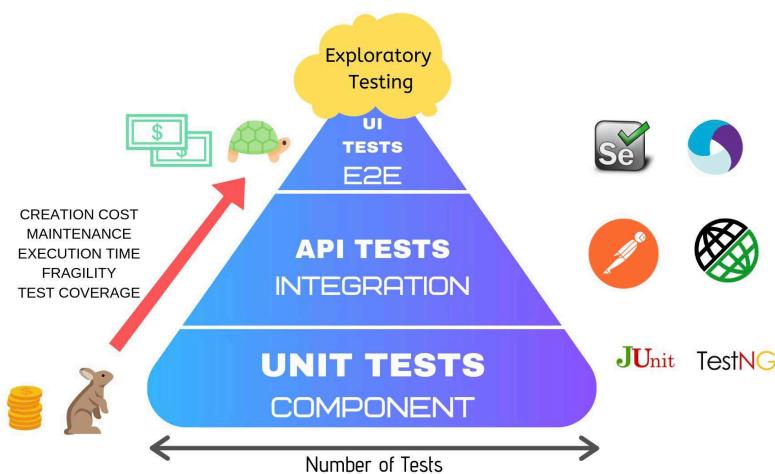


PART 2 - Web Knownledge/ HTML/ Locator/ XPath/ Css Selector

I. AUTOMATION TESTING LEVEL



1. Unit Tests (Component)

- Kiểm thử từng hàm hoặc chức năng nhỏ trong hệ thống.

2. API Tests (Integration)

- Kiểm tra việc giao tiếp giữa các phần của hệ thống thông qua API.

3. UI Tests (E2E)

- Mô phỏng thao tác của người dùng thật trên giao diện hệ thống.

4. Exploratory Testing

- Kiểm thử không theo kịch bản cố định, tập trung vào việc tự do khám phá và tìm lỗi dựa trên kinh nghiệm người test.

II. LOCATOR

- Locator là phương pháp được sử dụng để tìm kiếm và tương tác với các phần tử HTML trên trang web.
- Độ ưu tiên:

Thứ tự ưu tiên	Tiêu chí	Mô tả chi tiết
1 - Unique	Duy nhất	Giá trị của attribute phải là duy nhất (không bị trùng).
2 - Meaning	Có ý nghĩa	Attribute có liên quan đến nghiệp vụ hoặc chức năng (VD: <code>id</code> , <code>class</code> , <code>name</code> có ý nghĩa).
3 - Attribute	<code>id / class / name</code>	Ưu tiên sử dụng 3 loại attribute này vì chúng thường chạy nhanh và ổn định.
4 - Attribute khác	Không thuộc <code>id/class/name</code>	Dùng bất kỳ attribute nào khác nếu đảm bảo duy nhất (1) và có ý nghĩa (2).
5 - Link	Hạn chế dùng thẻ <code>a[href]</code>	Tránh dùng <code>a[href]</code> – nên dùng theo text hoặc title. Phù hợp khi chạy trên nhiều môi trường/server khác nhau.

- Một số Locator:

Locator	Mô tả
ID	Tìm phần tử theo thuộc tính <code>id</code> .
Name	Tìm phần tử theo thuộc tính <code>name</code> .
Class Name	Tìm phần tử theo tên class.
Tag Name	Tìm phần tử theo tên thẻ HTML.
Link Text	Tìm phần tử <code><a></code> theo văn bản liên kết.
Partial Link Text	Tìm phần tử <code><a></code> theo phần văn bản liên kết.
CSS Selector	Tìm phần tử dựa trên cú pháp CSS.
XPath	Tìm phần tử dựa trên cú pháp XPath, linh hoạt nhất.

- Lưu ý:

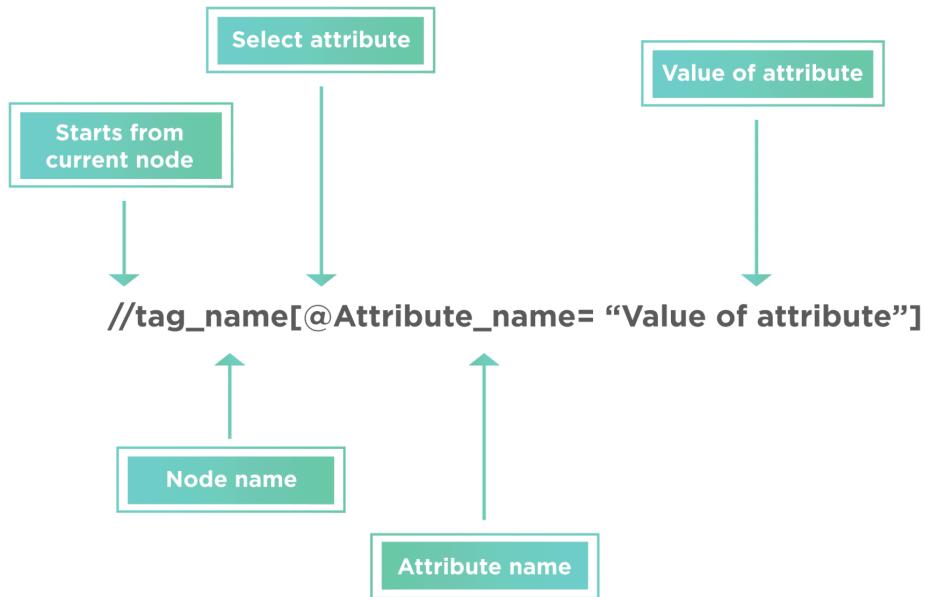
Không nên dùng	Lý do
Sử dụng <code>*</code> trong tag name: <code>//*[@id='login_username']</code>	Gây chậm, khó đọc, dễ gây hiểu nhầm.

Không nên dùng	Lý do
Attribute value nằm trong dấu nháy đôi: <code>//input[@id="login_username"]</code>	Làm code xấu, dễ lỗi khi chèn vào chuỗi.

III. XPATH

1. Xpath

- XPath (XML Path Language) là ngôn ngữ truy vấn giúp chọn và thao tác với các phần tử (element) trong tài liệu HTML/XML.
- Trong kiểm thử tự động (Selenium, Playwright...), XPath giúp bạn xác định chính xác phần tử trên trang web để tương tác (click, nhập, kiểm tra...).
- Cú pháp:



- Giải thích:

- `//` → chọn bất kỳ vị trí nào trên tài liệu.
- `/` → chọn nút con trực tiếp từ cha (không nhảy cấp)
- `tagname` → tên thẻ HTML (ví dụ `input`, `button`, `div`...).

- `[@attribute='value']` → tìm thẻ có thuộc tính (attribute) bằng với giá trị mình chỉ định.

2. Phân loại XPath

Loại	Định nghĩa	Cú pháp ví dụ	Ưu/Nhược
XPath tuyệt đối	Bắt đầu từ gốc <code>html</code> , đi qua từng thẻ một theo thứ tự	<code>/html/body/div[2]/form/input</code>	Ưu: chính xác. Nhược: dễ gây nếu thay đổi UI
XPath tương đối	Bắt đầu từ bất kỳ vị trí nào, tìm đến phần tử theo điều kiện	<code>//form[@id='loginForm']/input</code>	Ưu: ngắn gọn, linh hoạt. Nhược: chạy chậm hơn chút

3. Cấu trúc XPath cơ bản

Mục tiêu	Ý nghĩa	XPath ví dụ
Tất cả thẻ <code>input</code>	Chọn tất cả thẻ <code>input</code>	<code>//input</code>
Theo thuộc tính	Thẻ <code>input</code> có <code>type=email</code>	<code>//input[@type='email']</code>
Theo ID bất kỳ	Bất kỳ thẻ nào có ID là <code>loginBtn</code>	<code>//*[@id='loginBtn']</code>
Tìm con trong cha	Tìm <code>input</code> là con trực tiếp của <code>form</code>	<code>//form/input</code>
Tìm con ở bất kỳ cấp	Tìm <code>input</code> nằm trong <code>div.box</code> , có thể là cháu/chắt	<code>//div[@class='box']//input</code>
Theo vị trí	Input thứ 2 trong toàn trang	<code>(//input)[2]</code>

4. Hàm XPath thông dụng

Hàm	Ý nghĩa	Ví dụ
<code>contains()</code>	Chứa một đoạn trong attribute	<code>//a[contains(@href, 'login')]</code>
<code>starts-with()</code>	Bắt đầu bằng chuỗi	<code>//input[starts-with(@name, 'user')]</code>
<code>text()</code>	Lấy nội dung text trực tiếp, không bao gồm text từ các phần tử con	<code>//button[text()='Đăng nhập']</code>
<code>contains(text(), 'Hello')</code>	Text trực tiếp có chứa chuỗi	<code>//p[contains(text(),'Hello')]</code>

Hàm	Ý nghĩa	Ví dụ
contains(., 'Hello')	Text nằm trực tiếp hoặc trong child node	//div[contains(., 'Hello')]
contains(string(), 'Hello')	Tương tự ., lấy toàn bộ chuỗi văn bản hiển thị	//span[contains(string(), 'Hello')]
not()	Chọn phần tử không chứa text	//div[not(contains(text(), 'Hello'))]
concat()	Nối chuỗi (hiếm dùng trong thực tế)	concat('Hello', ' ', 'World')

5. Hàm định vị theo vị trí

Hàm	Ý nghĩa	Ví dụ
position()	Phần tử đầu tiên	//li[position()=1]
last()	Phần tử cuối cùng	//li[last()]
last()-1	Phần tử áp chót	//div[last()-1]

6. XPath Axes – định vị theo quan hệ

- XPath Axes là cách để bạn tìm các phần tử HTML (node) dựa trên mối quan hệ với phần tử hiện tại.
- **Ghi nhớ:** XPath luôn xét từ node hiện tại, từ đó di chuyển theo hướng được chỉ định bởi axis để tìm các node phù hợp.

Axis	Mô tả	Ví dụ
parent::	Lấy node cha trực tiếp	//input/parent::*
ancestor::	Tất cả tổ tiên (cha, ông, cụ...)	//div/ancestor::*
child::	Con trực tiếp	//ul/child::li
descendant::	Con/cháu ở mọi cấp	//div/descendant::input
preceding::	Node đứng trước hiện tại (không cần đồng cấp)	//h2/preceding::*
following::	Node đứng sau	//h2/following::*
preceding-sibling::	Các anh cùng cấp	//li/preceding-sibling::*
following-sibling::	Các em cùng cấp	//li/following-sibling::*

7. Phép logic trong Xpath

Phép toán	Ý nghĩa / Ứng dụng	Cú pháp ví dụ
and	Cả 2 điều kiện đúng mới chọn – lọc chính xác phần tử	//input[@type='text' and @name='email']
or	Chỉ cần 1 trong 2 điều kiện đúng là được – mở rộng phạm vi chọn	//input[@type='text' or @type='password']
not()	Loại bỏ phần tử có điều kiện đó – chọn phần tử không thỏa điều kiện	//input[not(@type='submit')]
and + not	Kết hợp điều kiện & phủ định – chọn div có class là box nhưng không chứa chữ "Ãn"	//div[@class='box' and not(contains(text(), 'Ãn'))]

IV. CSS

1. CSS

- CSS Selector là cú pháp được dùng để chọn phần tử HTML dựa trên ID, class, thuộc tính, quan hệ vị trí...
- Trong Selenium/Playwright, bạn dùng CSS Selector để tìm phần tử khi viết test.
- Không đi ngược.

2. Các loại CSS Selector cơ bản

Kiểu chọn	Cú pháp	Ý nghĩa	Ví dụ
Theo ID	#id	Chọn phần tử có id="loginBtn"	#loginBtn
Theo class	.class	Chọn phần tử có class	.form-control
Theo tag	tag	Chọn tất cả các thẻ đó	input , div
Tag + class	tag.class	input có class input-text	input.input-text
Tag + ID	tag#id	button có id là submitBtn	button#submitBtn

Kiểu chọn	Cú pháp	Ý nghĩa	Ví dụ
Nhiều class	.box.rounded	Phần tử có cả 2 class <code>box</code> và <code>rounded</code>	

3. CSS Selector theo thuộc tính

Mục tiêu	Cú pháp	Ví dụ
Có thuộc tính	tag[attribute]	input[name]
Thuộc tính bằng	tag[attr='value']	input[type='email']
Thuộc tính bắt đầu bằng	tag[attr^='val']	input[name^='user']
Thuộc tính kết thúc bằng	tag[attr\$='val']	img[src\$='.png']
Thuộc tính chứa đoạn	tag[attr*= 'val']	a[href*='login']

4. Quan hệ cha – con – anh em

Mối quan hệ	Cú pháp	Ý nghĩa	Ví dụ
Cha > Con trực tiếp	A > B	input là con trực tiếp của form	form > input
Cha → cháu bất kỳ cấp	A B	input là con hoặc cháu của div	div input
Cùng cấp (ngay sau)	A + B	input đứng ngay sau label	label + input
Cùng cấp (sau bất kỳ)	A ~ B	Tất cả p sau div, cùng cấp	div ~ p

5. Theo thứ tự (vị trí)

Cú pháp	Ý nghĩa	Ví dụ
:first-child	Phần tử đầu tiên	ul li:first-child
:last-child	Phần tử cuối	ul li:last-child
:nth-child(n)	Phần tử thứ 2	ul li:nth-child(2)
:nth-last-child(n)	Phần tử cuối cùng (ngược lên)	ul li:nth-last-child(1)

6. Phép logic trong CSS

Loại kết hợp	Ý nghĩa tương tự	Cú pháp ví dụ
AND (và)	Tất cả điều kiện phải đúng: tag + class + attribute	<code>input.form-control[type="text"]</code>
Nhiều class	Có cả class btn và class primary	<code>.btn.primary</code>
Nhiều attribute	input có cả type = text và name = email	<code>input[type="text"][name="email"]</code>
OR (hoặc)	Chọn tất cả input hoặc textarea	<code>input, textarea</code>
NOT (phù định)	Chọn input nhưng loại trừ type = submit	<code>input:not([type="submit"])</code>