XPATH Crash Course:

1. What is XPath?
    a. Xpath means the address of the every element.
    b. There are multiple ways to write a XPATH.
2. What's the importance of XPATH in WebAutomation?
3. Types of XPaths:
    a. Absolute XPaths
    b. Relative XPaths
4. How to write the XPATHS?
    a. XPATH syntax and Functions:
        i. /tagName
        ii. //tagName[@attributeName="attributeValue"]
        iii. And, OR functions
        iv. Text(), normalize-space(),
        v. Index, position, last, etc.
    b. Difference b/w "/" and "//"?
        i. "/" is very slow.
        ii. "//" is fast it directly points out to element.
    c. What is the right platform to write and verify XPath.
        i. Chrome DevTools
            1. Note: Do not write Xpaths in ChromeDevTools.
            2. Most of the time you'll get the wrong answer in this, because ChromeDevTools are also used for Key-finder inspite of XPaths.
        ii. Console $x
        iii. Using SelectorHub
    d. How to write XPath using XPath functions?
        i. Using single forward slash (/):
            1. (Whenever we're writing using the single forward slash we need to traverse from top).
            2. E.g. -> /html/body/div[1] (This also known as Absolute XPath).
        ii. Using double forward slash (//):
            1. Here, we don't need to traverse from starting/top.
            2. E.g. -> //a[@title='Platform']
            3. (Let's say you need to use the Partial value of your attribute value you can go with contains) -> //a[contains(@title,'Platfo')]

4. You can also do (or), (and) function:
    a. (or) is used to check if anyone xpath is present then it will find:-

       e.g. -> //a[contains(@title,'platfo')] or

       //a[contains(@class,'yahoo')]

    b. (and) is used to check if both the xpaths are able to find the element on the page.

       e.g. -> //a[contains(@title,'hello') and

       //a[contains(@class,'world')]

iii. Using the text() function:
    1. Syntax: //tagname[text()='value']
    2. E.g. -> //a[text()='Platform']
    3. E.g. -> //a[contains(text(),'Platform']
iv. Using the normalize-space() function:
    1. Syntax: //tagname[normalize-space()='value']
    2. E.g. -> //h3(normalize-space()='SelectorHub')]

e. How to write Relative XPath using axes and without using axes?
    i. Means writing XPath w.r.t some other parent.
    ii. Using axes:
        1. Condition -> writing the xpath for Forum.
            a. First we need to write the xpath for 'forum', and then need to mention its parent element.
            b. E.g. -> //a[@title='blog']/ancestor::ul//a[@title='Forum']
    iii. Without axes:
        1. Condition -> writing the xpath of Forum w.r.t. Blog
        2. Flow would be: First write the xpath of blog go to common parent and from parent writing the xpath of that particular element – forum.
        3. E.g. -> //ul/li/a[@title='Blog']/

f. Learn to write XPath for SVG Elements.
    i. Syntax: //*[local-name()='svg']
    ii. You can also use indexes for mentioning the position of the svg.
    iii. E.g. -> //*[local-name()='svg'][3]
    iv. Syntax for child-elements: //*[name()='path']

g. How to write XPath for Shadow elements.
    i. What is Shadow DOM?
        1. It's a DOM inside DOM.
        2. There are (2) types of shadow DOM -> Open and Closed.
        3. Actual DOM will always start with <!docttype HTML>.
        4. Open Shadow-DOM only supports XPaths but not the Closed shadow-DOM supports XPath.

h. How to write Xpath for iFrame Elements and iFrame.
        1. Syntax: //iFrame[@title='allowScreen']

i. Writing XPath for dynamic Elements which disappear from DOM.
    i. The elements that aren't present in DOM, for this you need to use selector hub.
        1. Right click on the element and select SelectorHub.