



# Introduction to Web Automation



## It's time to get hands-on with Automation!

To do so, we present you with a brief introduction to automating website tests and Selenium, the tool you will be using to do this. Let's take a look!

Sounds awesome, right? Try this for yourself now:

### Practice

In this practice, we are going to create our first automation framework using Selenium. To achieve this, start by creating a maven java project including the latest version of Selenium Dependency.

Follow this tutorial to install Selenium:  
[https://www.selenium.dev/documentation/webdriver/getting\\_started/install\\_library/](https://www.selenium.dev/documentation/webdriver/getting_started/install_library/).

Make sure to also have Google Chrome installed and to download the respective ChromeDriver for your Chrome version, you can use the following link:  
[https://www.selenium.dev/documentation/webdriver/getting\\_started/install\\_drivers/](https://www.selenium.dev/documentation/webdriver/getting_started/install_drivers/)

Now we are almost ready to start!

Check the following link to get familiarized with basic Selenium actions. Create in your project an Example class and implement the code displayed in this article.  
[https://www.selenium.dev/documentation/webdriver/getting\\_started/first\\_script/](https://www.selenium.dev/documentation/webdriver/getting_started/first_script/)

After the first example is correctly implemented, we are going to create our own Selenium scripts.

To do this, create a new class with a public static void main method. Use the Selenium methods in the previous example to implement the following actions on the main method:

Navigate to the webpage (<https://wikipedia.org>)

1. Navigate to the webpage (<https://wikipedia.org>)
2. Write in the search box the text "Microsoft"
3. Click on the search button aside the search box
4. Wait for the new webpage to load
5. Get the title of the new webpage and check if it matches with "Microsoft", print a message in the console indicating whether it matches or not.
6. Get all the anchor links on the web page (hint: find by HTML tag "a"), and for each one of them, print on the console its respective text.

**Note:** For steps b, c, d, e, you can find the elements using the selenium Class By and its method CSS. To get the CSS locator of an element on a web page, right click on it and click inspect, then, the HTML code of the web page will be displayed. Right-click on the highlighted element and select copy > selector (only with chrome)

Make sure to execute the main class and make the required adjustments for the script to properly work.

