

Requisitos Esenciales de JavaScript

Para comenzar con Playwright de manera efectiva, deberías tener un entendimiento sólido de los siguientes conceptos clave de JavaScript (o TypeScript, que es muy recomendable para proyectos más grandes):

1. Fundamentos del Lenguaje

- **Variables y Tipos de Datos:** Saber cómo declarar variables (`let`, `const`) y manejar tipos como `string`, `number`, `boolean`, `array`, y `object`.
- **Estructuras de Control:** Dominio de las estructuras básicas como `if/else`, `for`, `while` y `switch`.
- **Funciones:** Entender cómo definir y llamar funciones, incluyendo las modernas **funciones flecha** (`=>`).

2. JavaScript Asíncrono

Este es, sin duda, el aspecto **más crucial** para Playwright, ya que todas las interacciones con el navegador son asíncronas.

- **Promesas (Promises):** Entender el concepto de Promesas (estados: *pending*, *fulfilled*, *rejected*).
- **async/await:** Debes sentirte cómodo usando las palabras clave `async` y `await`. Playwright utiliza estas estructuras constantemente para esperar a que las acciones del navegador (como hacer clic, navegar o esperar elementos) se completen. *Ejemplo: Playwright usa `await page.click('selector');`*

3. Módulos y Entorno Node.js

Playwright se instala a través de `npm` o `yarn` y se ejecuta en Node.js.

- **Gestión de Paquetes:** Familiaridad con el uso de **npm** o **yarn** para instalar dependencias.
- **Sistema de Módulos:** Saber cómo importar y exportar módulos (usando `require()` o las sintaxis `import/export` de ES Modules) para organizar tu código de prueba.

4. Estructuras de Datos Comunes

- **Arrays:** Saber manipular arrays (por ejemplo, con métodos como map, filter, forEach).
 - **Objetos:** Entender cómo acceder a propiedades de objetos y trabajar con la sintaxis de **desestructuración** (Destructuring), aunque esto último es más un plus.
-

Conceptos Adicionales Muy Útiles

Si bien los anteriores son los requisitos, estos conceptos de JavaScript mejorarán significativamente tu experiencia y la calidad de tus pruebas:

- **TypeScript (Recomendado):** Aunque Playwright soporta JS puro, la mayoría de los equipos modernos optan por **TypeScript**. Te proporciona tipado estático, lo que resulta en pruebas más robustas, menos errores y mejor autocompletado en tu IDE.
- **Selectores de CSS:** Aunque no es JavaScript puro, un buen entendimiento de los selectores de CSS es fundamental, ya que son la forma primaria de identificar elementos con el método locator() de Playwright.

En resumen, **dominar el asincronismo con async/await** y tener una **base firme en Node.js** son los pilares de JavaScript que te permitirán dominar Playwright rápidamente.