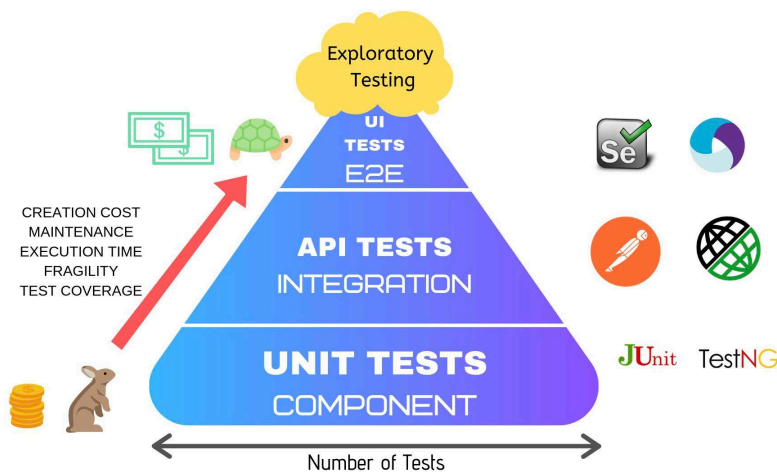


# PART 2 - Web Knowledge/ HTML/ Locator/ XPath/ Css Selector

## 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	id / class / name	Ưu tiên sử dụng 3 loại attribute này vì chúng thường chạy nhanh và ổn định.
4 - Attribute khác	Không thuộc id/class/name	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>&lt;a&gt;</code> theo văn bản liên kết.
Partial Link Text	Tìm phần tử <code>&lt;a&gt;</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:

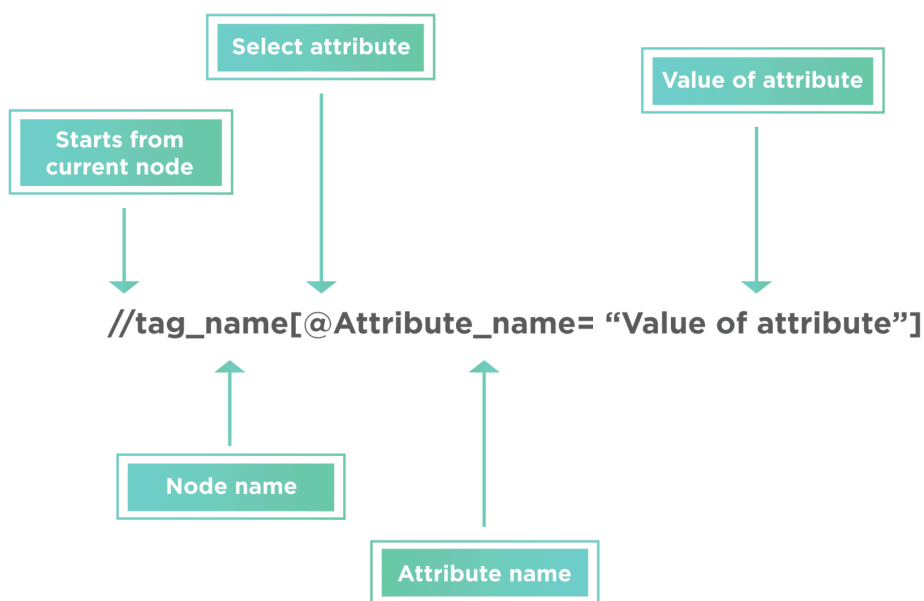
Không nên dùng	Lý do
Sử dụng <code>*</code> trong tag name: <code>//*[@id='login_username']</code>	Gây chậm, khó đọc, dễ gây hiểu nhầm.

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>	<b>Ưu:</b> chính xác. <b>Nhược:</b> dễ gãy 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>	<b>Ưu:</b> ngắn gọn, linh hoạt. <b>Nhược:</b> 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/chất	<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
<code>contains(., 'Hello')</code>	Text nằm trực tiếp hoặc trong child node	<code>//div[contains(., 'Hello')]</code>
<code>contains(string(), 'Hello')</code>	Tương tự <code>.</code> , lấy toàn bộ chuỗi văn bản hiển thị	<code>//span[contains(string(), 'Hello')]</code>
<code>not()</code>	Chọn phần tử không chứa text	<code>//div[not(contains(text(), 'Hello'))]</code>
<code>concat()</code>	Nối chuỗi (hiếm dùng trong thực tế)	<code>concat('Hello', ' ', 'World')</code>

## 5. Hàm định vị theo vị trí

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

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

## 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	<code>//input[@type='text' and @name='email']</code>
or	Chỉ cần 1 trong 2 điều kiện đúng là được – mở rộng phạm vi chọn	<code>//input[@type='text' or @type='password']</code>
not()	Loại bỏ phần tử có điều kiện đó – chọn phần tử không thỏa điều kiện	<code>//input[not(@type='submit')]</code>
and + not	Kết hợp điều kiện & phủ định – chọn div có class là box nhưng không chứa chữ "Ẩn"	<code>//div[@class='box' and not(contains(text(), 'Ẩn'))]</code>

## 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	<code>#id</code>	Chọn phần tử có <code>id="loginBtn"</code>	<code>#loginBtn</code>
Theo class	<code>.class</code>	Chọn phần tử có class	<code>.form-control</code>
Theo tag	<code>tag</code>	Chọn tất cả các thẻ đó	<code>input</code> , <code>div</code>
Tag + class	<code>tag.class</code>	<code>input</code> có class <code>input-text</code>	<code>input.input-text</code>
Tag + ID	<code>tag#id</code>	<code>button</code> có id là <code>submitBtn</code>	<code>button#submitBtn</code>

Kiểu chọn	Cú pháp	Ý nghĩa	Ví dụ
Nhiều class	<code>.box.rounded</code>	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	<code>tag[attribute]</code>	<code>input[name]</code>
Thuộc tính bằng	<code>tag[attr='value']</code>	<code>input[type='email']</code>
Thuộc tính bắt đầu bằng	<code>tag[attr^='val']</code>	<code>input[name^='user']</code>
Thuộc tính kết thúc bằng	<code>tag[attr\$='val']</code>	<code>img[src\$='.png']</code>
Thuộc tính chứa đoạn	<code>tag[attr*='val']</code>	<code>a[href*='login']</code>

### 4. Quan hệ cha – con – anh em

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

### 5. Theo thứ tự (vị trí)

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

### 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 (phủ định)	Chọn input nhưng loại trừ type = submit	<code>input:not([type="submit"])</code>