Scrolling

**1. Using page.mouse.wheel()**

You can use the mouse.wheel() method to scroll programmatically by simulating mouse wheel events.

**Example:**

javascript

Copy code

// Scroll down

await page.mouse.wheel({ deltaY: 100 }); // Scrolls down by 100 pixels

**2. Using page.evaluate() for Scroll**

You can execute JavaScript in the context of the page to scroll to specific elements or positions.

**Example: Scroll to a Specific Element**

javascript

Copy code

const elementLocator = page.locator('div#target'); // The element you want to scroll to

await page.evaluate(element => element.scrollIntoView(), await elementLocator.elementHandle());

**Example: Scroll to a Specific Position**

javascript

Copy code

await page.evaluate(() => {

window.scrollTo(0, 500); // Scrolls to 500 pixels down from the top

});

**3. Using page.keyboard.press() for Scrolling**

You can simulate keyboard presses to scroll the page, especially useful for navigating using the keyboard.

**Example: Scroll Down Using Arrow Keys**

javascript

Copy code

await page.keyboard.press('ArrowDown'); // Scrolls down

await page.keyboard.press('PageDown'); // Scrolls down a full page

**4. Using the scrollIntoView Method**

This method can be used on the locator to scroll a specific element into view.

**Example:**

javascript

Copy code

const elementLocator = page.locator('div#target'); // The element you want to scroll into view

await elementLocator.scrollIntoView(); // Scrolls the element into the visible area

**5. Scroll Using Touch Screen Events (for mobile testing)**

For mobile devices, you might want to simulate touch gestures.

**Example: Simulating a Touch Scroll**

javascript

Copy code

await page.touchscreen.tap(100, 100); // Tap to focus

await page.touchscreen.swipe(100, 100, 100, 300); // Swipe down