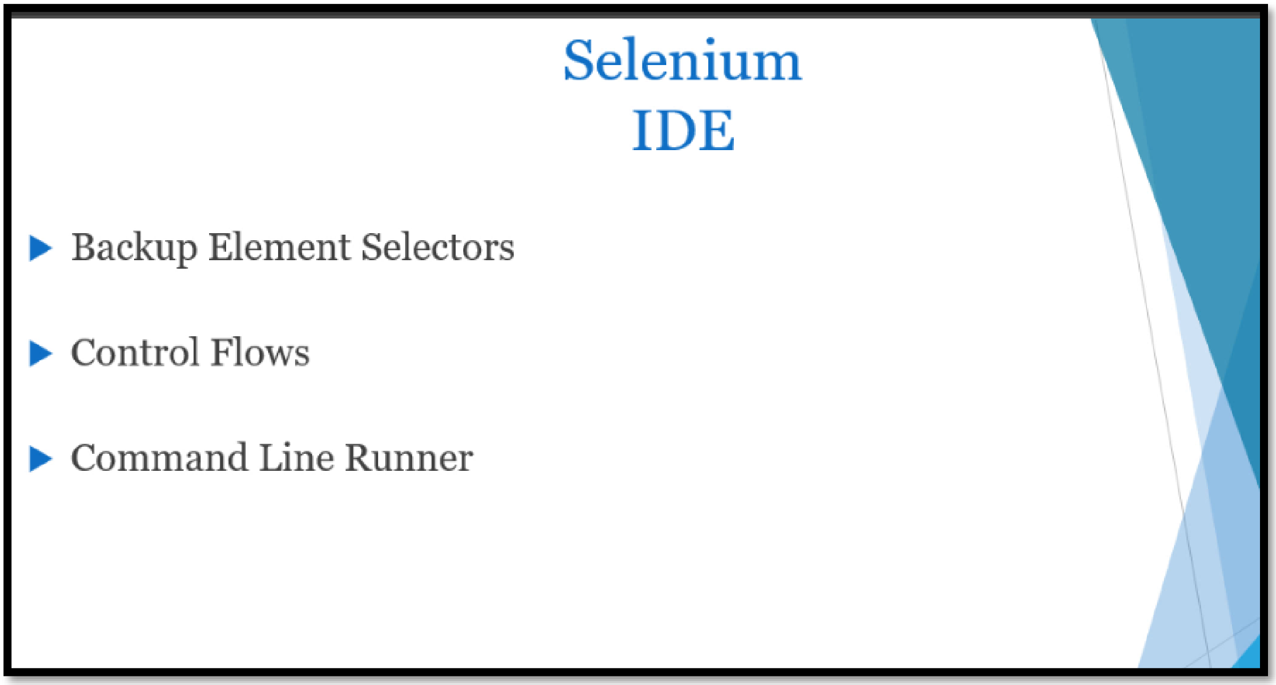
Selenium 4
IDE New Features

Video Playlist Link https://www.youtube.com/playlist?list=PLfp-cJ6BH8u_4AMzeLVizVfqn4SCywSTJ

When it comes to Selenium IDE, the purpose is to record actions we take on a website then playback those actions from the website. In the past, Selenium IDE was only available as a Firefox extension but now it's available as a Firefox extension and Chrome extension. Microsoft Edge will be available soon as an extension. There are plans for Selenium IDE to be available as an Electron app. The Electron app will allow us to use the Debugging Protocol and listen out for events from the browser. Some new features include the Backup Element Selectors, Control Flows, and the Command Line Runner also known as CLI runner.



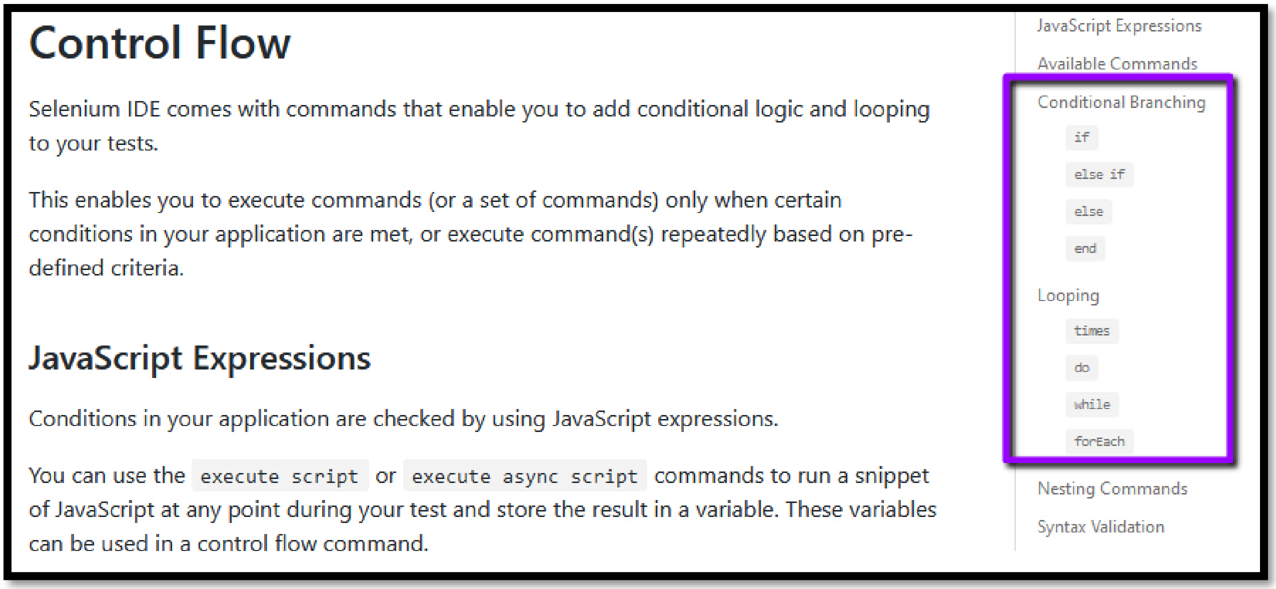
Selenium
IDE

- Backup Element Selectors
- Control Flows
- Command Line Runner

The Backup Element Selectors record more than 1 locator for each element. If we execute a test that does not locate an element then this feature will fall back to a different locator until the element is found. For example, our test runs and it cannot locate an element using a linkText locator. The Backup Element Selectors will check to see what other locators were recorded for this element then grab another locator such as id, xpath, or cssSelector.

There are 2 types of Control Flows. We have Conditional Branching and Looping. The Conditional Branching Flow decides what happens next if a certain condition is satisfied or not satisfied. Loop

Control Flows execute an instruction for a number of times. We see the commands for each flow are if, else if, else, and end for Conditional Branching while Looping has times, do, while, and forEach.



Control Flow

Selenium IDE comes with commands that enable you to add conditional logic and looping to your tests.

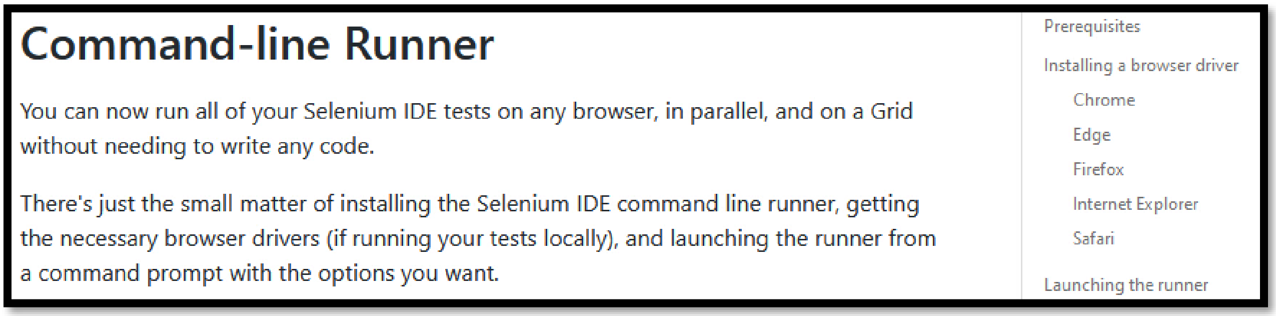
This enables you to execute commands (or a set of commands) only when certain conditions in your application are met, or execute command(s) repeatedly based on pre-defined criteria.

JavaScript Expressions

Conditions in your application are checked by using JavaScript expressions.

You can use the `execute script` or `execute async script` commands to run a snippet of JavaScript at any point during your test and store the result in a variable. These variables can be used in a control flow command.

The Command Line Runner also called Selenium side runner allows us to run our test on any browser: Chrome, Edge, Firefox, Internet Explorer, and Safari. In this 1st sentence, it says “in parallel, and on a Grid without needing to write any code”. That’s why we can imagine Selenium IDE as an on ramp to the Selenium family because little to no programming knowledge is required to start automating our test. By the way, the new name is called Selenium IDE TNG. TNG is an acronym for The Next Generation so the complete name is Selenium IDE The Next Generation.



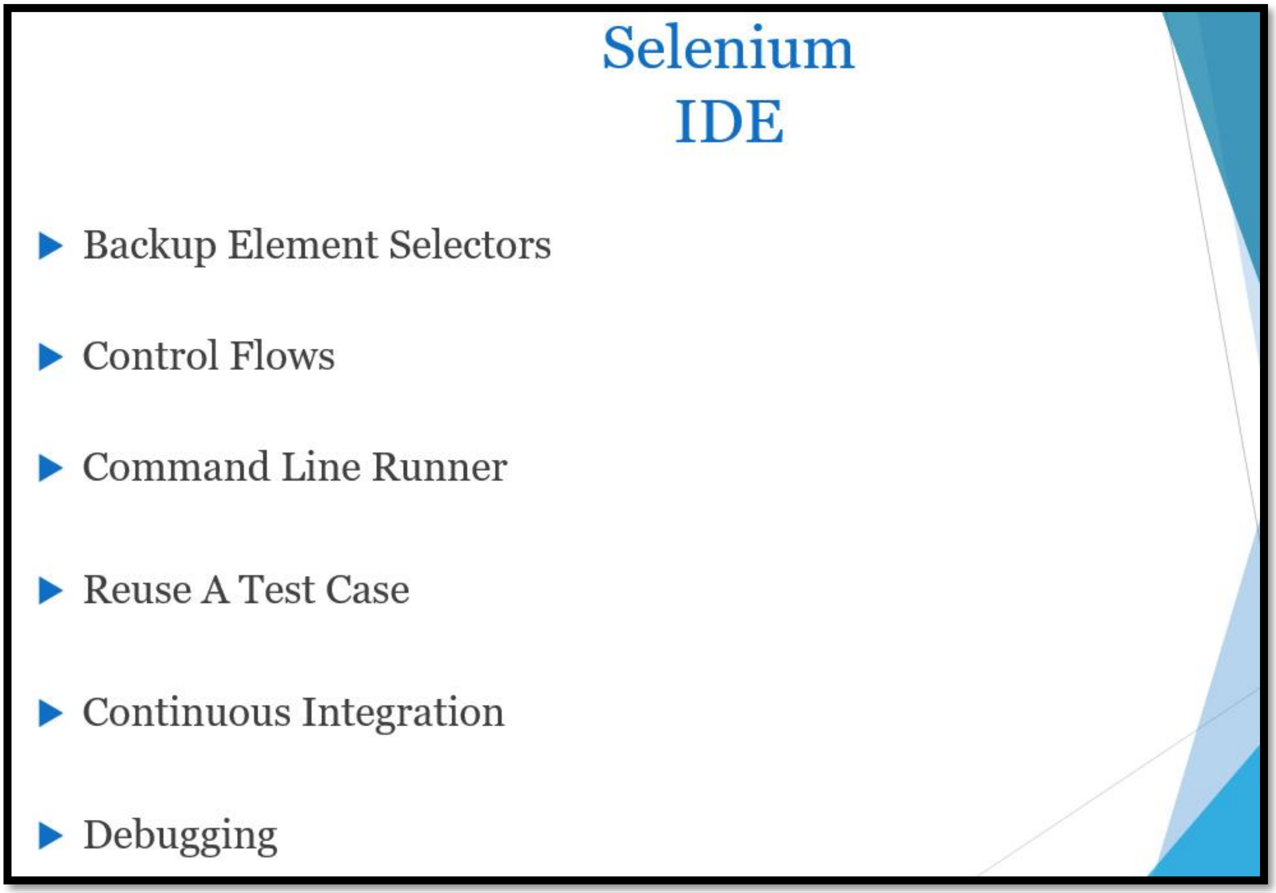
Command-line Runner

You can now run all of your Selenium IDE tests on any browser, in parallel, and on a Grid without needing to write any code.

There's just the small matter of installing the Selenium IDE command line runner, getting the necessary browser drivers (if running your tests locally), and launching the runner from a command prompt with the options you want.

Do you see Code Export under Introduction? Code Export is not a new feature but it’s a good feature that pushes code to Selenium WebDriver in your chosen language and test framework. To export, all we do is right click a test or suite, select Export then pick a supported language, test framework and click the Export button. Oh, I almost forgot, Selenium IDE has a way to create new commands as part of their plugin system. Those plugins can be delivered as an extension. A big shout out goes to Applitools because they are the ones who revived Selenium IDE and has a plugin that makes it possible to perform codeless visual testing.

We can also reuse a Test Case implementing a run command and a store command. The run command executes a test from within another test. Store is a command that allows us to save a value that we can reuse later. Last but not least is Continuous Integration and Debugging. Selenium Side Runner assist with integrating our code continuously into a shared repository. The debugging feature allows us to set breakpoints to pause our test at a certain point, play our test to a certain point, record our test from a certain point, and play our test from a certain point.



Selenium
IDE

- Backup Element Selectors
- Control Flows
- Command Line Runner
- Reuse A Test Case
- Continuous Integration
- Debugging

Next, let's talk about the new features for Selenium Grid.

Social Media Contact

✔ YouTube https://www.youtube.com/c/RexJonesII/videos

✔ Facebook http://facebook.com/JonesRexII

✔ Twitter https://twitter.com/RexJonesII

✔ GitHub https://github.com/RexJonesII/Free-Videos

✔ LinkedIn https://www.linkedin.com/in/rexjones34/