

# Playwright Cheat Sheet (Core Functions)

## Test Structure

```
'test(name, callback)' – Define a test case  
'test.describe(name, callback)' – Group related tests  
'test.beforeEach / test.afterEach' – Hooks before/after each test
```

## Page Navigation

```
'page.goto(url)' – Navigate to a page  
'page.reload()' – Reload the page  
'page.url()' – Get current URL
```

## Selectors & Locators

```
'page.locator(selector)' – Create a locator  
'page.getByRole(role, options)' – Find by accessible role  
'page.getText(text)' – Find by visible text  
'page.getByPlaceholder(text)' – Find by input placeholder  
'page.getLabel(label)' – Find by input label
```

## Actions

```
'locator.click()' – Click element  
'locator.fill(value)' – Fill input  
'locator.type(value)' – Type text (like a real user)  
'locator.press('Enter')' – Simulate key press  
'page.check(selector) / page.uncheck(selector)' – Checkbox actions  
'page.selectOption(selector, value)' – Dropdown selection
```

## Assertions

```
'expect(locator).toBeVisible()' – Check if element is visible  
'expect(locator).toHaveText('Hello')' – Check if element has specific text  
'expect(locator).toHaveValue('admin')' – Check if element has specific value  
'expect(page).toHaveURL(/.*dashboard/)' – Check if page URL matches pattern  
'expect(page).toHaveTitle(/Playwright/)' – Check if page title matches pattern
```

## Screenshots & Debug

```
'page.screenshot({ path: 'screenshot.png' })' – Take screenshot  
'page.pause()' – Pause test for debugging
```

## API Testing

```
`request.get(url)` / `request.post(url, data)` – API requests  
`expect(response.ok()).toBeTruthy()` – Assert API response
```

### ■ Example Test

```
import { test, expect } from '@playwright/test';

test('user can log in successfully', async ({ page }) => {
  await page.goto('https://the-internet.herokuapp.com/login');
  await page.locator('#username').fill('tomsmith');
  await page.locator('#password').fill('SuperSecretPassword!');
  await page.getByRole('button', { name: 'Login' }).click();
  await expect(page).toHaveURL(/.*secure/);
  await expect(page.locator('.flash.success')).toBeVisible();
  await page.screenshot({ path: 'login-success.png' });
});
```