

KillTest

높은 품질, 더 나은 서비스



연습 문제

<https://www.killtest.kr>

1년간 무료 업데이트 제공

Exam : CSeT-F

**Title : A4Q Certified Selenium
Tester Foundation**

Version : DEMO

1.Which of the following statements is true about test automation metrics?

- A. Meaningful test automation metrics should be established quickly
- B. Meaningful test automation metrics are the same no matter the size of the project
- C. Meaningful test automation metrics are simply comparing relative test effort for manual versus automated tests
- D. Meaningful test automation metrics are part of demonstrating business value

Answer: D

Explanation:

Meaningful test automation metrics are part of demonstrating business value. Test automation metrics provide a quantitative way to measure the success of the test automation process, helping to identify areas where it is working well and areas where it could be improved. These metrics can be used to demonstrate the value of test automation to stakeholders, such as by showing the cost savings and increased productivity it can provide. Common test automation metrics include test execution time, test coverage, defect detection rate, and the cost of maintaining test scripts.

2.Consider the following test steps;

#	Test step	Expected outcome
1	Login with admin credentials	Admin logged in
2	Create user <i>jsmith</i>	User created
3	Create user <i>jsmith</i>	UserAlreadyExists error is shown
4	Delete user <i>jsmith</i>	User deleted
5	Log out from the system	Admin logged out

Which of the following function signatures is the best for keyword implementing test step number 1 ?

- A. def login ():
- B. def login_with_admin_credentials () :
- C. def login(username, password) :
- D. def create_user (username, user_email, password)

Answer: C

Explanation:

This function signature is the most concise and clear, and allows for the username and password to be passed to the function as parameters. This will allow for the login process to be automated without having to hardcode the username and password into the function.

3.Which of the following methods can find an element by its class name? Select two options

- A. find__element__by__css__selector
- B. find.__element__by__id.
- C. find_element_bY_xpath
- D. find_element_by_tag_name
- E. find_element_by_partial_link_text

Answer: A,D

Explanation:

The correct methods for finding an element by its class name are A. findelementbycssselector and D. findelementbytagname. The findelementbycssselector() method can be used to locate an element by its class name, while the findelementbytagname() method can be used to locate an element using the tag name of the element (such as 'div', 'span', 'a', etc).

4.Which of the following BEST explains how Selenium fits within the test automation architecture?

- A. Selenium supports designing test cases
- B. Selenium interfaces with the SUT
- C. Selenium supports implementing test cases
- D. Selenium is a tool, not part of the test automation architecture

Answer: B

Explanation:

Selenium interfaces with the System Under Test (SUT) to interact with it and run tests on it [1]. Selenium is a tool that is used to automate web browsers, which allows testers to create automated tests to verify the functionality of the SUT. It helps to reduce the need for manual testing and allows testers to test multiple browsers quickly and efficiently.

Selenium is a software tool that is used for automating web browsers. It interfaces with the System Under Test (SUT) by interacting with the web browser and the web application that is being tested. Selenium uses a set of APIs that allow you to control the browser and interact with elements on the web page, such as buttons, text fields, and links. By automating the browser and interacting with the web application in the same way that a user would, Selenium can be used to automate functional tests, regression tests, and other types of tests.

5.What is the correct sequence of standard Python log levels from lowest to highest?

- A. DEBUG, WARNING, CRITICAL, INFO, ERROR
- B. INFO, WARNING, ERROR, CRITICAL, BLOCKER
- C. NOTE, DEBUG, INFO, WARNING, ERROR
- D. DEBUG, INFO, WARNING, ERROR, CRITICAL

Answer: D

Explanation:

According to the official Python guide, the standard Python log levels from lowest to highest are DEBUG, INFO, WARNING, ERROR, and CRITICAL. These log levels are used to denote the severity of a log message, with DEBUG being the lowest level and CRITICAL being the highest. Each log level includes all the lower levels, so a message at the WARNING level will also be logged at the INFO and DEBUG levels.