



GroTechMinds
Making Life Better Everyday

50 Important Playwright with Python Interview Questions.



+91 80099 00785



contact@grotechminds.com



www.grotechminds.com



1. What is Playwright?

Playwright is an open-source automation framework that allows testing across modern web browsers. It supports multiple languages, including Python, and can automate Chromium, Firefox, and WebKit.

2. What are the main features of Playwright?

Some of the main features include multi-browser support, headless mode, auto-waiting for elements, capturing screenshots and videos, and support for handling multiple tabs and frames.

3. Why would you choose Playwright over Selenium?

Playwright provides better support for modern web applications, offering features like automatic waiting for elements, browser context isolation, faster execution, and native support for handling tabs, frames, and shadow DOMs.

4. What browsers does Playwright support?

Playwright supports Chromium (Google Chrome), Firefox, and WebKit (Safari).

5. What is the difference between Playwright and Puppeteer?

Playwright is designed to work with multiple browsers (Chromium, Firefox, WebKit), while Puppeteer is mainly focused on Chrome/Chromium. Playwright also has more advanced features like cross-browser support and better network management.

6. What is a BrowserContext in Playwright?

A BrowserContext in Playwright is an isolated, incognito-like environment where a browser instance can be executed without sharing cookies or cache with other contexts, useful for parallel tests.





7. Can you explain what auto-waiting in Playwright means?

Auto-waiting means Playwright automatically waits for elements to be ready (such as visible or interactable) before interacting with them, reducing the need for explicit waits in test scripts.

8. How does Playwright handle multiple tabs?

Playwright allows you to handle multiple tabs through the page object. You can create new tabs and interact with each tab by switching between the page instances.

9. What are selectors in Playwright?

Selectors are used to identify elements on the web page. Playwright supports various types of selectors, including CSS, XPath, text-based, and even custom selectors.

10. What is headless mode in Playwright?

Headless mode runs the browser without a graphical user interface (GUI). It is commonly used in automated testing to run tests faster.

11. How do you handle dropdowns and pop-ups in Playwright?

Playwright provides methods to handle dropdowns using `selectOption()` for dropdowns and `page.on()` event listeners for pop-ups and dialogs.

12. What is Page in Playwright?

A Page is a single tab or a single frame in the browser that Playwright interacts with. Each page is a fresh context for performing user actions or extracting data.

13. How do you capture a screenshot in Playwright?

Playwright allows capturing screenshots using the `screenshot()` method. It can capture the entire page or a specific element.





14. What is the use of the waitForSelector method?

waitForSelector() is used to pause the script until a specified element appears on the page. This is helpful to avoid timing issues.

15. What is Playwright's approach to handling iframes?

Playwright allows interaction with elements inside iframes using the frame() method to access and manipulate iframe content.

16. How does Playwright handle network interception?

Playwright provides APIs to intercept and modify network requests and responses using route(), setRequestInterception(), and on() methods.

17. How can you emulate mobile devices using Playwright?

Playwright supports device emulation using the playwright.devices[] method. It allows emulation of various devices such as iPhone or iPad with custom viewport sizes and user agent strings.

18. What is tracing in Playwright?

Tracing is a feature that allows you to record the browser's execution steps, capturing screenshots, and other data that can help debug failed tests.

19. Can you explain browser isolation in Playwright?

Browser isolation ensures that each BrowserContext runs in a separate browser session without sharing cookies, local storage, or session data, which is useful for parallel testing.

20. What are the benefits of headless browser testing?

Headless browser testing is faster since it skips rendering the GUI, making it ideal for CI/CD pipelines. It also reduces memory usage.





21. Can you explain parallelism in Playwright?

Playwright supports parallel execution by running tests in separate browser contexts or across different browser instances simultaneously, improving test speed.

22. How does Playwright handle file uploads?

Playwright uses `setInputFiles()` to handle file uploads by simulating the action of selecting files from the local file system.

23. What is the role of context in Playwright?

A context in Playwright is an instance of a browser that provides an isolated environment for running tests. Multiple contexts can be created to simulate multiple users.

24. What types of waits does Playwright support?

Playwright supports three main types of waits: implicit waiting (built-in auto-waiting), explicit waiting (using `waitFor()` methods), and hard waits (using timeouts).

25. How can you record tests in Playwright?

Playwright provides a CLI tool called `codegen` that can generate test scripts by recording user interactions on the browser.

26. How does Playwright handle authentication in tests?

Playwright allows you to handle authentication by storing cookies or tokens and reusing them in subsequent tests or contexts to avoid logging in repeatedly.

27. How can you simulate slow network conditions in Playwright?

Playwright allows you to simulate slow network conditions using `context.setNetworkConditions()` to throttle the network speed during testing.





28. What is the significance of beforeAll and afterAll hooks in Playwright?

beforeAll and afterAll hooks are used to set up and tear down resources before and after a suite of tests run, like initializing a browser or clearing data.

29. Can Playwright run on cloud-based services?

Yes, Playwright can be integrated with cloud-based services like BrowserStack, Sauce Labs, or Playwright's own cloud solution for scalable, distributed testing.

30. How does Playwright manage cookies?

Playwright allows you to get, set, or clear cookies using methods like `context.addCookies()` and `context.clearCookies()`.

31. What is the use of page.route() in Playwright?

`page.route()` is used to intercept network requests, allowing you to mock responses or modify requests during the test execution.

32. How do you handle authentication pop-ups in Playwright?

Playwright can handle basic authentication pop-ups by passing the credentials in the `page.authenticate()` method.

33. What is a web socket, and does Playwright support it?

A web socket is a protocol used for two-way communication between a client and a server. Playwright supports intercepting and testing web socket communications.

34. Can Playwright capture videos during test execution?

Yes, Playwright can capture videos by enabling the video recording feature when creating a new `BrowserContext`.

35. How does Playwright handle incognito mode?

Every new `BrowserContext` created in Playwright runs in incognito mode by default, ensuring session isolation.





36. What are the common use cases of Playwright in testing?

Playwright is commonly used for end-to-end testing, cross-browser testing, mobile emulation, headless testing, and UI regression testing.

37. What are the limitations of Playwright?

Playwright is relatively new, so it may have less community support compared to Selenium. It also has a learning curve for advanced features like network interception.

38. How can you handle timeouts in Playwright?

Playwright allows you to handle timeouts globally using `context.setDefaultTimeout()` or individually for actions like clicks or navigation.

39. What is the use of playwright.config file?

The `playwright.config` file is used to configure global settings like browser options, timeouts, retries, test folders, and other configuration parameters.

40. Can Playwright be used for API testing?

While Playwright is primarily for UI automation, it supports making HTTP requests and can be used for basic API testing using its request module.

41. What is a HAR file, and how does Playwright use it?

HAR (HTTP Archive) files capture a record of network interactions. Playwright can generate HAR files to record network activity and analyze performance issues.

42. How does Playwright handle Shadow DOM?

Playwright can handle elements inside Shadow DOM using built-in support for selecting shadow DOM elements via CSS selectors or custom locators.





43. How can Playwright be integrated with CI/CD tools?

Playwright can be integrated with CI/CD tools like Jenkins, GitHub Actions, CircleCI, etc., to run automated tests in the pipeline using headless browsers.

44. What is the role of fixtures in Playwright?

Fixtures in Playwright provide reusable components that are initialized and cleaned up automatically for test suites, such as browsers, contexts, or databases.

45. How does Playwright support localization testing?

Playwright allows you to set the locale and language preferences in the browser context to test websites with different language settings.

46. How do you monitor network performance using Playwright?

Playwright allows you to intercept and measure network request/response times using `page.on('request')` and `page.on('response')`.

47. What are the benefits of using Python with Playwright?

Python is known for its simplicity and readability, making Playwright with Python an easy choice for writing tests quickly and integrating with other Python-based tools.

48. How does Playwright handle SSL certificate errors?

Playwright can handle SSL certificate errors by configuring the browser context to ignore these errors using `ignoreHTTPSErrors`.

49. What is a viewport in Playwright, and how is it managed?

A viewport is the visible area of the browser window. You can set or adjust the viewport size in Playwright for different screen resolutions.





50. What is the difference between headful and headless mode in Playwright?

Headful mode runs the browser with its full graphical user interface (GUI), while headless mode runs it in the background without any GUI, improving performance and speed for automation.

46. How do you monitor network performance using Playwright?

Playwright allows you to intercept and measure network request/response times using `page.on('request')` and `page.on('response')`.

47. What are the benefits of using Python with Playwright?

Python is known for its simplicity and readability, making Playwright with Python an easy choice for writing tests quickly and integrating with other Python-based tools.

48. How does Playwright handle SSL certificate errors?

Playwright can handle SSL certificate errors by configuring the browser context to ignore these errors using `ignoreHTTPSErrors`.

49. What is a viewport in Playwright, and how is it managed?

A viewport is the visible area of the browser window. You can set or adjust the viewport size in Playwright for different screen resolutions.





GroTechMinds
Making Life Better Everyday

*Thank
you!*

FOLLOW FOR MORE



+91 80099 00785



contact@grotechminds.com



www.grotechminds.com