

■ 50 Most Asked JavaScript + Playwright Interview Q&A;

■ Q1. What is the difference between var, let and const?

■ Answer: var is function-scoped and can be redeclared. let and const are block-scoped. const cannot be reassigned.

■ Q2. What is closure in JavaScript?

■ Answer: Closure is a function that remembers variables from its outer scope even after the outer function has executed.

■ Q3. Difference between == and ===?

■ Answer: === compares values with type coercion. === compares value and type strictly.

■ Q4. What are promises?

■ Answer: Promises represent asynchronous operations with states: pending, fulfilled, rejected.

■ Q5. Why async/await is important in Playwright?

■ Answer: Playwright actions are asynchronous. async/await ensures sequential execution and proper waiting.

■ Q6. What is auto-waiting in Playwright?

■ Answer: Playwright automatically waits for elements to be ready before performing actions.

■ Q7. Difference between page.locator() and page.\$()?

■ Answer: locator() supports auto-wait and retries. page.\$() returns immediately without waiting.

■ Q8. What is Page Object Model (POM)?

■ Answer: POM is a design pattern where each page is represented as a class containing locators and methods.

■ Q9. How to handle multiple tabs in Playwright?

■ Answer: Use context.waitForEvent('page') to capture new tab and switch control.

■ Q10. How to run tests in parallel?

■ Answer: Configure workers in playwright.config.js.

■ Q11. What is fixture in Playwright?

■ Answer: Fixtures provide reusable setup and teardown logic for tests.

■ Q12. How to capture screenshot on failure?

■ Answer: Use test hooks or configuration to capture screenshots automatically on failure.

■ Q13. What is network interception?

■ Answer: It allows monitoring or modifying network requests using page.route().

■ Q14. What is headless mode?

■ Answer: Running browser without UI. Default in Playwright unless specified.

■ Q15. Difference between forEach and for...of?

■ Answer: forEach does not support await properly. for...of works better with async/await.

■ Q16. What is destructuring?

■ Answer: Extracting values from arrays or objects into variables.

■ Q17. What is spread operator?

■ Answer: Used to copy or merge arrays/objects using ...

■ Q18. What are hooks in Playwright?

■ Answer: beforeEach, afterEach, beforeAll, afterAll are used for setup and cleanup.

■ Q19. How to handle dropdown?

■ Answer: Use page.selectOption().

■ Q20. How to handle alerts?

■ Answer: Use page.on('dialog') event listener.

■ Q21. What is test.describe()?

■ Answer: Groups related tests together.

■ Q22. How to retry failed tests?

■ Answer: Set retries in configuration file.

■ Q23. What is cross-browser testing?

■ Answer: Running tests across Chromium, Firefox, and WebKit.

■ **Q24. What is CI/CD integration?**

■ Answer: Running tests automatically using tools like Jenkins or GitHub Actions.

■ **Q25. How to wait for API response?**

■ Answer: Use page.waitForResponse().

■ **Q26. Difference between synchronous and asynchronous?**

■ Answer: Synchronous blocks execution. Asynchronous allows non-blocking operations.

■ **Q27. What is callback function?**

■ Answer: Function passed as argument to another function.

■ **Q28. What is arrow function?**

■ Answer: Short syntax function using => and lexical this binding.

■ **Q29. How to parameterize tests?**

■ Answer: Use test.describe() with loops or test.each().

■ **Q30. How to debug Playwright tests?**

■ Answer: Use PWDEBUG=1 or page.pause().

■ **Q31. How to generate HTML report?**

■ Answer: Use npx playwright show-report.

■ **Q32. What is test runner in Playwright?**

■ Answer: Built-in runner to execute and manage tests.

■ **Q33. How to handle file upload?**

■ Answer: Use page.setInputFiles().

■ **Q34. How to handle iframe?**

■ Answer: Use frameLocator().

■ **Q35. What is expect in Playwright?**

■ Answer: Assertion library used to validate results.

■ Q36. What is toHaveText()?

■ Answer: Assertion to verify element text.

■ Q37. How to manage environment variables?

■ Answer: Use process.env or .env files.

■ Q38. What is API testing in Playwright?

■ Answer: Using request fixture to test backend APIs.

■ Q39. What is timeout in Playwright?

■ Answer: Maximum time Playwright waits before failing test.

■ Q40. How to handle dynamic elements?

■ Answer: Use locator with auto-waiting and proper selectors.

■ Q41. What is BDD in Playwright?

■ Answer: Using tools like Cucumber with Playwright for behavior-driven testing.

■ Q42. How to structure framework?

■ Answer: Use folders: tests, pages, utils, config.

■ Q43. What is test isolation?

■ Answer: Each test runs independently without shared state.

■ Q44. How to mock API response?

■ Answer: Use page.route() to intercept and mock.

■ Q45. What is global setup?

■ Answer: Runs before all tests using globalSetup in config.

■ Q46. How to handle cookies?

■ Answer: Use context.addCookies() and context.cookies().

■ Q47. Difference between locator and selector?

■ Answer: Locator is Playwright API abstraction. Selector is string query.

■ Q48. What is trace viewer?

■ Answer: Tool to debug tests with detailed execution trace.

■ Q49. How to reduce flaky tests?

■ Answer: Use proper waits, stable locators, retries.

■ Q50. Why Playwright over Selenium?

■ Answer: Better auto-waiting, faster execution, built-in parallelism and modern architecture.
