

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐÔNG Á**

🙠🙡🕮🙡🙠



BÁO CÁO THỰC HÀNH

**KIỂM THỬ PHẦN MỀM**

**Đề tài:**

**Xây dựng Selenium framework theo mô hình**

**Page Object Model (POM)**

**GVHD: Tạ Quốc Ý**

**SVTH: Đỗ Thái Dương**

**Lớp: ST19A1B**

*Đà Nẵng, ngày 22 tháng 04 năm 2022*

MỤC LỤC

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 3](#_Toc102002214)

[1.1. Python là gì? 3](#_Toc102002215)

[1.2. Selenium là gì? 4](#_Toc102002216)

[1.3. Tại sao lại sử dụng Selenium Python? 4](#_Toc102002217)

[1.4. Downloading Python bindings cho selenium: 4](#_Toc102002218)

[1.5. Một số thư viện cần thiết khác đồng hành với Selenium: 5](#_Toc102002219)

[1.5.1. html-testRunner: 5](#_Toc102002220)

[1.5.2. autopep8: 5](#_Toc102002221)

[1.6. Ưu điểm của POM là gì? 6](#_Toc102002222)

[CHƯƠNG 2: THỰC HÀNH 6](#_Toc102002223)

[2.1. Yêu cầu ban đầu: 6](#_Toc102002224)

[2.1.1. Cài đặt Python, Selenium, html-testRunner, autopep8: 6](#_Toc102002225)

[2.1.2. Có kiến thức cơ bản với Selenium: 7](#_Toc102002226)

[2.1.3. Trang web phục vụ Automation Testing: 7](#_Toc102002227)

[2.2. Cấu trúc thư mục dự án: 8](#_Toc102002228)

[2.3. Các test modules và test cases: 9](#_Toc102002229)

[2.3.1. login\_logout: 9](#_Toc102002230)

[2.3.1.1. Login: 9](#_Toc102002231)

[2.3.1.2. Logout: 9](#_Toc102002232)

[2.3.2. homepage: 10](#_Toc102002233)

[2.3.2.1. test\_homepage: 10](#_Toc102002234)

[2.4. Chạy scriptest và thu nhận kết quả: 10](#_Toc102002235)

[2.5. Testcase Reports: 11](#_Toc102002236)

[CHƯƠNG 3: KẾT LUẬN 11](#_Toc102002237)

[TÀI LIỆU THAM KHẢO 11](#_Toc102002238)

# CƠ SỞ LÝ THUYẾT

*Chúng ta sẽ tìm hiểu lý thuyết của Framework và ngôn ngữ được chọn để thực hiện đồ án lần này:* ***Selenium****,* ***Python****.*

## Python là gì?

Python là một ngôn ngữ lập trình bậc cao, thông dịch, hướng đối tượng, đa mục đích và cũng là một ngôn ngữ lập trình động. Cú pháp của Python là khá dễ dàng để học và ngôn ngữ này cũng mạnh mẽ và linh hoạt không kém các ngôn ngữ khác trong việc phát triển các ứng dụng, nó được thiết kế một cách thân thiện, python sử dụng từ khóa bằng tiếng Anh căn bản để dễ dàng diễn tả.

Cài đặt Python là việc cần thiết đầu tiên nên ưu tiên để có thể tiến hành automation testing với các framework như Selenium, hướng dẫn cài đặt [tại đây](https://viettuts.vn/python/cai-dat-python):

## Selenium là gì?

Selenium là một trong những công cụ kiểm thử phần mềm tự động mã nguồn mở (open source test automation tool) mạnh nhất hiện nay cho việc kiểm thử ứng dụng Web. Selenium script có thể chạy được trên hầu hết các trình duyệt như IE, Mozilla FireFox, Chrome, Safari, Opera; và hầu hết các hệ điều hành như Windows, Mac, Linux.

Selenium WebDriver (Selenium 2): là thư viện cho phép người dung lập trình (scripting) test script trên các ngôn ngữ lập trình khác nhau như Python, Java, C#, Ruby…

## Tại sao lại sử dụng Selenium Python?

Selenium Python bindings cung cấp một API đơn giảm để viết functional/acceptance test sử dụng selenium webdriver.

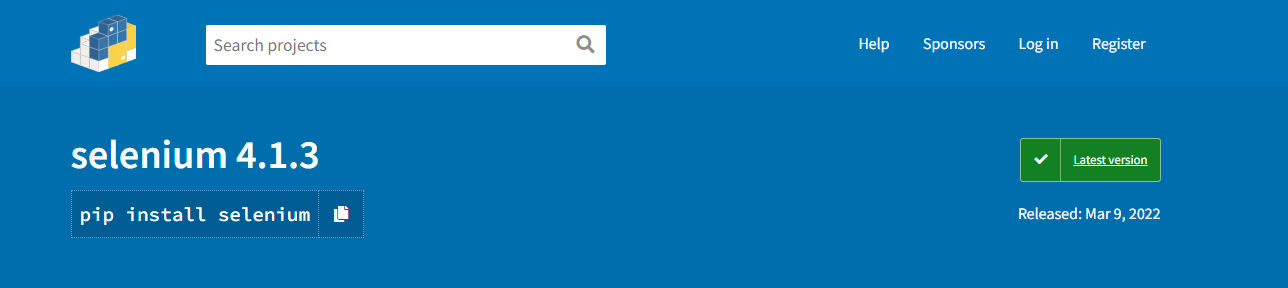
Thông qua Selenium Python API bạn có thể truy cập tất cả các chức năng của selenium webdriver một cách trực quan.

Selenium Python bindings cung cấp một API thuận tiện để truy cập Webdrivers như Firefox, IE, Chrome, Remote. Hiện tại hỗ trợ Python version 2.7,3.2.

## Downloading Python bindings cho selenium:

Bạn có thể download Python bindings cho Selenium từ PyPI page cho gói selenium. Tuy nhiên một cách tiếp cận tốt hơn sẽ là sử dụng pip để install gói selenium.

Python 3.5 có sẵn pip trong thư viện chuẩn. Sử dụng pip bạn có thể cài đặt selenium trên trang chủ của PyPi như sau:

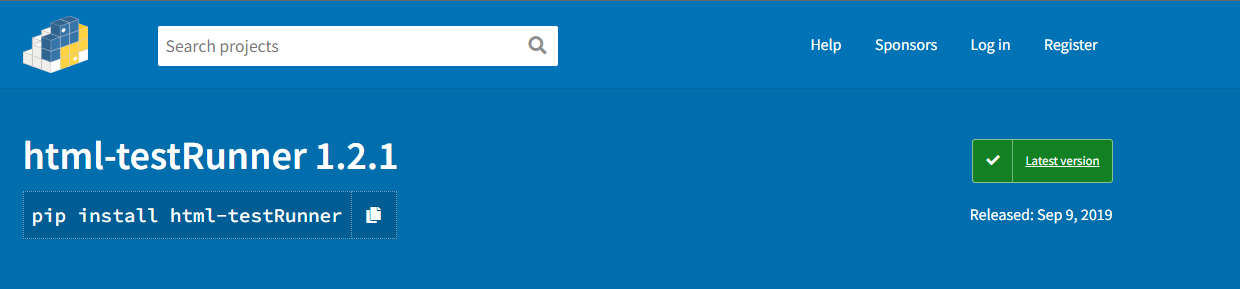


Bạn có thể nghiên cứu sử dụng virtualenv để tạo môi trường cô lập cho python. Python 3.5 có pyvenv nó hầu hết giống như virtualenv.

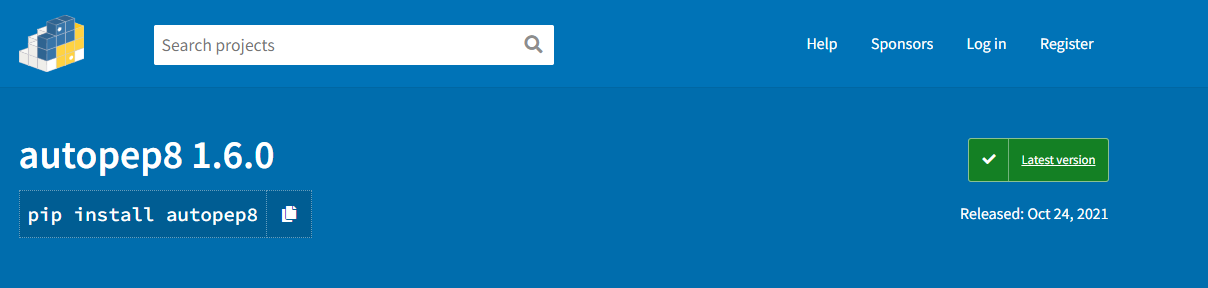
Hướng dẫn cài đặt chi tiết [tại đây](https://nguyenvanhieu.vn/cai-dat-moi-truong-selenium-voi-python/)

## Một số thư viện cần thiết khác đồng hành với Selenium:

### html-testRunner:

Thư viện giúp tạo các báo cáo testing dưới dạng html. Chúng ta có thể dễ dàng xem các báo cáo testing success hay fail chỉ với vài thao tác, cài đặt trên trang trủ PyPi:

### autopep8:

**autopep8** tự động định dạng mã Python để phù hợp với hướng dẫn kiểu PEP 8. Nó sử dụng tiện ích pycodestyle để xác định những phần nào của mã cần được định dạng. autopep8 có khả năng khắc phục hầu hết các vấn đề định dạng có thể được báo cáo bởi pycodestyle, cài đặt trên trang trủ PyPi:

## Ưu điểm của POM là gì?

Trong phần trên, chúng ta đã hiểu tại sao một dự án không thuộc POM lại có thể khiến dự án của chúng tôi khó bảo trì, điều này trở thành một lợi thế rất lớn của POM. Dưới đây là một số ưu điểm của việc sử dụng Mô hình Đối tượng Trang làm nổi bật hiệu quả của nó:

Làm cho mã có thể bảo trì - Vì các lớp thử nghiệm tách biệt với các lớp chứa các phần tử web và hoạt động trên chúng, nên việc cập nhật mã rất dễ dàng nếu bất kỳ phần tử web nào cập nhật hoặc một phần tử mới được thêm vào.

Làm cho mã có thể đọc được- Người dùng có thể dễ dàng đọc qua dự án và kịch bản thử nghiệm do có một ranh giới nhỏ giữa các lớp thử nghiệm và các trang web khác nhau.

Và, làm cho mã có thể tái sử dụng- Nếu nhiều tập lệnh thử nghiệm sử dụng các phần tử web giống nhau, thì chúng ta không cần viết mã để xử lý phần tử web trong mọi tập lệnh thử nghiệm. Đặt nó trong một lớp trang riêng biệt sẽ làm cho nó có thể sử dụng lại được bằng cách làm cho nó có thể truy cập được bằng bất kỳ tập lệnh thử nghiệm nào.

# THỰC HÀNH

## Yêu cầu ban đầu:

*Tiến hành cài đặt các thư viện và các yêu cầu đầu vào trước khi xây dựng project testing theo mô hình POM.*

### Cài đặt Python, Selenium, html-testRunner, autopep8:

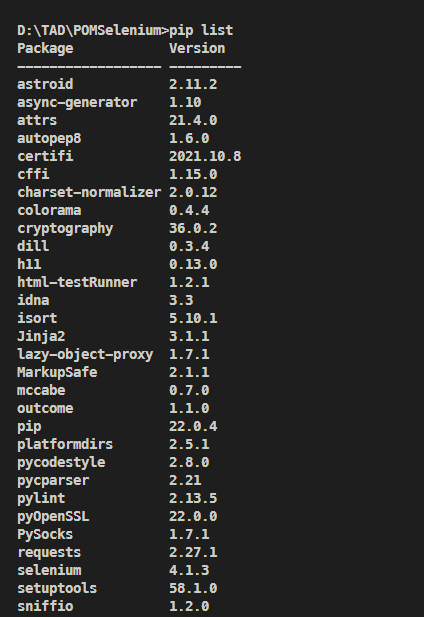
Đã tiến hành giải thích ở chương 1: CƠ SỞ LÝ THUYẾT

Kiểm tra xem các thư viện là được cài đặt chưa bằng lệnh:

pip list

Kết quả:

Nếu đã cài đặt thành công, các thư viện sẽ hiển thị trong danh sách như hình dưới:



### Có kiến thức cơ bản với Selenium:

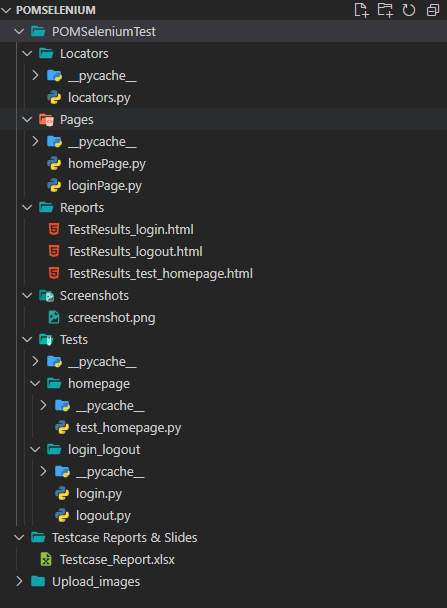
Để tiến hành xây dựng một test module hoàn chỉnh, việc có kiến thức cơ bản với Selenium là điều cần thiết, chúng ta có thể tìm hiểu cơ bản về Selenium thông qua trang docs chính thức của Selenium [tại đây](https://selenium-python.readthedocs.io/):

### Trang web phục vụ Automation Testing:

Phần không thể thiếu khi thực hiện automation testing, đây là nơi các script testing được thực thi và cho ra kết quả cho người dùng. Ở đây, chúng ta sử dụng một trang mã nguồn mở với địa chỉ: <https://opensource-demo.orangehrmlive.com/> để phục vụ cho việc testing.

## Cấu trúc thư mục dự án:

Theo như tên đề tài **Xây dựng Selenium framework theo mô hình Page Object Model (POM),** để xây dựng Việc Automation Testing hoàn chỉnh theo mô hình trên, hiểu được cấu trúc dự án là điều quan trọng. Chúng ta có thể tiến hành xây dựng cấu trúc của dự án theo hướng như sau (tuỳ chọn):



Trong đó:

Locations: như là một nơi chứa các Object Repository cho việc Testing trở nên thuận tiện hơn, thường bao gồm các địa chỉ id, class, hay xPath của một element nào đó trên cây DOM.

Pages**:** nơi chứa các view và các phương thức chính hay chức năng chính cho object, nơi ta có thể tách biệt view và logic script, tiện lợi cho việc maintain sau này.

Reports: nơi chứa các testcase result sau khi việc testing automation hoàn thành, được lưu dưới dạng html thông qua thư viện html-testRunner, có thể dễ dàng view trên browser.

Screenshots: nơi chứa các screenshots sau khi testing automation hoàn thành, được người dùng định nghĩa vị trí và thời điểm để take screenshots ở các màn hình testing nhất định.

Tests: thư mục quan trọng nhất trong cây dự án, nơi chứa các script test để chạy các testcase cho dự án, thường là automation testing. Số lượng và thời lượng các testcase được chạy và đưa ra result phụ thuộc vào người viết định nghĩa.

Testcase Reports & Slides (tuỳ chọn): báo cáo kết quả testing theo dạng Excel.

Upload\_images (tuỳ chọn): thư mục chứa các hình ảnh dùng cho việc testing các testcase liên quan đến việc tải ảnh lên.

Bạn có thể download source code để thực hành thông qua link Github sau:

<https://github.com/tduongtad1304/POM-Selenium>

## Các test modules và test cases:

### login\_logout:

<https://github.com/tduongtad1304/POM-Selenium/tree/master/POMSeleniumTest/Tests/login_logout>

Gồm các test cases:

#### Login:

<https://github.com/tduongtad1304/POM-Selenium/blob/master/POMSeleniumTest/Tests/login_logout/login.py>

#### Logout:

<https://github.com/tduongtad1304/POM-Selenium/blob/master/POMSeleniumTest/Tests/login_logout/logout.py>

### homepage:

<https://github.com/tduongtad1304/POM-Selenium/tree/master/POMSeleniumTest/Tests/homepage>

Gồm các test cases:

#### test\_homepage:

<https://github.com/tduongtad1304/POM-Selenium/blob/master/POMSeleniumTest/Tests/homepage/test_homepage.py>

## Chạy scriptest và thu nhận kết quả:

Để thực hiện thu nghiệm kết quả, sau khi đã có source code, tiến hành chạy một Terminal mới và chạy dòng script như hình (Ví dụ cho Visual Studio Code):

Testing login:

python -m POMSeleniumTest.Tests.login\_logout.login

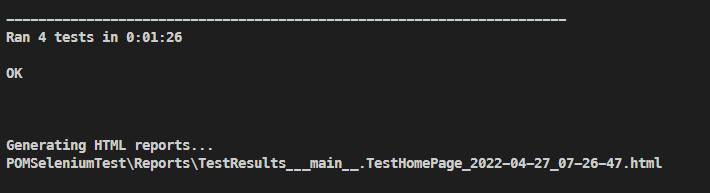
Testing logout:

python -m POMSeleniumTest.Tests.login\_logout.logout

Testing homepage:

python -m POMSeleniumTest.Tests.homepage.test\_homepage

Kết quả thu được trong Terminal (ví dụ mẫu với test module test\_homepage):



## Testcase Reports:

Dưới đây là báo cáo kết quả các testcase mẫu đã thu được: (có thể download qua link [tại đây](https://github.com/tduongtad1304/POM-Selenium/raw/master/Testcase%20Reports%20%26%20Slides/Testcase_Report.xlsx))

# KẾT LUẬN

Chúng ta đã hiểu mẫu thiết kế theo mô hình POM là gì và cách nó nâng cao các tập lệnh thử nghiệm.

Ngoài ra, chúng ta đã thấy những lợi thế khác nhau của việc sử dụng. Khung mô hình đối tượng trang (POM) so với cách tiếp cận thông thường. Đặc biệt là lợi ích của nó là cải thiện khả năng đọc mã và giảm các bản sao.

Hơn nữa, sau đó chúng ta có thể làm cho khung thử nghiệm của mình hiệu quả bằng cách sử dụng nhiều lớp trang cho các trang web khác nhau của ứng dụng web. Các lớp trang này bao gồm các phần tử có trên các trang tương ứng và các phương pháp hành động có liên quan của chúng.

Ngoài ra, chúng tađã tạo một lớp thử nghiệm sử dụng các đối tượng được tạo trong các lớp trang. Ngoài ra, nó giữ kịch bản thử nghiệm tách biệt với kho lưu trữ đối tượng.

# TÀI LIỆU THAM KHẢO

" Selenium Webdriver with Python: Tutorial with Example," <http://www.guru99.com/selenium-python.html/>

"Getting Started," <https://selenium-python.readthedocs.io/getting-started.html/>

 " Page Object Model," <https://www.toolsqa.com/selenium-webdriver/page-object-model/>

" TEST AUTOMATION HƯỚNG DẪN CƠ BẢN SELENIUM TRONG PYTHON," <https://co-well.vn/nhat-ky-cong-nghe/test-automation-huong-dan-co-ban-selenium-trong-python/>

- **Hết** -