



I'm not robot



Continue

## Civics test questions and answers 2018

Leave content to prevent post-traumatic stress disorder, is it helpful to provide psychotherapy to anyone who has been exposed to significant trauma? Most often asked automation testing interview questions for beginners and Advanced level candidates: Test automation plays a very important role throughout the software lifecycle. In most cases, we focus only on tool-specific questions while preparing for automation testing. However, we should also consider the fact that learning and knowledge of the instrument are simply average and that is not the ultimate goal. Therefore, as we prepare for the automation tester interview, we need to consider Automation as a whole and focus on the framework and steps involved. We all know that software testing is a very important part of software development. However, with fast-growing software development methodology and environments, it is difficult to manually test everything for a limited time, with cost limits. Thus, automation testing is rapidly growing as the market accelerates up the pace of development. This tutorial contains top interview questions about automation testing. I have tried to quote short and fast questions that are very much specific to automation as a whole and are not specific to any tool. Top 39 Automation Testing Interview Questions We have covered basic test automation questions as well as some more accurate questions from expert-level candidates with up to 2-5 years of experience. Q #1) What is automation? Answer: Automation is any activity that can reduce people's efforts. Q #2) What is automation testing? Answer: The process of using special software tools or scripts to perform tests on tasks such as data entering, completing test steps and comparing results, etc. is known as automation testing. Q #3) What can all things automate? Answer: Regression test suiteSmoke/Sanity test suiteBuild deploymentTest data creationAutomatization behind GUI such as testing APIs and methods. Q #4) When is automation useful? Answer: Automation testing is useful in the following cases:a) regression testing. If an error correction or a new module is applied, we need to make sure that the function that has already been applied or changed is not affected. In this case, we will end the regression test several times. For example: after each change request or error correction, after each iteration of a step-by-step development approach, etc.b) Non-functional testing: Testing the non-functional aspects of the application. For example, load testing or performance testing, etc. is very difficult for people to monitor and analyze.c) Complex computational checks or test scenarios that are prone to human error.d) Repeatedly performing the same tests: Sometimes we need to run the same set of data from a test case, or after each tier version or multiple combinations of hardware, software, or both. Automation test cases in the above scenarios contribute to the test speed and minimise human error. Q #5) How do I identify test cases suitable for automation? Answer: Identifying test cases suitable for automation is the most important step towards automation. Q #6) Can you achieve 100% automation? Answer: 100% automation would be difficult to achieve because there would be a lot of edge test cases and in some cases that are rarely performed. Automate cases that aren't executed that often don't add value to an automated set. Q #7) How to decide on a tool that one should use automation to test their projects? Answer: To identify the tool for automation testing your project.a) understand your project requirements thoroughly and identify the testing scenarios that you want to automate.b) Look for a list of tools that support your project requirements.c) Define your budget automation tool. Select your budget.d tools) Identify whether you already have skilled resources for the tools. If you don't have the necessary skilled resources, identify the costs of training existing resources or recruiting new resources.e) Now compare the basic criteria of each tool, such as: How easy is it to develop and maintain tool scripts? Can a non-technical person also carry out test cases with low training? Does the tool support a variety of platforms, such as web, mobile, desktop, etc., based on the requirements of your project? Does the tool have a test reporting feature? If not, is it an easy-to-set tool? How is a browser-to-browser support tool for web-based applications? How many test types can this tool support? How many languages does the tool support? Once you've compared the tools, select a tool that meets your budget and supports your project requirements, and give you more benefits based on the basic criteria mentioned above. Q #8) Currently I do not have any automation in place for my project, but now I want to implement automation, what would be my steps? Answer: First, determine which types of testing/cases you want to automate. Identify the Tool/ORMe Framework Create utility files and environmental files. Run scriptingT recognize and work with reporting. Allocate time for script improvement and maintenance. The steps needed to test the project are: understanding the advantages and disadvantages of automation testing and identifying test scenarios that are suitable for automation. Select the automation tool that is best suited to automate identified scenarios) find a tool expert to help set up the tool, and the environment you need to fill out test incidents using the tool. Train the team so that they can write scripts in the programming language supported by the tool. Create a test framework or identify an existing one that meets your requirements. Write to the operating system, browsers, equipment, etc. Write programming scripts for manual testing to convert them to automatic test cases. Report the status of the test case by using the tool reporting function. Manage scripts for ongoing changes or new features. Q #9) How do I decide which tool you need to use? Answer: The conclusion of which tool is most suitable for the project requires a lot of brainstorming and discussions. Q #10) Once you have identified the tool what would be your next steps? Answer: If we finish the tool, our next step would be to design a framework. Q #11) What is the framework? Answer: The framework is a set of structure for the entire automation kit. It is also a guideline that, if followed, can lead to a structure that is easy to maintain and repair. These steps are as follows:Coding standards Managing and handling test data Manage and handle elements (QTP object repository)Handling environmental files and attributes file>Data-handling logsQ #12) What are the attributes of a good framework? Answer: Features are as follows:Modular: The framework should be customizable with changes. Testers should be able to modify scripts according to the environment or change the login information. Reusable: Common methods or utilities should write a common file that is available to all scripts. Consistent: The kit must be written in a single format, following all accepted encoding domains. Independent: Scripts should be written in such a way that they are independent of each other. If one test fails, it should not hold back the remaining test cases (unless it is a sign-in page)Log: It is good to have the logging functionality applied within the framework. This would help if our scripts last longer (eg night mode) if the script does at some point, having a log file will help us discover the location with the type of error. Reporting: It is good to have the reporting function automatically integrated into the framework. Once scripting is done, we can send the results and reports by email. Integration: The automation framework should be such that it is easy to integrate with other applications, such as continuous integration or automatic script execution as soon as the build is deployed. Q #13) Can you do without a framework? Answer: frameworks are guidelines, not mandatory rules, so we can do without a framework, but if we create and follow it, it would be easy to implement and maintain it. Q #14) What are the different types of automation tool you are aware of? Answer: Open source tool such as Selenium, JMeter, etc. Paid tools like QTP, Load Runner, Ranorex, RFT and Rational Robot.Q #15) What is the general framework? Answer: Normally, the structure should be - (This would be different from project project)src (source) folder with actual test scripts. lib (library) folder with all libraries and common methods. Class folder with all class (in a suitcase using java). Log folder with log file(s). A file folder that has all the Web element IDS. A contains files that contain URL, environment, and sign-in information. Q #16) Where do you store information such as URL, username, password? Answer: This information should always be stored in a separate file. Q #17) Why would you want to keep such information in a separate file and not directly from the code? Answer: URL, Login and passwords are such fields that are used very often and they change according to the environment and authorization. If we encode this code, we need to change it in every file with its own reference. If there are more than 100 files, which in turn can cause errors. So this information is stored in a separate file, so the update becomes easy. Q #18) What are the different types of frameworks? Answer: Different types of frameworks include:Keyword-driven frameworkData-Driven frameworkHybrid frameworkLinear ScriptingQ #19) Can you say some good coding practices during automation? Answer: Some good coding practices are as follows:Add relevant comments. Identify the reusable methods and write it in a separate file. Follow language-specific coding conventions. Manage test data in a separate file. Run your scripts regularly. Q #20) Any test that you think should not be automated? Answer: Tests that are rarely completed. Investigative testingUsability testingTest, which is performed quickly if it is done manually. Q #21) Do you think tests can only be conducted at the user interface level? Answer: Today, as we move into Agile mode, testing is not limited to the UI layer. Early feedback is an imperial, agile project. If we focus only on the UI layer, we actually wait until the UI is developed and available for testing. Rather, we can test even before the UI is actually developed. We can directly test APIs or methods using tools such as cucumber and FiNesse.In in this way, we provide feedback a lot early and test before developing the user interface. After this approach will help us test only the GUI aspect of small cosmetic changes or some validations of the UI and will help developers, giving more time to fix errors. Q #22) How do I choose which automation tool is best for you? Answer: Choosing an automation tool depends on a variety of factors, such as: the application we want to automate. Control overheads such as cost and budget. Time to learn and implement the tool. Restriction toolQ #23) What do you think will keep testers back to make automation? Is it possible to get over it? Answer: The main obstacle for testers is to learn programming/coding if they want to automate. Since testers do not code, adapting to coding is a bit tricky for testers. We're going to get through this, working with developers on automation. Given that it is the responsibility of the whole team, not just the testers. Giving dedicated time and focus on automation. Getting the right management support. You can save these automation testing interview questions in pdf and print for further reading. Q #24) What is the automation testing framework? Answer: The framework is generally a set of guidelines. The automation testing framework is called guidelines, assumptions, concepts and coding practices to create a filling environment where experiments are automated. The automation testing framework is responsible for the creation of test harnesses, the mechanism of which is linked to the application being tested, withdraws input from the file, performs test cases and generates test performance reports. The automation testing framework should be independent of the application and should be easy to use, modify or expand. Q #25) What are the important modules for automation testing framework? Answer: Important modules for automation testing framework are: Test Approval Tool: This tool provides confirm applications to test the expected values of the application to be tested. Such as: Testing, Junit, etc. Data setup: Each test case must take user data from either the database or file or the embedded test script. The data module for frameworks should take care of the use of test scripts and global variable data. Create Management Tool: The framework must be created and deployed to create test scripts. Continuous integration tool: CICD (continuous integration and continuous development) is available, a continuous integration tool is needed to integrate and implement changes in the framework of each iteration. Reporting Tool: A reporting tool is required to create a readable report after you have performed a test case to better view steps, results, and errors. Logging tool: The logging tool within the framework helps you better debug errors and errors. Q #26) Explain some automation testing tools. Answer: Some famous automation testing tools are explained below:(i) Selen: Selen is a test framework for testing web application automation. It supports multiple browsers and is OS independent. Selen also supports various programming languages such as Java, C#, PHP, Ruby and Perl, etc. Selen is a collection of open source libraries that can be used to develop additional test frameworks or testing scripts to test web-based applications. (ii) UFT: Unified Functional Testing is a functional testing tool. It offers a variety of features, such as APIs, web services, etc., and also supports multiple platforms, such as desktops, web and mobile. UFT scripts are written in a visual basic scripting language. (iii) Appium: Appium is an open source mobile application testing tool. It is used to automate testing of cross-platform, native, hybrid and web-based mobile applications. Appium automates any mobile app from any language that is apes and dls. Appium is based on client-server architecture and has evolved from selenium. (iv) Cucumber: Cucumber is an open source behavioural development tool. It is used for web-based application automation testing and supports such languages as ruby, java, scala, groovy, etc. Cucumber reads the filling specification, which is written in plain text, and tests the test application for these specifications. For the cucumber to understand scenarios as plain text, we need to follow some basic syntax rules called Gherkin. (v) TestComplete: TestComplete is a licensed automated UI testing tool to test the application on different platforms, such as desktops, web, mobile, etc. This gives you the flexibility to register a test case in one browser and run it in multiple browsers and thus supports cross browsers testing. TestComplete is a built-in object detection algorithm that uniquely identifies the object and stores it in the repository. Q #27) What are the different types of testing framework techniques? Answer: There are four types of automation testing framework techniques. These are:i) Modular Testing Framework: This framework is based on the concept of abstraction. Within this framework, the tester creates scripts for each module in the application test separate, and then these scripts are combined in hierarchical order to create large test cases. This creates an abstraction layer between modules, so the changes to the test scripts in one module do not affect any other module. Advantages of this framework: Easier maintenance and scalability of test cases. It is easier and faster to create test cases using already scripted modules. Disadvantages test cases are data embedded in them. Therefore, filling the same test script with different data is a major change at the script level. (ii) Data-driven testing framework:Within the data-driven framework, input data and expected output data corresponding to the input data shall be stored in a file or database, and the automated script shall run the same set of test stages for multiple data sets. With this framework, we can run multiple test cases where only input data differs and the execution steps are the same. Advantages: Reduces the number of test scripts needed to perform. We execute the same script several times with different data. Less coding to test automation. Greater flexibility to maintain and correct errors or improve functionality. Test data can also be generated before the automatic testing system is completed. Disadvantages: Only similar test cases with the same set of execution steps can be combined for multiple data sets. Different execution steps require a different test case. (iii) Keyword-based testing framework: this is an application-independent testing framework that uses data tables and explanatory keywords. Keywords explain the actions to be taken with the app you are testing, and the data input and expected output data. Keyword-based testing is an increase in data-driven testing. Advantages: You can use less encoding and the same script for multiple data sets. Automation expertise is not necessary to create a test case using existing operations. The same keywords can be used in several test cases. Disadvantages: Difficult to read, maintain and repair. Q #28) When do you prefer manual testing over automation testing? Answer: We prefer manual testing over automation testing in the following cases: the project is short-lived and writing scripts are time consuming and costly compared to manual testing. Flexibility is required. Automated test cases are programmed and operated in a specific manner. The usability test must be carried out. Applications/modules have just been developed and there have been no previous test cases. Ad-hoc or exploratory testing must be carried out. Q #29) Is automation testing a brisk methodology writing or not? Answer: Automation testing is useful for testing regression, smoke or peace of mind. All this type of testing of the traditional waterfall model happens at the end of the cycle and sometimes, if there are not many improvements in the application, we may not even do regression testing. Although in the case of an agile session, each iteration requires the execution of a regression test case when new features are added. The regression kit itself will also grow after each sprint, as the next sprint regression kit will have to be added to the functional test cases of the current sprint module. Thus, automation testing agile methodology is very useful and helps achieve maximum test coverage in less time sprint. Q #30) List some of the advantages and disadvantages of automation testing. Answer: Benefits: Less human resourcesReusabilityMore test coverage less time Reliability Parall pre-completion test casesFastDisadvantages: development and maintenance time is more. The tool's cost-effective resources are required. Environment setupTest script debugging is a problem. Q #31) List some of the advantages and disadvantages of manual testing. Answer: Benefits: No environment setup required. Programming knowledge is not required. Recommended dynamically changing human observation power to detect more errors. The cost of short-term projects is lower. FlexibilityDisadvantages: Tricky to make complex calculations. Reusability Time, taking great risks to people's mistakes or mistakes. More human resources are needed. Q #32) Can we test automation without a framework? If so, why do we need a framework? Answer: Yes, we can do automation testing even without using the framework. We can easily understand the tool we use for automation and program steps in the programming language that the tools support. If we automate test cases without a framework, then there is no consistency in programming scripts in test cases. The framework is necessary to provide guidelines that everyone must follow in order to maintain legibility, reusability and consistency of test scripts. The framework also provides one common basis for reporting and logging functionality. Q #33) How do I automate application core sign-in test cases? Answer: Assuming that the automation tool and framework are already in place of the test environment. Basic Login testing: Understanding project requirement: The sign-in feature has a user name text box, a password text box, and a sign-in button. Test Scenarios: Possible test scenarios for the login feature are: Blank user name and passwordInvalid user name and passwordA valid user name and invalid passwordValid user name and passwordPrepared data entry file data that corresponds to each scenario. Start the tool from the program. Identify the user name field, password field, and sign-in button. For each test scenario, obtain data from the data file and enter the relevant fields. The program clicked on the login button after entering the data. Confirm the error message by using negative scenarios and success message by using positive scenario test script approvals. Start the test set and create a report. Q #34) Is automation testing black timing testing or white box testing? Answer: Automation testing is mostly a black box testing as we just program steps that the manual tester performs the application test without knowing the low design or code application. Sometimes, automatic test scripts require access to database details that are used in the application you are testing or other encoding details, and therefore this may be a white box test type. Therefore, automated testing can be both black or white box type testing depending on the scenarios in which automation occurs. Q #35) How many test cases do you have automated daily? Answer: Well, the number depends on the complexity of the test cases. When message was limited, I was able to automate 5-6 test cases per day. Sometimes I was able to automate only one pilot case for complex scenarios. I also distributed my test cases to different components such as, take input, do calculations, control output, etc. very complex scenarios and has taken 2 or more days. Q #36) What factors determine the effectiveness of automation testing? Answer: Some of the factors that determine automation testing are: Time saved running scripts in manual execution test cases. Defects FoundTest Coverage or code coverageAdding time or development timeTimeTime or development timeTimeRemates stabilityTest reusability quality testability alq #37) Which test cases can be automated? Answer: The types of test boxes that can be automated are:(i) smoke test boxes: smoke tests are also called a building test. Smoke test cases are triggered every time a new phase is released to check the health of the build to perform tests. (ii) Regression test cases: regression test is a test to ensure that previously developed modules function as expected after adding a new module or correcting the error. Regression test cases are very important in a graduated sisalysystem, where a new function is added at each increment phase. In this case, the regression test shall be carried out at each stage. (iii) Complex calculation test cases: test cases involving some complex calculations to check the field of an application fall into this category. Complex calculation results are more prone to human error, so they produce accurate results when automated data is made. (iv) Data-driven test cases: test cases with the same steps and carried out several times with data changes are referred to as data-driven test cases. Automatic testing of such test cases is fast and cost-effective. (v) Non-functional test cases: test incidents, such as test and performance tests, require a simulated environment with multiple users and several combinations of hardware or software. Manual setting up multiple environments is impossible for each combination or number of users. Automated tools can easily create this environment to perform non-functional tests easily. Q #38) What are the phases of automation testing lifecycle? Answer: Phase automation testing lifecycle included: the decision to perform automation testing. Identify and read about the automation tool. Define the scope of automation testing. Design and develop a test suite. Test executionCare test scripts. Q #39) What is an automated test script? Answer: An automated test script is a short program that is written in a programming language to make a set of instructions for the test application to check whether the application is compliant. This program at startup gives test results a pass or does not depend on whether the application is like any expectation. Conclusion These are the main issues that do not depend on the automation tool or programming language. Automation testing interviews also include tool and programming language-based questions, depending on the tool you've been working on. Most test automation interview questions are focused framework, so it is recommended that you create and understand your pilot framework thoroughly. When I'm interviewing, and the candidate has answered my question about the framework, I also prefer to ask a language-specific question (core java in my case). Questions start with the basics of java to write logic in some basic scenarios such as: How do you extract text from a given line? How do you extract the URL? Each website, each frame, the number of links and its contents change dynamically, how would you deal with it? How do you handle images and flash objects? How do I find a line of words? Answers to all these test automation interview questions are very much a specific tool/language that you use for automation. So before you go for an interview, brush your programming skills. If you didn't get the chance to create your own framework and someone else has created it, then make the time to understand it thoroughly before sitting on the interview. Some tips for automation testing interviews would be: Know your tool thoroughly. Learn locator techniques by your tool. Practice programming using the language you use to test automation. Learn your framework and its components. It is always useful if you have been involved in the development of your framework. So, be in-depth modules of the framework that you have worked on. I hope these questions would be useful for you to prepare for a test automation interview.