MOST POPULAR AUTOMATION INTERVIEW QUESTIONS

ULTIMATE QA

WWW.ULTIMATEQA.COM

TABLE OF CONTENTS

MOST POPULAR AUTOMATION INTERVIEW QUESTIONS

What are the benefits of automated testing?

- Can help you save time and money
- Reusability and repeatability
- Automatic reporting
- Increased test coverage

What are the levels of automation?

- This is also known as the testing pyramid and refers to the following levels of tests:
- Unit/component testing – these are the base of the pyramid and should make up for the majority of tests. The unit tests focus on testing a small component or piece of functionality from the code. Their purpose is to test this component in isolated conditions. The unit test suite should take a very short time to run, and should run every time there is a change in the code.
- API/service testing – this is the middle tier of the pyramid and refers to the integration tests (can be database tests, API tests etc.). They focus on the business layer of the software, interaction with a database, sending and receiving messages. Compared to the UI tests, these tests can be created and updated faster, because changes in the business will be quickly translated into changes in the code.
- UI testing – this is the top layer of the pyramid.

What are some “best practices” for automated testing?

- Automated tests should be atomic - this means that each test should focus on one thing and one thing only that is verified
- Use page object model to avoid code duplication
- Avoid *thread.sleep()* and use Explicit waits everywhere possible
- When choosing selectors, the preferred order: id > name > css > xpath
- Avoid UI automation if there is another option (API, services)

Which tests should be automated?

- Regression tests. This way, the repetitive manual tasks are reduced.
- Complex, time-consuming tests. For example, tests that verify forms with a lot of data, where manual testing can be more prone to error
- Repetitive tests. The explanation is similar to that of regression tests, repetitive tests will take less time to execute automatically, and the manual tests can focus on more valuable testing (such as usability or exploratory)

- Tests that need to be run against different configurations (combinations of operating systems and browser, for example)
- Tests that execute the same scenario with different data (such as data-driven tests)
- Performance tests (stress and load tests)

When should you not automate tests?

- Tests that are seldom executed - in this case, it may be more time consuming to create the automated tests than it would take to manually execute them
- Exploratory testing - exploratory testing is unplanned testing, where test execution and test design are done at the same time
- Usability testing
- Tests whose execution is faster when done manually
- Tests that cannot be automated fully should not be automated at all

What should you consider when selecting the tools to use for automation testing?

- Which tool fits the project's budget
- How much training is needed for the team to learn the tool
- The level of programming knowledge required for the testing team
- How stable it is
- If it supports the types of tests needed
- Which platforms are supported (web, desktop, mobile)

What is a class in OOP?

A class is a template for creating an object, providing initial values for state (member variables) and implementations of behavior (member functions or methods).

What is an object in OOP?

An object is an instance of a class.

What is a method in OOP?

A method is a procedure associated with a class. It defines the behaviour of the objects created from the class. In other words, a method is an action that an object is able to perform.

What is an abstract class?

An abstract class is a class that cannot be instantiated, which means you cannot create an object of it. Abstract classes can contain abstract methods, and are declared using the keyword “abstract”.

What is an interface?

An interface is a contract that defines the capabilities that an implementing class should provide. It can contain only method declarations, without method definitions. The methods declared in an interface must be implemented by the classes that implement the interface.

What is the difference between a class, an abstract class and an interface?

- Abstract classes can contain abstract and non-abstract methods, while interfaces can only contain abstract methods
- Abstract classes can contain methods, fields, constants, etc. , and interface can contain only methods
- Abstract class doesn’t support multiple inheritance, while interface does
- An abstract class can have class members with different access modifiers, while all members of an interface are public
- An abstract class can contain static members, an interface cannot

What are access modifiers?

Access modifiers are keywords used to define the accessibility of classes, methods and other members. The types of access modifiers are:

- private: objects are accessible only from within the same class;
- protected: objects are accessible from within the same class, or from inherited classes;
- public: objects are accessible from anywhere within the project.

What is a constructor?

A constructor is a block of code used to initialize the objects. It is used to set initial values for object attributes and has its name matches the name of the class.

What is encapsulation?

Encapsulation is a fundamental concept in OOP. It refers to the bundling of data (together with the methods operating that data) into one unit - e.g. a class.

What is inheritance?

Inheritance in OOP is the mechanism in which an object or a class is based on another object or class, retaining similar implementation. Inheritance provides code reusability by allowing programmers to extend an existing class.

The base class is the class whose features are inherited by another class, and the derived class is the class in which those features are inherited.

What is polymorphism?

Polymorphism (meaning many forms) is the ability to implement an interface in multiple ways. In other words, it occurs when we have many classes that are related to each other by inheritance.

As opposed to inheritance, where we can inherit fields and methods from another class, with polymorphism we can use those methods to perform different tasks.

What is method overloading?

Method overloading happens when multiple methods are created with the same name, but differ from each other by the type of input and output.

What is method overriding?

Method overriding refers to implementing a method that overrides in the main class - to achieve method overriding, the implementation must have the same method name, parameters and return types.

What are the components of Selenium?

- Selenium IDE (Integrated Development Environment) – record and play tool for Mozilla Firefox. It is used to prototype the tests, and the tests from the IDE can be exported n multiple programming languages
- Selenium RC (remote Control) – currently known as Selenium 1, it was the first tool of the suite and it was the first tool to provide support for multiple programming languages
- Selenium WebDriver
- Selenium Grid – used for parallelization

What is Selenium WebDriver?

Selenium WebDriver is a collection of APIs used for automating web interactions. It is used to automate web application testing.

What is Selenese?

Selenese is a list of commands used to perform web operations in Selenium IDE.

What are the advantages of using Selenium WebDriver?

- It's a free and open source tool.
- Multi-language support: the framework can be done in the programming language of your choosing: Java, PHP, C#, JavaScript, Python and so on.
- It has multi-browser support and it works on multiple operating systems (Windows, Mac OS, Linux, Unix etc.).
- Has a very large user base and community.

What are some limitations of Selenium WebDriver?

- Selenium WebDriver is a browser automation tool, this means that it can only be used for testing web applications.
- It cannot be used to bypass CAPTCHAs.
- It lacks good built-in reporting capabilities; third party tools need to be used.
- It takes time to learn and requires a lot of knowledge from the testing team.

What testing types can be performed with Selenium WebDriver?

Selenium WebDriver can be used to perform:

- Regression testing
- Functional testing

What operating systems does Selenium WebDriver support?

- Windows
- MacOS
- Linux
- iOS
- Android

What browsers does Selenium WebDriver support?

Selenium WebDriver supports most modern browser, some of the most important are:

- Chrome
- Firefox
- Internet Explorer
- Opera
- Safari

What are the web locators in Selenium Webdriver?

- Id
- Name
- Link Text and Partial Link Text
- CSS Selector
- XPath
- Class Name
- Tag Name
- DOM

What is an XPath?

The XPath is used to locate the elements on the page based on their XML path.

What is the difference between Absolute and Relative XPath and which one should be used to locate the web elements?

- The absolute XPath contains the complete path from the root element to the selected element.
- Relative path starts by referencing the element you want and going from there. It is the preferred XPath because the absolute XPath may change if another web element is added or removed from the page.

How would you locate an element with a dynamic ID?

If all other locators, such as ID or class name, are unavailable, you can use XPath or CSS Selector. Also, you can discuss with the development team to add new tags specifically to be used for the automated tests.

What is the difference between CSS selector and XPath locators?

- CSS selector is faster than XPath.
- CSS selector are easier to read for humans
- CSS path is more reliable across the different browsers
- XPath is more computer-readable

How can you click on a link using Selenium WebDriver?

- By using the method *click():*
- element.click();

How can you type into a text box?

- By using the method *sendKeys(),* which takes a string parameter:
- *element.sendKeys("This is the text I am typing");*

How do you delete the text entered in a textbox?

- With the *clear()* command: *element.clear();*

What is the method that should be used to get the text of a web element in Selenium WebDriver?

- The method is *getText()* and it returns a string:
- string elementText = element.getText();

How can you select a value from a drop-down using Selenium WebDriver?

You will need to use Selenium WebDriver's Select class to locate the drop-down, then pick the item from the list, by value, by visible text or by index:

- dropdown.selectByValue("first option");
- dropdown.selectByIndex(1);
- *dropdown.selectByVisibleText("option text");*

How can you check if a web element is selected, using Selenium WebDriver?

You can use the *isSelected()* method. This will return a Boolean value.

How can you switch between frames in Selenium WebDriver? What about iframes?

The command is the same for both frames and iframes:
driver.switchTo().frame()
And this method takes one of the following 3 parameters:

- The index of the frame, the first frame has the index 0
- Name or ID of the frame
- Previously located WebElement

How can you switch between windows/tabs using Selenium WebDriver?

- Again, the *swithTo()* command should be used.
- First, you need to get the window handle, by using the method *getWindowHandle()* – this method returns a string. This string will be used as a parameter in the following command:
- *var handle = driver.getWindowHandle();*
- *driver.switchTo().window(hadle);*

How can you perform a double click in Selenium WebDriver?

- To double click a web element, you must use Selenium actions:
- *Actions myAction = new Actions(driver);*
- *myAction.doubleClick(element);*

How can you hover over an element?

- To hover over a web element, you must use Selenium actions:
- *Actions myAction = new Actions(driver);*
- *myAction.moveToElement(element).build().perform();*

How can you hover over an element?

- Actions myAction = new Actions(driver);
- act.dragAndDrop(fromElement, toElement).build().perform();

How can you send a key using Selenium WebDriver, for example ALT/CONTROL/SHIFT?

- This is achieved by using actions and the keyDown/keyUP methods:
- Actions myAction = Actions(driver);
- *myAction.keyDown(Keys.CONTROL). keyUp(Keys.CONTROL).build().perform();*

What is the difference between driver.close() and driver.quit()?

- *close()* – closes the current browser window
- *quit()* – closes all the windows opened by the program

What is the difference between driver.get(url) and driver.navigate().to(url)?

Both commands are used to navigate to the URL passed as parameter. The *get()* method refreshes the page and waits for the whole page to load. The navigate(0.to() keeps the browser history and cookies and allows other navigate() commands to be performed (back, forward, etc.)

How can you get the title of a web page using Selenium WebDriver?

By using *getTitle().* The method will return a string with the title of the page.

How can you retrieve a CSS property of an element?

- By using the *getCSSValue()* method. See below an example:
- *element.getCssValue("font-size");*

How can you retrieve a custom property of an element?

- By using the *getAttribute()* method. See below an example:
- *element. getAttribute ("HTML attribute");*

How can you handle pop-up alerts in Selenium WebDriver and what actions can be performed on them?

They can be handled by using the Alert interface. The methods that can be performed on the alerts are:

- *dismiss()* – closes the alert, by clicking “Cancel”
- *accept()* – clicks the “OK” button in the alert
- *getText()* – returns the text displayed in the pop=up
- *sendKeys()* – enters a string in the alert box; this method takes a string parameter

What are the navigation commands in Selenium WebDriver?

- navigate().back();
- navigate().forward();
- navigate().refresh();
- navigate().to();

How can you maximize the browser window using Selenium WebDriver?

- The window can be maximized by using this command:
- *Driver.manage().window().maximize();*

What is the difference between driver.FindElement() and driver.FindElements() ?

The *findElement()* method will return the (first) web element found that matches the used locator, while the *findElements()* method returns a list of all the web elements that match the locator.

How can you get a screenshot using Selenium WebDriver?

- By using the *getScreenshotAs()* method:
- *File screenshot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);*
- *FileUtils.copyFile(screenshot, new File(screenshot_path);*

What is the difference between an assert and a soft assert?

- Assert will fail and stop the test execution after the first failed condition.
- Soft assertion allows the test to continue running even if one of the assertions has failed. The soft assertion is a testing anti-pattern and should be avoided.

What is the difference between assert and verify commands?

- When the assert condition is true, the execution will continue. If the assert condition is false, then the execution stops and the test fails.
- The verify command will not stop the test execution even when it fails. The test will continue and the failure will be logged. Verify is used to check conditions that are not critical.

What are the wait types in Selenium and what are the differences between them?

- **Implicit wait:** With the implicit wait, the driver waits a certain amount of time. If the element is not found after the time passes, a “NoSuchElementException” is thrown. It is bad practice to use Implicit wait and it should be avoided.
- Explicit wait: This wait instructs the driver to wait for a condition to be met. If the element cannot be found after the maximum time is exceeded, an “ElementNotVisibleException” is thrown.
- Fluent wait: The fluent wait defines the maximum amount of time the web driver waits for a condition to be met, as well as the frequency with which it checks the condition before throwing an “ElementNotVisibleException” exception.

Why is implicit wait not recommended?

The implicit waits only work on find element(s) methods and return either element found or not found - after timeout. Also, an element may be present in the DOM, but not fully loaded. ImplicitWait will still return because that element is present. If you need to check for the absence of an element, you must always wait until timeout.

Why is it not recommended to use *thread.sleep()*?

If you the Sleep to wait for an element to load, you will have to wait the entire defined period of the time. So, for instance, if the sleep is set to 5000 Milliseconds (5 seconds), and the element is loaded in 2 seconds, the script will always wait for the additional unnecessary 3 seconds. This may not seem much for one test, but it will increase the test execution if *thread.sleep()* is used in many tests.

Selenium provides waits which have this exact purpose - wait for an element to load. These should be used instead of *thread.sleep().*

Why is it not recommended to use both implicit and explicit wait?

According to the Selenium official documentation, mixing both types of waits can lead to **unpredictable wait times**.

What are some Selenium exceptions?

- NoSuchElementException: The element is present in the DOM, but it is not visible on the page and cannot be interacted with
- NoSuchFrameException: It is usually thrown when the frame you are trying to find does not exist
- NoSuchWindowException: Similar to NoSuchFrameException, but it applies to windows or tabs

- NoAlertPresentException: The alert cannot be found. It is usually caused by Selenium being too fast and can be solved with a wait
- InvalidSelectorException: The selector is not valid or is syntactically incorrect. It usually applies to XPaths
- ElementNotVisibleException: Selenium tries to find an element that is not visible within the page
- ElementNotSelectableException: It usually means that the element is present on the page, but cannot be selected
- TimeoutException: It is thrown of a command fails to complete in enough time
- NoSuchSessionException: This exception may happen if you are trying to call a method after the browser was closed (with *driver.quit()*)
- StaleElementReferenceException: an object for a particular web element was created in the program, but the element is no longer present on the web page

What is Selenium Grid?

Selenium grid is a tool that allows the WebDriver scripts to run in parallel on multiple machines, using different browser versions and configurations

What are some factors to consider for estimating a Selenium automation project?

- Calculating the complexity (size can be estimated using the Fibonacci sequence)
- Identifying the build tools
- Identifying the input and output parameters
- Consider the existing test suites and test cases that will be automated

Read more:

https://www.guru99.com/locators-in-selenium-ide.html

https://www.softwaretestingmaterial.com/challenges-and-limitations-of-selenium-webdriver/

https://www.lambdatest.com/blog/13-reasons-why-selenium-webdriver-should-be-your-first-choice-for-automation-testing/

https://testguild.com/what-to-automate/

https://martinfowler.com/bliki/TestPyramid.html

https://sqa.stackexchange.com/questions/18342/how-to-handle-dynamic-changing-ids-in-xpath

https://ultimateqa.com/selenium-webdriver-implicit-explicit-waits-explicit-waits/

https://www.softwaretestingmaterial.com/difference-between-assert-and-verify/

https://blogs.perficient.com/2016/01/13/hard-and-soft-assertions-in-selenium/

https://www.guru99.com/implicit-explicit-waits-selenium.html

https://ultimateqa.com/automation-patterns-antipatterns/

https://blog.testproject.io/2019/03/11/how-to-select-the-right-test-automation-tool/

https://www.tutorialspoint.com/what-is-selenese