1. Browser & Context Handling

- chromium.launch()
  Launches a Chromium browser instance.
  Syntax: const browser = await chromium.launch({ headless: false });
- browser.newContext()
  Creates a new isolated browser context.
  Syntax: const context = await browser.newContext();
- context.newPage()
  Opens a new browser tab.
  Syntax: const page = await context.newPage();
- browser.close()
  Closes the entire browser.
  Syntax: await browser.close();
- context.close()
  Closes a specific browser context.
  Syntax: await context.close();

2. Page Navigation

- page.goto(url)
  Navigates to the given URL.
  Syntax: await page.goto('https://example.com');
- page.reload()
  Reloads the page.
  Syntax: await page.reload();
- page.goBack()
  Goes back to the previous page.
  Syntax: await page.goBack();
- page.goForward()
  Goes forward in history.
  Syntax: await page.goForward();
- page.url()
  Returns current URL.
  Syntax: await page.url();

3. Element Locators & Actions

- page.locator(selector)
  Locates an element.
  Syntax: const button = page.locator('text=Login');

- locator.click()
  Clicks on an element.
  Syntax: await locator.click();
- locator.fill()
  Fills text input.
  Syntax: await locator.fill('text');
- locator.type()
  Types text slowly.
  Syntax: await locator.type('hello');
- locator.hover()
  Hovers over element.
  Syntax: await locator.hover();
- locator.check()
  Checks checkbox.
  Syntax: await locator.check();
- locator.selectOption()
  Selects dropdown option.
  Syntax: await locator.selectOption('India');

4. Assertions

- expect(locator).toBeVisible()
  Asserts visibility.
  Syntax: await expect(locator).toBeVisible();
- expect(page).toHaveTitle()
  Checks page title.
  Syntax: await expect(page).toHaveTitle('Dashboard');
- expect(page).toHaveURL()
  Checks current URL.
  Syntax: await expect(page).toHaveURL(/dashboard/);
- expect(locator).toContainText()
  Checks substring text.
  Syntax: await expect(locator).toContainText('Success');

5. Waits & Synchronization

- page.waitForSelector()
  Waits for element.
  Syntax: await page.waitForSelector('#loader');
- page.waitForTimeout()
  Static wait.
  Syntax: await page.waitForTimeout(3000);

- page.waitForLoadState()
  Waits for load completion.
  Syntax: await page.waitForLoadState('networkidle');

6. Keyboard, Mouse & Uploads

- page.keyboard.press()
  Presses key.
  Syntax: await page.keyboard.press('Enter');
- page.mouse.click()
  Clicks coordinates.
  Syntax: await page.mouse.click(100, 200);
- page.setInputFiles()
  Uploads file.
  Syntax: await page.setInputFiles('#file', 'path/to/file.pdf');

7. Screenshots & Videos

- page.screenshot()
  Takes screenshot.
  Syntax: await page.screenshot({ path: 'shot.png' });
- context.newPage({ recordVideo })
  Records video.
  Syntax: await context.newPage({ recordVideo: { dir: 'videos/' } });

8. Network & API Handling

- page.route()
  Intercepts requests.
  Syntax: await page.route('**/api/*', route => route.abort());
- page.request.get()
  Performs GET request.
  Syntax: const res = await page.request.get('url');

9. Dialogs, Frames & Popups

- page.on('dialog')
  Handles alert.
  Syntax: page.on('dialog', d => d.accept());
- page.frame()
  Access frame.
  Syntax: const frame = page.frame({ name: 'iframeName' });
- page.waitForEvent('popup')
  Waits for popup.
  Syntax: const popup = await page.waitForEvent('popup');

10. Cookies & Storage

- context.addCookies()
  Adds cookies.
  Syntax: await context.addCookies([{ name: 'token', value: 'abc' }]);
- context.cookies()
  Fetches cookies.
  Syntax: await context.cookies();
- context.clearCookies()
  Clears cookies.
  Syntax: await context.clearCookies();

11. Tracing & Debugging

- context.tracing.start()
  Starts trace.
  Syntax: await context.tracing.start({ screenshots: true });
- context.tracing.stop()
  Stops trace.
  Syntax: await context.tracing.stop({ path: 'trace.zip' });
- page.pause()
  Pauses for debugging.
  Syntax: await page.pause();

12. Hooks & Fixtures

- test.beforeAll()
  Runs before all tests.
  Syntax: test.beforeAll(async () => {});
- test.beforeEach()
  Runs before each test.
  Syntax: test.beforeEach(async ({ page }) => {});
- test.afterEach()
  Runs after each test.
  Syntax: test.afterEach(async () => {});
- test.afterAll()
  Runs after all tests.
  Syntax: test.afterAll(async () => {});