

THAM KHẢO NHANH SELENIUM WEBDRIVER

Tự động hóa trình duyệt, tương tác phần tử, chờ và kiểm tra

Cài Đặt

Cài Đặt

```
pip install selenium webdriver-manager
# webdriver-manager auto-downloads browser drivers
```

Khởi Tạo Driver Cơ Bản

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
driver = webdriver.Chrome(
    service=Service(ChromeDriverManager().install()))
```

Chế Độ Headless

```
options = webdriver.ChromeOptions()
options.add_argument("--headless=new")
options.add_argument("--no-sandbox")
driver = webdriver.Chrome(options=options)
```

Trình Duyệt Được Hỗ Trợ

```
webdriver.Chrome() Google Chrome / Chromium
webdriver.Firefox() Mozilla Firefox (GeckoDriver)
webdriver.Edge() Microsoft Edge (Chromium)
webdriver.Safari() Apple Safari (chỉ macOS)
```

Trình Duyệt & Điều Hướng

Điều Hướng

```
driver.get("https://example.com")
driver.back() # browser back
driver.forward() # browser forward
driver.refresh() # reload page
```

Thuộc Tính Trình Duyệt

```
driver.title Tiêu đề trang hiện tại
driver.current_url URL trang hiện tại
driver.page_source Mã nguồn HTML đầy đủ trang
driver.get_cookies() Liệt kê tất cả cookie
```

Quản Lý Cửa Sổ

```
driver.set_window_size(1920, 1080)
driver.maximize_window()
driver.minimize_window()
driver.quit() # close all windows, end session
```

Tìm Phần Tử

Chiến Lược Locator

```
from selenium.webdriver.common.by import By
driver.find_element(By.ID, "login-btn")
driver.find_element(By.CLASS_NAME, "nav-item")
driver.find_element(By.CSS_SELECTOR, "div.card > h2")
driver.find_element(By.XPATH, "//input[@name='q']")
```

Chiến Lược By

```
By.ID Khớp thuộc tính id phần tử
By.NAME Khớp thuộc tính name phần tử
By.CLASS_NAME Khớp lớp CSS (một lớp)
By.TAG_NAME Khớp tên thẻ HTML
By.CSS_SELECTOR CSS selector (linh hoạt nhất)
By.XPATH Biểu thức XPath
By.LINK_TEXT Văn bản anchor chính xác
By.PARTIAL_LINK_TEXT Khớp một phần văn bản anchor
```

Tìm Nhiều Phần Tử

```
items = driver.find_elements(By.CSS_SELECTOR, "li.item")
for item in items:
    print(item.text)
# Returns empty list if none found (no exception)
```

Tương Tác

Click & Gõ

```
elem = driver.find_element(By.ID, "search")
elem.clear() # clear existing text
elem.send_keys("selenium python")
elem.submit() # submit parent form
```

Dropdown

```
from selenium.webdriver.support.ui import Select
select = Select(driver.find_element(By.ID, "country"))
select.select_by_visible_text("Canada")
select.select_by_value("ca")
select.select_by_index(2)
```

Thuộc Tính Phần Tử

```
•.text Nội dung văn bản hiển thị
•.get_attribute('href') Giá trị thuộc tính HTML
•.is_displayed() True nếu phần tử hiển thị
•.is_enabled() True nếu phần tử tương tác được
•.is_selected() True nếu checkbox/radio được chọn
•.tag_name Thẻ HTML (ví dụ 'input', 'div')
•.value_of_css_property('color') Giá trị thuộc tính CSS đã tính toán
```

Chờ

Chờ Tương Minh

```
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
elem = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.ID, "result")))
```

Điều Kiện Mong Đợi

```
presence_of_element_located Phần tử tồn tại trong DOM
visibility_of_element_located Phần tử hiển thị trên trang
element_to_be_clickable Phần tử hiển thị và được kích hoạt
```

```
text_to_be_present_in_element Phần tử chứa văn bản mong đợi
```

```
alert_is_present Alert JavaScript đang hiển thị
```

```
staleness_of Phần tử không còn trong DOM
```

title_contains Tiêu đề trang chứa văn bản

Chờ Ngầm Định

```
driver.implicitly_wait(10) # seconds, applies globally
# Explicit waits are preferred - more precise control
```

Frame & Cửa Sổ

Frame

```
driver.switch_to.frame("frame-name") # by name/id
driver.switch_to.frame(0) # by index
driver.switch_to.frame(elem) # by element
driver.switch_to.default_content() # back to main
```

Cửa Sổ & Tab

```
original = driver.current_window_handle
driver.switch_to.new_window("tab") # open new tab
driver.switch_to.window(original) # switch back
driver.close() # close current tab
```

Alert

```
alert = driver.switch_to.alert
print(alert.text)
alert.accept() # click OK
alert.dismiss() # click Cancel
alert.send_keys("input text")
```

Ảnh Chụp Màn Hình

Chụp Ảnh Màn Hình

```
driver.save_screenshot("page.png") # full page
elem = driver.find_element(By.ID, "chart")
elem.screenshot("chart.png") # single element
```

Ảnh Chụp Dạng Base64

```
b64 = driver.get_screenshot_as_base64()
png = driver.get_screenshot_as_png() # bytes
```

Actions

Chuỗi Action

```
from selenium.webdriver.common.action_chains import ActionChains
actions = ActionChains(driver)
actions.move_to_element(menu).click().perform()
```

Action Bàn Phím

```
from selenium.webdriver.common.keys import Keys
elem.send_keys(Keys.ENTER)
elem.send_keys(Keys.CONTROL, "a") # select all
actions.key_down(Keys.SHIFT).click(elem).perform()
```

Action Chuột

```
•.click(elem) Click phần tử
•.double_click(elem) Double-click phần tử
•.context_click(elem) Click chuột phải phần tử
•.move_to_element(elem) Hover qua phần tử
•.drag_and_drop(src, dst) Kéo nguồn đến đích
•.click_and_hold(elem) Nhấn và giữ nút chuột
•.release() Thả nút chuột
```

Kiểm Tra

Kiểm Tra Phổ Biến (pytest)

```
assert "Dashboard" in driver.title
assert driver.find_element(By.ID, "msg").text == "Done"
assert driver.current_url.endswith("/home")
assert len(driver.find_elements(By.CSS_SELECTOR, "tr")) > 0
```

Kiểm Tra Dựa Trên Chờ

```
WebDriverWait(driver, 5).until(
    EC.visibility_of_element_located((By.ID, "success")))
WebDriverWait(driver, 5).until(
    EC.invisibility_of_element_located((By.ID, "spinner")))
```

Thực Thi JavaScript

```
result = driver.execute_script("return document.title")
driver.execute_script(
    "arguments[0].scrollIntoView(true);", elem)
```

Mẫu Phổ Biến

Mẫu Page Object

```
class LoginPage:
    URL = "/login"
    user_loc = (By.ID, "username")
    def login(self, drv, user, pwd):
        drv.find_element(*self.user_loc).send_keys(user)
```

Context Manager

```
from selenium import webdriver
with webdriver.Chrome() as driver:
    driver.get("https://example.com")
    print(driver.title)
# driver.quit() called automatically
```

Retry & Dọn Dẹp

```
try:
    driver.get("https://example.com")
    WebDriverWait(driver, 10).until(
        EC.element_to_be_clickable((By.ID, "btn")))
finally: driver.quit()
```