PLAYWRIGHT AUTOMATION CHEAT SHEET

Installation

```
npm i playwright
```

Navigating to a URL

```
await page.goto('https://www.example.com');
```

Assertions

```
// Expect assertion
await expect(element).toBeVisible();
```

Launching a Browser

```
const { chromium } = require('playwright');
(async () => {
  const browser = await chromium.launch();
  const page = await browser.newPage();
  // Automate actions here...
  await browser.close();
})();
```

Debugging

```
// Debug
await page.pause();


Npx playwright test --debug
```

Selectors

```
text=(Text content)
css=(CSS selector)
xpath=(XPath expression)
id=(Element ID)


const element = await page.$('selector');
const elements = await page.$$('selector');
```

Interactions

```
// Click
await element.click();
// Fill input
await element.fill('text');
// Check if element is visible
const isVisible = await element.isVisible();
// Get element text
const text = await element.textContent();
```

Screenshots and Videos

```
// Screenshot
await page.screenshot({ path: 'screenshot.png' });
// Video
const context = await browser.newContext({
  recordVideo: { dir: 'videos/' }
});


await expect(element).toHaveText('Hello');
```

Browser Context and Cookies

```
// New context
const context = await browser.newContext();
// Get/Set cookies
const cookies = await context.cookies();
await context.addCookies([{ name: 'foo', value: 'bar' }]);
```

Timeouts and waiting

```
// Set timeout
page.setDefaultTimeout(5000);
// Wait for element
await page.waitForSelector('selector');
```

Parallel Execution

```
const { chromium } = require('playwright');
(async () => {
  const browser = await chromium.launch();
  const context = await browser.newContext();
  // Create array of promises
  const promises = [
    context.newPage(),
    context.newPage(),
    context.newPage(),
 ];
// Execute promises in parallel
  const pages = await Promise.all(promises);
// Automate actions on parallel pages...
  await browser.close();
})();
```

GetBy Selectors

```
// Select by text content
const element = await page.getByText('Click me');
// Select by label text (for input elements)
const input = await page.getByLabel('Username');
// Select by placeholder text (for input elements)
const input = await page.getByPlaceholder('Enter your name');
// Select by alt text (for image elements)
const image = await page.getByAlt('Product Image');
// Select by title text (for elements with a title attribute)
const link = await page.getByTitle('Learn More');
// Select by role (for accessible elements)
const button = await page.getByRole('button');
// Select by test id (for elements with a data-testid attribute)
const component = await page.getByTestId('product-card');
// Select by aria-label (for accessible elements)
const nav = await page.getByAriaLabel('Main Navigation');
```

Handling Popups and Modals

```
// Wait for and dismiss an alert
const alert = await page.waitForEvent('dialog');
await alert.dismiss();
// Wait for and accept a confirm dialog
const confirm = await page.waitForEvent('dialog');
await confirm.accept();
// Wait for and handle a prompt dialog
const prompt = await page.waitForEvent('dialog');
await prompt.accept('My input value');
```

Working with Frames and Shadows

```
// Switch to a frame
const frame = await page.frame({ url:
/\/iframe\.html/ });
const element = await frame.$('selector');
// Switch to a shadow root
const shadow = await element.shadowRoot();
const shadowElement = await shadow.$('selector');
```

Emulating Devices and Viewports

```
// Emulate a specific device
await page.emulate({
  viewport: { isMobile: true, hasTouch: true },
  userAgent: 'Mozilla/5.0 (Linux; Android 9; Pixel 3)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.61 Mobile
Safari/537.36'
});
// Set a custom viewport size
await page.setViewportSize({
  width: 1280,
  height: 720
});
```

This cheat sheet covers the basic commands and practices for launching browsers, navigating pages, interacting with elements, assertions, screenshots/videos, context management, timeouts, debugging, and parallel execution. You can refer to the official **Playwright documentation** for more advanced topics and examples.