



1 Locators

getByRole	ARIA role, most resilient
getByText	Match visible text
getByLabel	Match a form control's label
getByPlaceholder	Match input placeholder text
getByTitle	Match an element's title attribute
getByAltText	Match an image's alt text
getByTestId	Match data-testid attribute
page.locator('css=...')	Raw CSS selector (escape hatch)
page.locator('xpath=...')	Raw XPath selector (last resort)

2 Locator Chaining & Filters

locator.locator()	Chains, finds within a locator
filter({ hasText })	Keeps matches containing text
filter({ has })	Keeps matches containing a locator
.or(other)	Matches either locator
first() / last() / nth()	Picks one from multiple matches

3 Waiting & Assertions

Auto-waiting	Built-in wait before every action
expect(locator)	Web-first, retrying assertion
toBeVisible()	Waits until element is visible
waitForTimeout()	Explicit fixed delay, avoid, flaky
waitForLoadState()	Explicit wait for page load state

4 Actions

click()	Waits for actionable, then clicks
dblclick()	Double-clicks the element
fill()	Sets value instantly, no keystrokes
pressSequentially()	Key-by-key typing, replaces .type()
press()	Sends a single key press
check() / uncheck()	Sets checkbox/radio state
selectOption()	Chooses dropdown option(s)

TIP

Prefer **getByRole** over CSS/XPath, it survives redesigns and mirrors how real users find elements. Run **--trace on** in CI for full failure recordings.

5 Test Setup & Config

Fixtures	Reusable per-test setup
Browser contexts	Isolated session state
beforeEach/afterEach	Per-test hooks
playwright.config.ts	Central project configuration

6 Network Control

waitForResponse()	Waits for a matching response
waitForURL()	Waits for a URL pattern
page.route()	Intercepts / mocks requests
page.unroute()	Removes a route handler

7 Debugging Tools

UI Mode	Time-travel debugging, watch mode, pick locator
Trace viewer	Full run recording, DOM and network
Codegen	Records actions into a script
PW Inspector	Step-through debug UI
--debug flag	Launches inspector on run

8 Execution & Beyond the UI

Parallelism	Runs test files across workers
Sharding	Splits suite across machines
Retries	Re-runs a failed test automatically
Cross-browser	One API, Chromium, Firefox, WebKit
API testing	Calls APIs directly, no browser needed

9 Must-Know JS/TS Concepts

async / await	Pauses until a Promise settles
Promise	Future value of an async operation
Arrow functions	Concise (x) => x function syntax
try / catch	Handles runtime errors gracefully
Destructuring	Unpacks values from objects/arrays
Spread / rest (...)	Expands or collects values
Optional chaining (?.)	Safe access to nested properties

