Introduction to Playwright

What is Playwright?

- **Open-source automation tool** developed by Microsoft (released in 2020).
- Used for **browser automation** and **end-to-end (E2E) testing**.
- Also supports **API testing** with a dedicated API.
- Built on **Node.js** (a JavaScript runtime that runs outside the browser).

Key Features

1. **Cross-Browser Support**
    - Works with **Chromium** (Chrome, Edge), **Firefox**, and **WebKit** (Safari).
2. **Cross-Platform**
    - Runs on **Windows, macOS, and Linux**.
3. **Multi-Language Support**
    - Write tests in **JavaScript, TypeScript, Java, Python, or C#**.
4. **Mobile Web Testing**
    - Supports **Chrome (Android) & Safari (iOS)**.
5. **API Testing**
    - Built-in support for testing backend APIs.
6. **Auto-Waiting**
    - Automatically waits for elements to be ready before interacting.
7. **Handles Complex Elements**
    - Works well with **Shadow DOM** (hard for other tools).
8. **Parallel Execution**
    - Runs tests **simultaneously** in multiple browsers for speed.
9. **Built-in Reporting**
    - Supports **HTML, JSON, JUnit** reports + third-party tools like **Allure**.

Playwright Tools

- **Inspector** – Debug tests by viewing locators and click points.
- **Code Generation (Codegen)** – Records user actions and generates test scripts.
- **Tracing (Trace Viewer)** – Captures screenshots, logs steps, and records videos for debugging.

JavaScript vs. TypeScript

JavaScript (Dynamically Typed)

- No strict type checking.
- Variables can change types.

Example:

```
let age = 30;       // Number
let name = "John";  // String
age = "thirty";     // No error (changes to string)
```

TypeScript (Statically Typed)

- A **superset of JavaScript** (adds static typing).
- Variables **must** match their declared type.

Example:

```
let age: number = 30;  // Must stay a number
let name: string = "John";
age = "thirty";        // ERROR (Type 'string' not assignable to 'number')
```


Why Use TypeScript?

- **ECMAScript (ES)** is the standard for JavaScript.
- **TypeScript adds extra features** (like types) but still compiles to standard JavaScript.
- Helps catch errors **early** in development.