

# ■ Playwright Cheat Sheet (TypeScript / JavaScript)

## ■ Setup & Test Runner

Commands to set up Playwright and run tests from CLI.

```
npm init playwright@latest npx playwright test npx playwright test --headed npx  
playwright test tests/example.spec.ts npx playwright test -g "test name"
```

## ■ Test Structure

Organize tests using describe, beforeEach, afterEach, etc.

```
import { test, expect } from '@playwright/test'; test.describe('My Suite', () => {  
  test.beforeEach(async ({ page }) => { await page.goto('https://example.com'); });  
  test('basic test', async ({ page }) => { await expect(page).toHaveTitle(/Example/); });  
  test.afterEach(async ({ page }) => { await page.close(); }); });
```

## ■ Navigation

Methods to navigate across pages.

```
await page.goto('https://example.com'); await page.reload(); await page.goBack(); await  
page.goForward();
```

## ■ Locators & Selectors

Find elements using CSS, text, XPath, or position.

```
const btn = page.locator('button#submit'); const link = page.locator('text=Login');  
const title = page.locator('//h1[contains(text(),"Welcome")]'); await  
page.locator('button').first().click(); await page.locator('button').nth(2).click();
```

## ■■ Interactions

Simulate user actions like click, type, check, select, hover.

```
await page.click('text=Submit'); await page.fill('#username', 'myUser'); await  
page.type('#password', 'myPass'); await page.check('#accept'); await  
page.uncheck('#opt-out'); await page.selectOption('#country', 'India'); await  
page.hover('.menu-item');
```

## ■ Waits

Wait for elements, not just timeouts (best practice).

```
await page.waitForTimeout(2000); await page.waitForSelector('#message'); await  
expect(page.locator('#message')).toBeVisible();
```

## ■ Assertions

Validate page state, elements, text, and conditions.

```
await expect(page).toHaveTitle(/Example/); await expect(page).toHaveURL(/.*example/);
await expect(page.locator('#welcome')).toContainText('Hello'); await
expect(page.locator('#logo')).toBeVisible(); await
expect(page.locator('#submit')).toBeEnabled();
```

## ■ Frames & New Tabs

Handle iframes and new browser tabs.

```
const frame = page.frame({ name: 'myFrame' }); await
frame?.locator('#inside-frame').click(); const [newPage] = await Promise.all([
context.waitForEvent('page'), page.click('a[target="_blank"]'), ]); await
newPage.waitForLoadState();
```

## ■ API Testing

Use Playwright's request fixture to test APIs.

```
test('API test', async ({ request }) => { const response = await
request.get('/users/1'); expect(response.status()).toBe(200); const body = await
response.json(); expect(body.name).toBe('Leanne Graham'); });
```

## ■ Screenshots & Videos

Capture evidence during test runs.

```
await page.screenshot({ path: 'screenshot.png', fullPage: true }); await
page.video().saveAs('test-video.webm');
```

## ■ CLI Tips

Helpful CLI commands for debugging & trace analysis.

```
npx playwright install npx playwright test --debug npx playwright show-trace trace.zip
```