng nếu không tìm thấy.



Tương tác HTML DOM

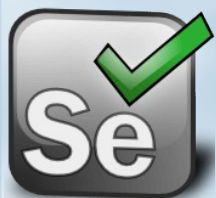
Google

Tìm với Google

Xem trang đầu tiên tìm được

Console Sources Network Performance Memory Application Security Audits

```
<div class="gsfi" id="lst-ib" style="height: 44px; >
  <div class="gsst_b sbib_c" id="gs_st0" dir="ltr" style="line-height: 44px; >...</div>
  <div class="sbib_b" id="sb_ifc0" dir="ltr">
    <div id="gs_lc0" style="position: relative; >
      <input class="gsfi" id="lst-ib" maxlength="2048" name="q" autocomplete="off" title="Tìm kiếm" type="text" value=""
        aria-label="Tìm kiếm" aria-haspopup="false" role="combobox" aria-autocomplete="list" style="border: none; padding:
        0px; margin: 0px; height: auto; width: 100%; background: url("data:image/gif;base64,R0lGODlhAQABAID/
        AMDAwAAAAACH5BAEAAAAALAAAAABAAEAAAIICRAEAOw%3D%3D") transparent; position: absolute; z-index: 6; left: 0px;
        outline: none;" dir="ltr" spellcheck="false"> == $0
      <div class="gsfi" id="gs_sc0" style="background: transparent; color: transparent; padding: 0px; position:
        absolute; z-index: 2; white-space: pre; visibility: hidden;"></div>
      <input class="gsfi" disabled autocomplete="off" autocapitalize="off" aria-hidden="true" id="gs_taif0" dir="ltr"
```

Tương tác HTML DOM

Ví dụ

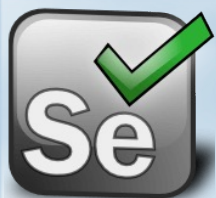
```
from selenium import webdriver
from selenium.webdriver.common.by import By

driver = webdriver.Chrome(executable_path="...")
driver.get("https://www.google.com.vn/")

element = driver.find_element(By.NAME, "q")
element.send_keys("software testing")
element.submit()

elements = driver.find_elements(By.CSS_SELECTOR, "div.g")
for ele in elements:
    print(ele.find_element(By.TAG_NAME, "h3").text)

driver.close()
```



Tương tác HTML DOM

Kết quả minh họa

```
Run: index x
/Users/duonghuuthanh/PycharmProjects/seleniumdemo/venv/bin/python /Users
What is Software Testing? Introduction, Definition, Basics & Types
Software Testing Tutorial: Free Course - Guru99
Software testing - Wikipedia
What Is Software Testing - Definition, Types, Methods, Approaches
Software Testing Help - Free Software Testing & IT Tutorials and ...
Software Testing Tutorial
Software Testing | Basics - GeeksforGeeks
Welcome to Software Testing Fundamentals (STF) ! - Software Testing ...
The 7 Common Types of Software Testing! - Usersnap
```




Tương tác HTML DOM

Điền văn bản vào ô nhập liệu

```
element.send_keys(text)
```

Xoá giá trị ô nhập liệu

```
element.clear()
```

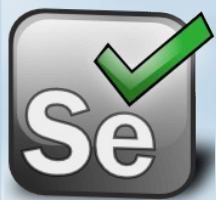
Giả lập bấm phím

```
from selenium.webdriver.common.keys import Keys
```

```
element.send_keys(Keys.RETURN)
```

Submit form

```
element.submit()
```



Tương tác với HTML DOM

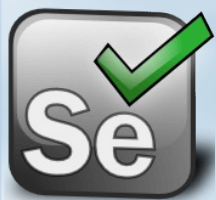
Tương tác Select Box

```
from selenium import webdriver
from selenium.webdriver.support.ui import Select

driver = webdriver.Chrome(executable_path="...")
driver.get("https://www.facebook.com/")

select = Select(driver.find_element_by_id("day"))
# Chọn option dựa trên chỉ số của nó trong Select
select.select_by_index(1)
# Chọn option dựa trên giá trị value
select.select_by_value("15")
# Chọn option dựa trên văn bản hiển thị của nó
select.select_by_visible_text("16")

driver.close()
```

Tương tác với HTML DOM

Lấy giá trị thuộc tính trong thành phần HTML

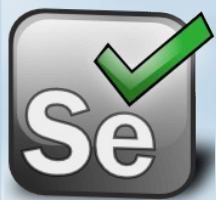
```
from selenium import webdriver

driver = webdriver.Chrome(executable_path="...")
driver.get("https://www.google.com.vn/")

element = driver.find_element_by_name("q")
print(element.get_attribute("title"))
print(element.get_attribute("maxlength"))
```

Lấy giá trị thuộc tính CSS

```
print(element.value_of_css_property("height"))
print(element.value_of_css_property("color"))
```



Tương tác với HTML DOM

Lấy thông tin một số thuộc tính khác

```
# Vị trí thành phần HTML trên trang web
print(element.location)
# Lấy tên thẻ
print(element.tag_name)
# Lấy chiều rộng, chiều cao của thành phần HTML
print(element.size)
# Kiểm tra thành phần có đang hiển thị không
print(element.is_displayed())
# Kiểm tra thành phần có đang enable
print(element.is_enabled())
# Kiểm tra radio/checkbox có đang được chọn không
print(element.is_selected())
```

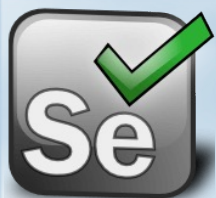



Wait

Selenium WebDriver cung cấp hai loại chờ: implicit và explicit.

Explicit Wait: WebDriver chờ một điều kiện thỏa mãn trước khi thực hiện một việc nào đó.

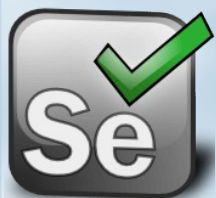
Implicit Wait: WebDriver chờ một thành phần DOM trong khoảng thời gian nào đó để thử lấy nó.



Implicit Wait

Ví dụ Implicit Wait

```
from selenium import webdriver  
  
driver = webdriver.Chrome(executable_path="venv/chromedriver")  
driver.implicitly_wait(10)  
driver.get("https://www.google.com.vn/")
```

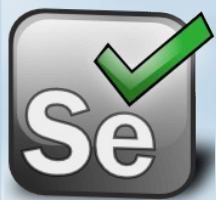
Explicit Wait

Ví dụ Explicit Wait

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as ec

driver = webdriver.Chrome(executable_path="venv/chromedriver")
driver.get("...")

try:
    element = WebDriverWait(driver, 10).until(
        ec.presence_of_element_located((By.ID, "username"))
    )
finally:
    driver.quit()
```



Explicit Wait

Mặc định WebDriver sẽ kiểm tra điều kiện mong muốn (ExpectedCondition) mỗi 500 mili giây để tìm thành phần mong muốn.

Nếu sau khoảng thời gian chờ thành phần cần tìm chưa có thì ngoại lệ TimeoutException sẽ được ném ra.



Q&A