Criteria	Playwright	Cypress
Language	Supports multiple languages such as JavaScript, Java, Python, and .NET C#	Supports JavaScript, TypeScript
Test Runner Frameworks Supported	Mocha, Jest, Jasmine	Mocha
Operating Systems Supported	Windows, Linux, and macOS	Windows, Linux, and macOS 10.9 and above
Open Source	Open Source and free	Open Source and Free
Architecture	Headless Browser with event-driven architecture. Playwright leverages direct WebSocket integration, enabling seamless transmission of test code in JSON format to the server through the WebSocket protocol. Upon establishing the connection or handshake, a continuous command exchange occurs between your test and the Playwright server. The client and server connections persist until either party decides to terminate them. Once the connection is closed, it is terminated from both ends. One of the key factors contributing to Playwright's exceptional speed is the uninterrupted nature of the connection, ensuring optimal performance as long as the connection remains active and uninterrupted by either party.	Executes test cases directly inside the browser. Cypress operates with a Node server, establishing constant communication, synchronization, and task execution between the entities. The interaction between the Node server and the browser transpires via WebSocket, which initiates upon the creation of the proxy. In Cypress, all commands are executed directly within the browser, eliminating the need for driver binaries and streamlining the testing process. The below image showcases the execution of tests within the browser, resulting in expedited test execution devoid of any network latency. Because of its architecture and integration with Cypress, it grants easy access to crucial components such as the DOM, local storage, network layer, and window object. This provides a notable advantage over other test automation

		frameworks, simplifying the execution of tests and enabling comprehensive analysis of the elements above.
Browsers Supported	Chromium, Firefox, and WebKit	Chrome, Firefox, and Edge
Support	Since Playwright is fairly new, the support from the community is limited as compared to Selenium	Strong community support from professionals across the world
Real Devices Support	Does not support real devices for Mobile Browser Tests but supports emulators	Supports real device cloud and remote servers
Which is faster?	Playwright's architecture allows for parallel test execution and can execute tests in multiple browsers simultaneously, leading to faster test runs, especially when conducting cross-browser testing. Additionally, Playwright's multi-language support allows you to choose a language that may be more performant for your specific use case. Playwright may have an advantage in terms of parallel execution and cross-browser testing.	Cypress, on the other hand, is known for its real-time reloading, which can speed up the development and debugging process. However, it primarily focuses on Chromium-based browsers, so if you need to test in other browsers, you may need to consider additional setup and potentially sacrifice some execution speed. While, Cypress provides immediate feedback during test development.
Which is better?	Playwright shines with its cross-browser support, multi-language compatibility, and the ability to automate modern web applications. If you need to ensure broad compatibility across browsers and platforms, Playwright's versatility may make it the better option, although it may require a steeper learning curve.Cypress excels in its ease of use, developer-friendly API, and real-time reloading, making it an excellent choice for front-end developers who prioritize simplicity and quick feedback, especially when predominantly testing in Chromium-based browsers.	Cypress excels in its ease of use, developer-friendly API, and real-time reloading, making it an excellent choice for front-end developers who prioritize simplicity and quick feedback, especially when predominantly testing in Chromium-based browsers.