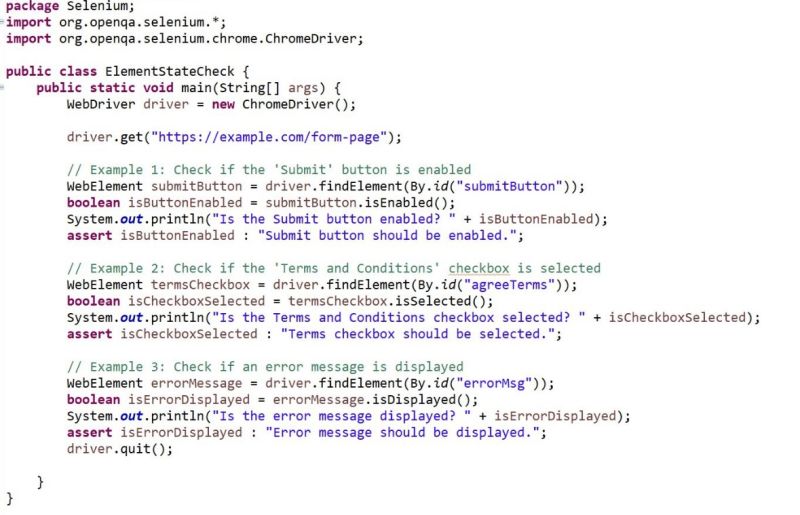
IMP notes

1. Web locator CSS is browser-dependent
2. Question: 𝐂𝐚𝐧 𝐲𝐨𝐮 𝐞𝐱𝐩𝐥𝐚𝐢𝐧 𝐭𝐡𝐞 𝐩𝐮𝐫𝐩𝐨𝐬𝐞 𝐚𝐧𝐝 𝐮𝐬𝐞 𝐨𝐟 𝐭𝐡𝐞 𝐢𝐬𝐄𝐧𝐚𝐛𝐥𝐞𝐝(), 𝐢𝐬𝐒𝐞𝐥𝐞𝐜𝐭𝐞𝐝(), 𝐚𝐧𝐝 𝐢𝐬𝐃𝐢𝐬𝐩𝐥𝐚𝐲𝐞𝐝() 𝐦𝐞𝐭𝐡𝐨𝐝𝐬 𝐢𝐧 𝐒𝐞𝐥𝐞𝐧𝐢𝐮𝐦?  
     
   Answer: In Selenium, 𝐢𝐬𝐄𝐧𝐚𝐛𝐥𝐞𝐝(), 𝐢𝐬𝐒𝐞𝐥𝐞𝐜𝐭𝐞𝐝(), 𝐚𝐧𝐝 𝐢𝐬𝐃𝐢𝐬𝐩𝐥𝐚𝐲𝐞𝐝() are methods used to verify the state of web elements. They help determine whether elements are interactive, selected, or visible, which is crucial for creating robust automation tests.  
     
   𝟏. 𝐢𝐬𝐄𝐧𝐚𝐛𝐥𝐞𝐝()  
   𝐏𝐮𝐫𝐩𝐨𝐬𝐞: The `isEnabled()` method checks if a web element, such as a button, input field, or any other form element, is enabled and can be interacted with.  
   𝐔𝐬𝐞 𝐂𝐚𝐬𝐞: You might use `isEnabled()` to verify that a "Submit" button is enabled only after all required form fields are filled out.  
     
   𝟐. 𝐢𝐬𝐒𝐞𝐥𝐞𝐜𝐭𝐞𝐝()  
   𝐏𝐮𝐫𝐩𝐨𝐬𝐞: The `isSelected()` method checks if a web element, typically a checkbox, radio button, or option in a dropdown, is selected.  
   𝐔𝐬𝐞 𝐂𝐚𝐬𝐞: You would use `isSelected()` to ensure that a checkbox or radio button is selected as part of a form submission process.  
     
   𝟑. 𝐢𝐬𝐃𝐢𝐬𝐩𝐥𝐚𝐲𝐞𝐝()  
   𝐏𝐮𝐫𝐩𝐨𝐬𝐞: The `isDisplayed()` method checks if a web element is visible on the web page. It is useful for ensuring that elements are present and visible to the user.  
   𝐔𝐬𝐞 𝐂𝐚𝐬𝐞: `isDisplayed()` could be used to confirm that an error message is shown when a user submits a form with invalid data.  
     
     
   \*\* These methods are fundamental in creating reliable Selenium tests. They ensure that your test scripts interact only with elements that are ready for action, which helps prevent flaky tests and improves the stability of your test suite.



1. 𝐐𝐮𝐞𝐬𝐭𝐢𝐨𝐧: 𝐖𝐡𝐢𝐜𝐡 𝐀𝐠𝐢𝐥𝐞 𝐜𝐞𝐫𝐞𝐦𝐨𝐧𝐢𝐞𝐬 𝐡𝐚𝐯𝐞 𝐲𝐨𝐮 𝐩𝐚𝐫𝐭𝐢𝐜𝐢𝐩𝐚𝐭𝐞𝐝 𝐢𝐧, 𝐚𝐧𝐝 𝐡𝐨𝐰 𝐡𝐚𝐯𝐞 𝐲𝐨𝐮 𝐜𝐨𝐧𝐭𝐫𝐢𝐛𝐮𝐭𝐞𝐝 𝐚𝐬 𝐚 𝐐𝐀 𝐩𝐫𝐨𝐟𝐞𝐬𝐬𝐢𝐨𝐧𝐚𝐥?  
     
   Answer: In my role as a QA professional, I have actively participated in several key Agile ceremonies, contributing in the following ways:  
     
   𝟏. 𝐒𝐩𝐫𝐢𝐧𝐭 𝐏𝐥𝐚𝐧𝐧𝐢𝐧𝐠:  
   𝐂𝐨𝐧𝐭𝐫𝐢𝐛𝐮𝐭𝐢𝐨𝐧: I collaborate with the product owner and development team to understand the user stories and their acceptance criteria. I ensure that the requirements are clear and testable by asking questions and providing feedback. Additionally, I help estimate the effort required for testing and prepare for the testing activities that will be needed during the sprint. This ensures that testing is well-aligned with sprint goals and user story definitions.  
     
   𝟐. 𝐃𝐚𝐢𝐥𝐲 𝐒𝐭𝐚𝐧𝐝-𝐔𝐩 (𝐃𝐚𝐢𝐥𝐲 𝐒𝐜𝐫𝐮𝐦):  
   𝐂𝐨𝐧𝐭𝐫𝐢𝐛𝐮𝐭𝐢𝐨𝐧: During daily stand-ups, I provide updates on the status of my testing activities, including any defects discovered and their impact. I also share any challenges or blockers that may hinder progress. By staying updated on the development team's progress, I can adjust my testing strategy accordingly and address issues in real-time. This communication helps keep the team aligned and facilitates quick resolution of problems.  
     
   𝟑. 𝐒𝐩𝐫𝐢𝐧𝐭 𝐑𝐞𝐯𝐢𝐞𝐰:  
   𝐂𝐨𝐧𝐭𝐫𝐢𝐛𝐮𝐭𝐢𝐨𝐧: In sprint reviews, I demonstrate the results of my testing, including defect reports and feedback on the quality of the delivered features. I provide insights into whether the features meet the acceptance criteria and highlight any issues that need addressing. My feedback during this meeting helps ensure that the product increment meets quality standards and aligns with stakeholder expectations.  
     
   𝟒. 𝐒𝐩𝐫𝐢𝐧𝐭 𝐑𝐞𝐭𝐫𝐨𝐬𝐩𝐞𝐜𝐭𝐢𝐯𝐞:  
   𝐂𝐨𝐧𝐭𝐫𝐢𝐛𝐮𝐭𝐢𝐨𝐧: I actively participate in sprint retrospectives to reflect on the testing process and its outcomes. I discuss what went well, identify any challenges or areas for improvement, and suggest actionable steps to enhance our QA practices. This helps in continuously refining our approach and improving overall team efficiency and effectiveness.  
     
   𝟓. 𝐁𝐚𝐜𝐤𝐥𝐨𝐠 𝐑𝐞𝐟𝐢𝐧𝐞𝐦𝐞𝐧𝐭 (𝐆𝐫𝐨𝐨𝐦𝐢𝐧𝐠):  
   𝐂𝐨𝐧𝐭𝐫𝐢𝐛𝐮𝐭𝐢𝐨𝐧: During backlog refinement sessions, I help refine and clarify user stories, ensuring they are testable and include comprehensive acceptance criteria. I provide input on potential testing challenges and assist in prioritizing backlog items based on their complexity and testability. This preparation helps in ensuring that the backlog is well-defined and ready for upcoming sprints.
2. Advanced XPATH for software tester  
     
   Mastering XPath is a game changer for [hashtag#software](https://www.linkedin.com/feed/hashtag/?keywords=software&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7236055237231271937) [hashtag#testers](https://www.linkedin.com/feed/hashtag/?keywords=testers&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7236055237231271937). While basic XPath is useful, advanced XPath techniques can take your automation skills to the next level.  
     
   Here are some powerful examples of advanced XPath:  
     
   1. Absolute XPath:   
   This specifies the full, exact path to an element, starting from the top of the HTML document.   
   For example: /html/body/div[2]/table/tr[3]/td[4]  
     
   2. Relative XPath:   
   This allows you to locate elements based on their attributes, without needing the full path.   
   For example: //input[@type='text' and @name='username']  
     
   3. Wildcard XPath:   
   This uses the \* symbol as a wildcard to match any element, making it easier to find elements with dynamic or changing attributes.   
   For example: //\*[contains(@class, 'button')]  
     
   4. Indexed XPath:   
   This helps you target a specific element within a group of matching elements, using the index position.   
   For example: (//div[@class='item'])[3]  
     
   5. Sibling XPath:   
   This lets you select elements that are at the same level as your target element.   
   For example: //h1/following-sibling::p[1]  
     
   6. Parent XPath:   
   This allows you to select the parent element of your target.   
   For example: //div[@id='container']/parent::body  
     
   7. Ancestor XPath:   
   This lets you select any ancestor (parent, grandparent, etc.) element of your target.   
   For example: //div[@id='content']//h2/ancestor::div  
     
   With these advanced XPath techniques in your toolkit, you can write more robust, reliable, and efficient automation scripts. Unlock your full potential as a software tester and start exploring the power of advanced XPath today! 🚀