XPATH Cheat Sheet

Every element does not have an id ---> static id, unique name, unique link text. For those elements we need to build xpath to find and then perform actions on them.

Whatever we use to find an element, id, name, xpath---> It should always be unique.
It should only find one matching node unless we want to capture a list of elements.

Difference between single '/' or double '//'

Single slash '/' anywhere in xpath signifies to look for the element immediately inside the parent element.
Double slash '//' signifies to look for any child or nested--- child element inside the parent element.

1. **Syntax:**

```
//tag[@attribute='value']
```

2. **Multiple Attribute:**

```
//tag[@attribute_name1='attribute_value1'][@attribute_name2='attribute_value2]
```

3. **Using AND:**

```
//tag[@attribute_name1='attribute_value1' and @attribute_name2='attribute_value2]
```

4. **Using OR:**

```
//tag[@attribute_name1='attribute_value1' or @attribute_name2='attribute_value2]
```

5. **contains():** It is used to identify an element, when we are familiar with some part of the attributes value of an element.

```
//tag[contains(@attribute_name,'attribute_value')]
```

6. **starts-with():** It is used to identify an element, when we are familiar with the attributes value (starting with the specified text) of an element.

```
//tag[starts-with(@attribute_name,'attribute_value')]
```

7. **text():** This mechanism is used to locate an element based on the text available on a webpage

```
//div[@class='homepage---hero']//a[text()='Enroll now']
```

8. **last():** Selects the last element (of mentioned type) out of all input element present

```
findElement(By.xpath("(//input[@type='text'])[last()]"))
```

9. **position():** Selects the element out of all input element present depending on the position number provided

```
findElement(By.xpath("(//input[@type='text'])[position()=2]"))
```

10. Finding elements using index

By providing the index position in the square brackets, we could move to the nth element.

```
findElement(By.xpath("//label[2]"))
```

Relative xpath using single '/' for Login link

```
//div[@id='navbar']/div/div/div/ul/li[2]/a
```

Relative xpath using double '//' for Login link.

```
//div[@id='navbar']//ul/li[2]/a
```

Don't use "*", always use the tag name.

11. Parent

Syntax: xpath-to-some-element//parent::<tag>

12. Preceding Sibling

Syntax: xpath-to-some-element//preceding-sibling::<tag>

13. Following Sibling

Syntax: xpath-to-some-element//following-sibling::<tag>

Summary

XPath Syntax	Note
//T[@id='i']	Locating Element with tag-name T and id i
//*[@id='i']	Locating element(s) with id i
//T[@name='n']	Locating Element with tag-name T with name n
//*[@name='n']	Locating element(s) with name n
//T[@A='V']	Locating Element with tag-name T with an attribute A and associated value exactly V
//T[contains(@A,'V')]	Locating Element with tag-name T with attribute A containing substring of the value is V
//T[starts-with(@A,'V')]	Locating Element with tag-name T with attribute A and it's value starts with V
//T1[@A1='V1'] | //T2[@A2='V2']	Locating Element with tag-name T with attribute A1 and value V1 or Element with tag-name with attribute A2 and value V2
//T[@A1='V1' or @A2='V2']	Locating element(s) with tag-name T with attributes A1 with value V1 or attribute A2 with value V2
//T[text()='V']	Locating element with tag-name T with exact inner text V
/T[contains(text(),'V')]	Locating element with tag-name T containing inner text V
//T/..	Locating the parent of and element with tag-name T
//T/parent::*	
//T[count(*)=0]	Locating elements with tag -name T who has no child elements
//T[count(*)=1]	Locating elements with tag -name T who has only one child element
//T2/following-sibling::T1	Locating element(s) with tag-name following some sibling(s) of element
//T2/following::T1	Locating element(s) with tag-name T following any element of type
//T2/preceding-sibling::T1	Locating element(s) with tag-name preceding some sibling(s) of element
//T2/preceding::T1	Locating element(s) with tag-name T1 preceding any element of type T2
//T[@disabled]	Locating element(s) with tag-name T that is disabled
//T[@disabled='true']	
//T[not(@disabled)]	Locating element(s) with tag-name T that is not disabled
//T[@disabled='false']	
//T[@checked]	Locating checkbox or radio element(s) that is checked
//T/*[1]	Locating first child of element T
//T/*[last()]	Locating last child of element T
//T[1]	Locating first element with tag-name T
//T[last()]	Locating last element with tag-name T child
//*[2][name()='T']	Second child element that is an T

Exercise:

http://letskodeit.teachable.com/pages/practice

-Find the price of the course “Python Programming Language”
-Find the course that costs “25”