1.

Watch it in action

//a[text()='Watch it in action']

Home

//a[text()='Home']

+Documentation

//div[@class='masthead']//a[text()='Documentation']

Downloads

//div[@class='masthead']//a[text()='Documentation']

Contact

//div[@class='masthead']//a[text()='Contact']

2.

## **Sibling**

Chúng ra có thể sử dụng sibling để tìm phần tử có liên quan đến phần tử khác

(//\*[text()='Learn more']/following-sibling::\*)[1]

## **Ancestor:**

Tìm phần tử dựa vào phần tử cha, chúng ta có thể sử dụng tổ tiên của xpath

### **Following**

hàm này sẽ trả về phần tử tiếp theo của phần tử hiện tại

### **Preceding**

Hàm này sẽ trả về những phần tử ở trước phần tử hiện tại

Following-sibling

 để xác định các phần tử trong Selenium. Nó xác định anh chị em tiếp theo của nút đó . Các anh chị em phải được đặt ở mức bằng nhau của nút hiện có và phải có cùng một cha mẹ.

Preceding-sibling

chọn / duyệt qua tất cả các  thẻ ****anh em**** trong tài liệu HTML đứng trước thẻ hiện tại.

\* nhieeud kí tự (liệt kê tất cả các ... gì đó trong ..)

@attribute()=: chứa giá trị tuyệt đối trong attribute

//\*[contains(@name,'btnClick')]

--> Tìm các tags có thuộc tính name chứa đoạn text "btnClick"

//\*[contains(text(),'here')]

--> Tìm các tags có text chứa chữ "here"

//\*[contains(@href,'swtestacademy.com')]

--> Tìm các link có url chứa "swtestacademy.com"

Operator "or"

* ****OR:****sử dụng 2 điều kiện, nếu 1 trong 2 đúng sẽ là đúng. XPath sẽ liệt kê ra những element đáp ứng được ít nhất 1 điều kiện lọc.
* ****AND:****sử dụng 2 điều kiện, cả 2 điều kiện đúng mới được tính là đúng, 1 trong 2 bị sai sẽ tính là sai và bỏ qua element.

Đôi lúc chúng ta không chỉ tìm kiếm element dựa trên một thuộc tính duy nhất được bởi vì các thuộc tính của element có thể thay đổi. Vậy nên toán tử "or" giúp ta tìm kiếm element dựa trên nhiều điều kiện. Giả sử ta có 2 điều kiện A và B và dưới đây là bảng kết quả thu được khi sử dụng toán tử "or" cho 2 điều kiện A, B:

****Syntax****: //tag[XPath Statement 1 ****or**** XPath Statement 2]

Operator "and"

Một trong những toán tử quan trọng nhất trong ngôn ngữ lập trình, đối lập với toán tử "or" không ai khác ngoài toán tử "and". Toán tử and dùng để gộp nhiều điều kiện lại với nhau.

****Syntax****: //tag[XPath Statement-1 ****and**** XPath Statement-2]

## **Text**

Chúng ta có thể tìm kiếm tags bằng đoạn text chính xác.

****Syntax****: //tag[****text()=’text value‘]****

Tagname & Attribute & Value -> Kiểm tra attribute là duy nhất)

//tagname[@attribute=‘value’]

Unique + Meaning

Attribute: Id/Class/Name

Attribute value meaning- relate to field name

NOT(Id/Class/Name)=> Any attribute + Unique + Related to field name

Link - Không sử dụng với attribute = href(server)