1. What are the limitations of Selenium testing?

Some common limitations of Selenium testing are:

1. Selenium does not provide any Test Management feature.
2. Selenium lacks built-in Reporting feature.
3. In Selenium, we cannot test mobile applications and windows applications
4. Selenium has limited support for Image testing, like QR code or any image verifications.
5. Selenium doesn't have built-in object repository.
6. Selenium IDE supports only Firefox & Chrome browser.
7. Selenium doesn't support windows-based applications. It supports only web-based applications

2. What are the types of waits supported by WebDriver?

WebDriver offers several types of waits to handle synchronization issues:

- **Implicit Wait:** Implicit waits are applied globally to the WebDriver instance. When an implicit wait is set, WebDriver will wait for a specified amount of time for an element to be present in the DOM before throwing a NoSuchElementException. If the element is found within the specified time, the script continues executing without waiting further.
- **Explicit Wait:** Explicit waits allow the automation script to wait for a certain condition to be met before proceeding. The conditions can be based on various factors, such as the presence of a specific element, its visibility, or a certain text appearing on the page. With explicit waits, you can specify a timeout period and a condition that needs to be satisfied for the wait to end.
- **Fluent Wait:** Defines the maximum amount of time to wait for a condition, as well as the frequency with which to check the condition.

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

The major difference between driver.close() and driver.quit() is that:

driver.close() is used to close the browser or page currently which is having the focus.

driver.quit() is used to close all the windows/tabs and terminate the WebDriver session completely.

4. Can Captcha be automated?

It´s possible but automating CAPTCHA is challenging because CAPTCHAs are designed to be difficult for machines to solve while being relatively easy for humans. CAPTCHAs can be automated if you are able to decode the image using OCR (Optical Character Recognition). However, to do that, you'll need to write complex algorithms to sort out the image pattern

5. What does the switchTo() command do?

The switchTo() command is used to switch the focus or context to a different frame or window within a web application. It provides methods to navigate between different frames, windows, or pop-ups, allowing you to interact with elements inside them.

The switchTo() command is typically used after opening a new window or switching to an iframe within a web page

6. When do we use findElement() and findElements()?

The findElement() and findElements() methods are used to locate web elements on a web page

- **findElement:** A command to uniquely identify a web element within the web page.
- **findElements**: A command to identify a list of web elements within the web page.

7. Explain how API testing differs from unit testing.

API testing is a broader level of testing that ensures the seamless integration and functionality of the API in real-world scenarios and Unit testing goal is to test each individual module as it becomes available and to verify whether the module carries out the required functionality.

8. How do APIs work?

APIs serve as intermediaries that enable different software systems to communicate and interact with each other. They define a set of rules and protocols that govern how different software components can request and exchange information or functionality.

Overview of how APIs work:

- An API is defined by its specification
- APIs typically follow a client-server architecture
- API requests and responses are usually formatted using common data formats such as JSON or XML
- APIs often use HTTP methods (GET, POST, PUT, DELETE, etc.) to define the type of operation being performed.
- APIs expose specific endpoints or URIs that correspond to different functionalities or resources.
- APIs may require authentication and authorization to ensure that only authorized clients can access or modify data.

9. What are the advantages of API Testing?

Some the advantages of API Testing are:

- Early Detection of Issues: API testing allows for early detection of issues in the application's functionality.
- Faster and Efficient Testing: API testing focuses on the core functionalities and business logic, allowing testers to cover a large number of test cases efficiently

- Automation Potential: APIs are highly amenable to automation, and API testing can be easily automated using various testing frameworks and tools
- Integration and Interoperability Testing: API testing ensures that these integrations work seamlessly and that data exchange and interactions between systems are functioning correctly.
- Performance and Scalability Testing: API testing plays a vital role in assessing the performance and scalability of the application. By simulating various levels of load and stress, testers can measure the API's response time, throughput, and resource consumption.
- Enhanced Security Testing: By testing the API for security vulnerabilities, organizations can ensure that sensitive data and resources are adequately protected.

10. What are the advantages of UI Testing?

Some the advantages of UI Testing are:

- Identifying Functional Issues: UI testing helps identify functional issues or bugs in the user interface of an application
- Ensuring Correct Display: UI testing ensures that the application's user interface is rendered correctly across different devices, screen resolutions, and browsers.
- Validating User Interactions: UI testing allows testers to validate user interactions within the application. They can verify that the UI elements respond appropriately to user input, such as clicking buttons, entering text, selecting options, or navigating through menus. This helps ensure that the application's user interface is intuitive, interactive, and user-friendly.
- Enhancing Usability: UI testing helps improve the usability of the application by identifying areas where the user interface can be enhanced or optimized.
- Early Bug Detection: UI testing, when performed early in the development cycle, helps detect bugs or issues before they escalate into more significant problems.
-

11. What is Latency in API testing?

In API testing, latency refers to the period of time between when the request arrives at your server and when the client receives the first byte of the response.

12. List the most common status code that can response an API

Some the most common status code that can response an API are:

- 200 (OK)
- 201 (Created)
- 204 (No Content)
- 400 (Bad Request)
- 401 (Unauthorized)
- 403 (Forbidden)
- 404 (Not Found)
- 500 (Internal Server Error)
- 501 (Not Implemented)

- 503 (Service Unavailable)