What is XPath ?  
  
XPath (XML Path Language), as defined by the World Wide Web Consortium (W3C), is a query-language for selecting nodes from an XML document. In addition, XPath can be used to compute values (e.g., strings, numbers, or Boolean values) from the content of an XML document. The following statements are important information regarding XPath:  
  
XPath is a syntax for defining parts of an XML document  
XPath uses path expressions to navigate in XML documents  
XPath contains a library of standard functions  
XPath is a major element in XSLT and in XQuery  
XPath is a W3C recommendation  
  
  
 How to use Absolute XPath and Relative XPath?  
  
Using Absolute XPath  
  
Using Absolute XPath the user can directly access or find elements from the DOM. The main disadvantage of using Absolute XPath is that, if there are any changes made in the path of the element then that XPath fails to identify that particular element.  
  
Absolute XPath begins with a single forward slash "/", which means you can select the element from the root.  
  
Here is an example of Absolute XPath  
  
/html/body/div[5]/div[2]/div/div[2]/div[2]/h2[1]  
  
  
Using Relative XPath  
  
Using Relative XPath the user can select an element from anywhere from the DOM…  
Relative XPath starts with the double forward slashes ‘//’, which means it can search the element anywhere in the webpage.  
  
Here is an example of Relative XPath   
  
.//\*[@id='answers']/h2[1]/a[1]

Explain the pause feature in Selenium ?

The pause command is a simple wait command and useful to delay the execution of the automated testing for the specified time. Note that the wait time is in MILLIseconds. So if you want to wait for 3 seconds, enter 3000.nium IDE.

Can selenium handle Windows based pop up?

Selenium can handle Windows based pop up. There may be scenarios where a web page opens more than one window after performing some actions on it. The child windows that get opened may be a pop up containing some information or advertisement.

Selenium uses the **getWindowHandles ()** and **getWindowHandle ()** methods to work with child windows. The getWindowHandles () method contains all the window handle ids of the opened windows. The window id handles are held in the form of Set data structure [containing data type as String].

The getWindowHandle () method is used to store the window handle id of the present active window. As we know, getWindowHandles () method is used to store all the opened window handle ids. To iterate through all the handles, iterator () and next () methods are used.

Since getWindowHandles () stores the window ids in the form of Set data structure, we have to **import java.util.Set** in our code. Also, for using the iterator () method, we have to **import java.util.Iterator** and **import java.util.List**.

Finally to switch to a particular window, switchTo.().window() method is used. The handle id of the window where we want to switch is passed as an argument to that method.

The steps to be followed to implement the above concept −

* After the application is launched, let us first store all the window handle ids in a Set data structure with the help of getWindowHandles () method.
* We shall iterate through all the window handle ids with the help of iterator () and next () methods.
* Let us then grab the current window handle id with the help of getWindowHandle () method.
* Then switch to that window with switchTo.().window() method.