|  |
| --- |
| **Question: Write an example of the webelement located by CSS**  /\* below syntax will find "input" tag node which contains "id=email" attribute \*/  css=input[id=email]  In selenium we can write it as  WebElement Email = driver.findElement(By.cssSelector("input[id=email]"));  Email.SendKeys("[hello@sampleemail.com](mailto:hello@sampleemail.com)"); |
| There are there important special characters: **1. '^' symbol, represents the starting text in a string.** **2. '$' symbol represents the ending text in a string.** **3. '\*' symbol represents contains text in a string.**  **CSS Locators for Sub-string matches(Start, end and containing text) in selenium** /\*It will find input tag which contains 'id' attribute starting with 'ema' text. Email starts with 'ema' \*/  css=input[id^='ema']  **How to write an WebElement by CSS using starts-with**  /\*It will find input tag which contains 'id' attribute ending with 'ail' text. Email ends with 'mail' \*/  css=input[id$='mail']  /\* It will find input tag which contains 'id' attribute containing 'mai' text. Email contains 'mai' \*/  css=input[id\*='mai']  **CSS Element locator using child Selectors**  /\* First it will find Form tag following remaining path to locate child node.\*/  css=form>label>input[id=PersistentCookie]  [XPath tutorial for Selenium](http://www.seleniumeasy.com/selenium-tutorials/xpath-tutorial-for-selenium)  XPath is designed to allow the navigation of XML documents,with the purpose of selecting individual elements, attributes, or some other part of an XML document for specific processing.  What is XML? The Extensible Markup Language (XML) is the context in which the XML Path Language, XPath, exists.  XML provides a standard syntax for the markup of data and documents. |
|  |
|  |
| **difference between implicit wait & explicit wait** |
| **Implicit Wait:** Implicit waits are used to provide a default waiting time (say 30 seconds) between each consecutive test step/command across the entire test script. Thus, subsequent test step would only execute when the 30 seconds have elapsed after executing the previous test step/command.  driver.manage().timeouts().implicitlyWait(30, TimeUnit.***SECONDS***);  **Explicit Wait:** Explicit waits are used to halt the execution till the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, explicit waits are applied for a particular instance only.  WebDriverWait wait = **new** WebDriverWait(Gmail, 100);  wait.until(ExpectedConditions.*visibilityOfElementLocated*(  By.*xpath*(".//input[@type='password' and@aria-label='Enter your password']"))); |
| **2 table join sql query**   |  | | --- | | **CUSTOMER** | | Id | | FirstName | | LastName | | City | | Country | | Phone |  |  | | --- | | **ORDER** | | Id | | OrderDate | | OrderNumber | | CustomerId | | TotalAmount |    SQL JOIN Examples **Problem:** List all orders with customer information    1. **SELECT OrderNumber, TotalAmount, FirstName, LastName, City, Country** 2. **FROM [Order] JOIN Customer** 3. **ON [Order].CustomerId = Customer.Id**   The general LEFT OUTER JOIN syntax is:    1. **SELECT OrderNumber, TotalAmount, FirstName, LastName, City, Country** 2. **FROM Customer C LEFT JOIN [Order] O** 3. **ON O.CustomerId = C.Id** 4. **ORDER BY TotalAmount**  SQL RIGHT JOIN Example **Problem:** List customers that have not placed orders    1. **SELECT TotalAmount, FirstName, LastName, City, Country** 2. **FROM [Order] O RIGHT JOIN Customer C** 3. **ON O.CustomerId = C.Id** 4. **WHERE TotalAmount IS NULL**  SQL FULL JOIN Examples  |  | | --- | | **SUPPLIER** | | Id | | CompanyName | | ContactName | | City | | Country | | Phone | | Fax |  |  | | --- | | **CUSTOMER** | | Id | | FirstName | | LastName | | City | | Country | | Phone |   **Problem:** Match all customers and suppliers by country    1. **SELECT C.FirstName, C.LastName, C.Country AS CustomerCountry,** 2. **S.Country AS SupplierCountry, S.CompanyName** 3. **FROM Customer C FULL JOIN Supplier S** 4. **ON C.Country = S.Country** 5. **ORDER BY C.Country, S.Country**   This returns suppliers that have no customers in their country,  and customers that have no suppliers in their country,  and customers and suppliers that are from the same country |
|  |
| **Write an assertion code for x<y where x=6 , y=5**  @Test  **public** **void** done(){  System.***out***.println("IM in test assert");  **int** i=5;  **int** x=6;  **try**{  Assert.*assertEquals*(i, x);  }**catch** (Exception t){  System.***out***.println(t);  } |
|  |
| **Explain TestNG Annotations** |
| 1. @BeforeSuite 2. @BeforeTest 3. @BeforeClass 4. @BeforeMethod 5. @Test Case 1 6. @AfterMethod 7. @AfterClass 8. @AfterTest 9. @AfterSuite 10. @DataProvider   **#1)** **@Test**  This is the main part of our automation script where we will write the business logic, the things which we want to automate. We can pass attributes to our test method.  **Below are the lists of attributes that we can pass to our Test method:**   * **alwaysRun**: This is used when we want to make sure a method always runs even if the parameters on which the method depends, fails. If set to true, this test method will always run. Eg: *@Test(alwaysRun = true)* * **dataProvider**: TestNG dataProvider is used to provide any data for parameterization. Eg. *@Test(dataProvider = “Hello”).* * **dataProviderClass**: This is the class from where we pass the data to data provider. In our case dataProvider class name is “Hello”. * **dependsOnGroups**: It is the list of groups this method depends on. Eg: *@Test (groups = { “City” ,”State” })* * **dependsOnMethods:**This command is used to execute a method based on its dependent method. Eg: *@Test (dependsOnMethods = { “OpenBrowser” ,”database is up” })* * **description**: It is the description for the method. Eg: *@Test(description = “test method”)* * **invocationCount**: It refers to the number of times a method should be invoked. It will work as a loop. Eg: *@Test(invocationCount = 7)* . Hence, this method will execute 7 times. * **invocationTimeOut**: This refers to the maximum number of milliseconds a method should take for all the invocationCount to complete. This attribute will be ignored if invocationCount is not specified. Eg: *@Test(invocationCount =7,invocationTimeOut = 30 )* * **priority**: This command sets the priority of the test method. Lower priorities will be scheduled first. Eg:*@Test(priority =1 )*   **And a PDF** |
| |  | | --- | | Write down the page object page files frame for the amazon.ca | |  | | Write down the selenium code to automate login functionality of amazon.ca | |  | | Write the code for Fibonacci series | |  | | write the code to find out the given number is prime number or not | |  | | write down the code for XPath for dynamic element  **A PDF** | |  | | Write down the logic for N Factorial | |  | | Write down the logic for reverse a string | |

|  |
| --- |
| Write down the logic for reverse a string |
|  |
|  |
| Tell me about your self |
|  |
| how you are good at manual testing as well ? |
|  |
| What is data driven framework, Explain  PDF - How Data Driven Testing Works (Examples of QTP and Selenium) — Software Testing Help |
|  |
| What is Inheritance ?  Book Notes |
| Can we access private variables in inheritance from base class to derive class ?  Book Notes –no you cant read in depth from the copy notes  …………………………………………………………………………………………………   |  | | --- | | What is "verify" in Selenium  Both Assert and Verify commands are used to find whether a given input is present or not on the webpage. There are some difference between Assert and Verify in Selenium.  Let’s see the basic difference between Assert and Verify in Selenium:  **Assert command in selenium:**  When an “assert” command fails, the test execution will be aborted. So when the Assertion fails, all the test steps after that line of code are skipped. The solution to overcoming this issue is to use a try-catch block. We use the Assertion in the try catch block. Mostly, the assert command is used when the end result of the check value should pass to continue to the next step.  In simple words, if the assert condition is true then the program control will execute the next test step but if the condition is false, the execution will stop and further test step will not be executed.  To overcome this we use Soft Assert in TestNG. Checkout below post to know what is Soft Assert.  **Verify command in selenium:**  When a “verify” command fails, the test will continue executing and logging the failure. Mostly, the Verify command is used to check non-critical things. In such cases where we move forward even though the end result of the check value is failed.  In simple words, there wont be any halt in the test execution even though the verify condition is true or false. | |  | | How do you approach API Testing ? | |  | | What methods/calls you have used in API Testing ??  There are mainly 4 methods involve in API Testing like Get, POST, Delete, and PUT.  GET- The GET method is used to extract information from the given server using a given URI. While using GET request, it should only extract data and should have no other effect on the data.  POST- A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.  PUT- Replaces all current representations of the target resource with the uploaded content.  DELETE- Removes all current representations of the target resource given by a URI. | |  | |  | | what is PUT & DELETE in API Testing  READ PDF - REST & SOAP Request Methods \_ Resources \_ SoapUI | | How do you automate validations in Selenium ?  READ THE PDF  Automating Different Input Fields Using Selenium Webdriver \_ Grazitti Interactive  Login page, there would be 3 valid scenarios - 1) valid Login 2) Invalid Login 3) Negative Testing scenario  system is expected to prompt the user with an error message.  But a negative testing scenario would be where you are trying to break the application. For Instance, 1) leaving Password blank, 2) trying to navigate using the URL, 3) using the forward button of the IE bypassing the Login page etc.....  @Config(url="http://systemunder.test", browser=Browser.FIREFOX)  public class TestLogin extends AutomationTest {  @Test  public void testLoginWorks() {  setText(By.id("username"), "valid\_username")  .setText(By.id("password"), "valid\_password")  .click(By.id("btnLogin"))  .validatePresent(By.id("logout\_link"));  }  @Test  public void testPasswordBlank() {  setText(By.id("username"), "invalid\_username")  .setText(By.id("password"), "")  .validateText(By.id("error\_message"), "Password is blank.");  }  @Test  public void testUrl() {  navigateTo("/profile")  .validateUrl("/login"); // make sure they get redirected to login page.  }  } | |  | | Explain WSDL concept in API Testing  Read PDF - Web Services Explained  What is difference between WebServices & API Testing  **API vs Web Service**  API and Web service serve as a means of communication. The only difference is that a Web service facilitates interaction between two machines over a network. An API acts as an interface between two different applications so that they can communicate with each other. An API is a method by which the third-party vendors can write programs that interface easily with other programs. A Web service is designed to have an interface that is depicted in a machine-processable format usually specified in Web Service Description Language (WSDL). Typically, “HTTP” is the most commonly used [protocol](http://www.differencebetween.net/category/technology/protocols-formats/) for communication. Web service also uses SOAP, [REST](http://www.differencebetween.net/technology/internet/difference-between-rest-and-soap/), and XML-RPC as a means of communication. API may use any means of communication to initiate interaction between applications. For example, the [system](http://www.differencebetween.net/miscellaneous/difference-between-caste-system-and-class-system/) calls are invoked using interrupts by the [Linux](http://www.differencebetween.net/technology/difference-between-ubuntu-and-linux/) kernel API.  An API exactly defines the methods for one [software](http://www.differencebetween.net/category/technology/software-technology/) program to interact with the other. When this action involves sending [data](http://www.differencebetween.net/technology/difference-between-data-warehousing-and-data-marts/) over a network, Web services come into the picture. An API generally involves calling functions from within a software program.  In case of Web applications, the API used is web based. Desktop applications such as spreadsheets and word documents use VBA and COM-based APIs which don’t involve Web service. A [server](http://www.differencebetween.net/technology/software-technology/difference-between-application-server-and-web-server/) application such as Joomla may use a PHP-based API present within the server which doesn’t require Web service.  A Web service is merely an API wrapped in HTTP. An API doesn’t always need to be web based. An API consists of a complete set of rules and specifications for a software program to follow in order to facilitate interaction. A Web service might not contain a complete set of specifications and sometimes might not be able to perform all the tasks that may be possible from a complete API.  The APIs can be exposed in a number of ways which include: COM objects, DLL and .H files in C/C++ programming language, JAR files or RMI in Java, [XML](http://www.differencebetween.net/technology/difference-between-xml-and-xsd/) over HTTP, JSON over HTTP, etc. The method used by Web service to expose the API is strictly through a network.  Summary:  1. All Web services are APIs but all APIs are not Web services.  2. Web services might not perform all the operations that an API [would](http://www.differencebetween.net/language/difference-between-would-and-should/) perform.  3. A Web service uses only three styles of use: SOAP, REST and XML-RPC for  communication whereas API may use any style for communication.  4. A Web service always needs a network for its operation whereas an API doesn’t need  a network for its operation.  5. An API facilitates interfacing directly with an application whereas a Web service is a  Read more: [Difference Between API and Web Service | Difference Between](http://www.differencebetween.net/technology/internet/difference-between-api-and-web-service/#ixzz50KMlo9eQ) <http://www.differencebetween.net/technology/internet/difference-between-api-and-web-service/#ixzz50KMlo9eQ>   |  | | --- | | Day to Day activity in Agile?  Explain Agile in depth. | |  | | How long sprint was?  2 Weeks | |  | | Explain framework which was created by you?  READ PDF - Test Automation Framework | |  | | What is difference between Data driven and keyword driven?  READ PDF - What is the difference between keyword driven and Data driven frameworks in Automation\_ - Software Quality Assurance & Testing Stack Exchange | |  | | Where did you write your testcases? | | TestLink. Jira. | | How did you integrate with selenium? | |  | | How did you use Jenkins with selenium? | |  | | Challenges for automation testing? | |  | | How did you automate your script in selenium? | |  | | Explain Backend testing? | |   ………………………………………………………………………………………………………………………. |