🔑 Funciones Clave de Playwright en JavaScript

1. Funciones de Interacción y Acciones (Clase Page / Locator)

Estas son las funciones que se usan para simular la interacción de un usuario con la página. Se usan generalmente en un objeto locator o directamente en la page.

Función	Descripción	Ejemplo
locator(selector)	Crea un localizador para seleccionar un elemento en el DOM.	page.locator('#login-btn')
click()	Hace **clic** en un elemento.	locator.click()
fill(value)	Ingresa un **valor** en un campo de entrada de texto.	locator.fill('mi_usuario')
type(text, options)	Escribe texto carácter por carácter (simulando mecanografía).	locator.type('secreto')
press(key)	Simula la pulsación de una	locator.press('Enter')

	tecla (e.g., 'Enter', 'Tab').	
check() / uncheck()	Marca/desmarca una **casilla de verificación** o **radio button**.	locator.check()
selectOption(value)	Selecciona una **opción** de un menú desplegable (<select>).	locator.selectOption('valor1')
waitForEvent(eventName)	Espera a que ocurra un evento específico (e.g., nueva página, diálogo).	page.waitForEvent('popup')
goto(url)	Navega a una **URL** específica.	page.goto('https://ejemplo.com')

---

2. Funciones de Verificación y Estado (Clase Page / Locator)

Estas funciones se usan para **verificar el estado** actual de la página o de un elemento.

Función	**Descripción**	**Ejemplo**
isVisible()	Devuelve un booleano indicando si un elemento	await locator.isVisible()

	es **visible**.	
isDisabled()	Devuelve si un elemento está **deshabilitado**.	await locator.isDisabled()
textContent()	Devuelve el **contenido de texto** de un elemento.	await locator.textContent()
getAttribute(name)	Devuelve el **valor de un atributo** del elemento (e.g., 'class', 'id').	await locator.getAttribute('href')
screenshot(options)	Toma una **captura de pantalla** de la página o de un elemento.	page.screenshot({ path: 'ss.png' })

---

3. Funciones de Configuración y Contexto (Clases Browser / BrowserContext)

Estas funciones se usan para inicializar y configurar el entorno de automatización.

Función	**Descripción**	**Ejemplo**
launch(options)	**Inicia** una nueva instancia del navegador.	await playwright.chromium.launch()

newPage()	Crea un **nuevo objeto Page** (una nueva pestaña o ventana).	const page = await context.newPage()
newContext(options)	Crea un **nuevo contexto de navegador** aislado, útil para emulaciones o autenticación.	browser.newContext({ locale: 'es-ES' })
route(url, handler)	**Intercepta** peticiones de red para simular o modificar respuestas.	page.route('**/api/*', () => {...})
addInitScript(script)	Inyecta código **JavaScript** que se ejecuta al inicio de cada página.	context.addInitScript(script)
close()	**Cierra** el objeto BrowserContext o la Browser instancia.	await browser.close()

---

4. Funciones Específicas del Framework de Pruebas (@playwright/test)

Al usar el framework de pruebas de Playwright, estas funciones son esenciales.

Función	Descripción	Ejemplo
test()	Define y agrupa una **prueba individual** (el caso de prueba).	test('El login funciona', async ({ page }) => { ... });
expect()	Contiene las **aserciones** de Playwright para validar resultados.	await expect(locator).toHaveText('Bienvenido')
test.describe()	Agrupa pruebas en un **conjunto** o **suite** lógico.	test.describe('Pruebas de Carrito', () => { ... });
test.use()	Permite **configurar parámetros** específicos (como viewport, rutas base) para un *suite* de pruebas.	test.use({ viewport: { width: 600, height: 800 } })
test.beforeEach()	Ejecuta código **antes** de cada prueba dentro de un archivo/suite.	test.beforeEach(async ({ page }) => { page.goto('/') });