

## Selenium tricky interview questions

### Table of content

[Selenium Tricky Questions](#)

## Selenium Tricky Questions

When does Implicitly Wait is applicable for

findElements method.?

Implicit wait in selenium is applicable for all findElement statements, but the applicability of implicit wait to the findElements is Limited.

Implicitly wait is applicable on findElements only when there are no elements present, selenium webdriver moves to next command the moment it finds a single element.

*If there is Zero matching elements selenium waits till the time it reaches provided implicitly wait*

*Scenario 1 :* Consider a page having 100 checkboxes and applied implicit wait is 30 Seconds

When the page is loading, selenium tries to find all the matching elements, but at that moment, there no matching elements.

After 10 seconds of wait 1 checkbox is loaded, now selenium finds that one checkbox and concludes that there is only one element, so it ends the wait and moves to the next command.

Selenium does not wait till 100 checkboxes loaded into the page, as the webdriver does not know how many elements going to be present on the page.

*Scenario 2 :* Consider a page having 0 checkboxes and applied implicit wait is 30 Seconds

Selenium tries to find all the matching elements in DOM, but so far, no checkbox is loaded.

Now there are zero matches, so selenium waits till 30 seconds, and if does not find element after 30 seconds selenium webdriver throws NoSuchElementException

How do you check an option is present in a dropdown or not ? \*New

Answer 1 : Find the dropdown and Create an object to the **Select** class and Retrieve all options

using **getOptions()** method and Iterate all options and compare the text.

Answer 2 : Find the dropdown and create the object to **Select** class and try to select the option using **selectByVisibleText** , if the option is there, then it will be selected; otherwise, it will throw an exception. So based on this exception, we can decide whether the element is there or not.

Answer 3 : Write the XPath/CSS which matches all the options( **//select/option** ) inside the dropdown and get InnerHtml (why the hell I am not using getText ?, have a Guess). And compare the text and check.

Answer 4 : Write **locator** for the option directly and **find the element** and do nothing, if we can find element then result is positive otherwise negative

Doubt on 4 : How do you decide, it present only on dropdown not anywhere else on the page, because in both cases **Answer 4** results positive.

Answer 5 : Write an Xpath/Css **locator** based on the parent(dropdown) and then find a child (the target) like, **//select[@id='some']/option[text()='cherchertech']** , if element found, then positive otherwise negative. We can have try and catch block here.

Answer 6 : What if try and catch block is irritating/irrelevant to test, then use the same **XPath** as **Answer 4** but use findElements instead of findElement. Now check the number of elements, if 0 then negative otherwise positive All the above answers are correct, use it based on the requirement but answer in the interview based on your experience, I would go with answers 5 and 6

How do you check whether an element enabled or not ?, condition: there is no such attribute as "disabled" in the element ?

Explanation : When there is no such attribute as disabled in the element, the isEnabled() method from selenium does not work.

Solution : There is some attribute that makes the element to be disabled, so we have to **find the element** and get the attribute using the getAttribute() method then compare what is the value for enabled and disabled. Based on this, we can conclude whether the element is enabled or not.

Why do we write **WebDriver driver = new FirefoxDriver()** , why not **SearchContext driver = new FirefoxDriver()** ?

Don't hurry to answer like, WebDriver is an interface and FirefoxDriver is Class. This is right, but you need to understand the question first.

He is asking about SearchContext, which is a parent interface of WebDriver, so that the answer would be. Yes, we can write like that because SearchContext is the parent of all the Interfaces and classes present in selenium.

Is it possible to automate captcha?

Yes, we can automate the captcha, but there is a limitation that we can automate our own captcha but not others'

For example, a company has captcha on their website, so somebody has to check it, but at the same time, it is not possible for a manual tester to check all the captcha's.

So we have to automate the captcha in the dev/QA environment by getting the captcha answer in some attribute of the element, so based on that attribute, we can enter the value to the text bar, which accepts captcha value.

We should remove this attribute while pushing the code to the Live environment.

## Testcase failed saying "ElementNotVisible", but when analyzed manually, the element is visible ? How to Handle it ?

There are a couple of things that may cause this issue.

*The element may not be visible in automation due to the speed of selenium.*

*If you closed a hidden division pop up and tried to perform the action, then there is a chance that hidden division popup' animation was not over, which could cause this issue.*

*There could be another element which has the same **XPath** or **locator** on some other page*

Example : Consider you have an element that has **XPath** as `//button[@id='abc']` on page X; by clicking some tab on xpage navigates the user to Y page, Now there is an element on Y page which have XPath same as `//button[@id='abc']`.

But when you launch your application, the application may be directly landed on page Y. So with this scenario, if you try to perform on the element on Y page, it could throw an Exception.

Step by step Solution :

*First, verify whether it is really any of the above scenarios?*

*Print number of elements present with that **XPath** using the `findElements` method*

*If there is only one element, please follow program 1; if there is more than 1 element, then follow program 2.*

### Recommended Readings

[selenium framework interview questions](#)

[Core Java Interview Questions Set 1](#)

[Sprint](#)

[Benefits of Java](#)

**Selenium Relative Locators But I am hesitant to use**

**Robot class in selenium**

**Tags & Hooks in Cucumber with Selenium | BDD**

**Parameterize Cucumber BDD with selenium**