**The Selenium Browser Automation Project**

Selenium is an umbrella project for a range of tools and libraries that enable and support the automation of web browsers.

It provides extensions to emulate user interaction with browsers, a distribution server for scaling browser allocation, and the infrastructure for implementations of the [W3C WebDriver specification](https://www.w3.org/TR/webdriver/) that lets you write interchangeable code for all major web browsers.

This project is made possible by volunteer contributors who have put in thousands of hours of their own time, and made the source code [freely available](https://selenium.dev/documentation/en/front_matter/copyright_and_attributions/#license) for anyone to use, enjoy, and improve.

Selenium brings together browser vendors, engineers, and enthusiasts to further an open discussion around automation of the web platform. The project organises [an annual conference](https://seleniumconf.com/) to teach and nurture the community.

At the core of Selenium is [*WebDriver*](https://selenium.dev/documentation/en/webdriver/), an interface to write instruction sets that can be run interchangeably in many browsers. Here is one of the simplest instructions you can make:

Java Python C# Ruby JavaScript Kotlin

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.WebDriverWait;

import static org.openqa.selenium.support.ui.ExpectedConditions.presenceOfElementLocated;

import java.time.Duration;

public class HelloSelenium {

public static void main(String[] args) {

WebDriver driver = new FirefoxDriver();

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));

try {

driver.get("https://google.com/ncr");

driver.findElement(By.name("q")).sendKeys("cheese" + Keys.ENTER);

WebElement firstResult = wait.until(presenceOfElementLocated(By.cssSelector("h3>div")));

System.out.println(firstResult.getAttribute("textContent"));

} finally {

driver.quit();

}

}

}

See the [*Quick Tour*](https://selenium.dev/documentation/en/getting_started/quick/) for a full explanation of what goes on behind the scenes when you run this code. You should continue on to the [narrative documentation](https://selenium.dev/documentation/en/introduction/) to understand how you can [install](https://selenium.dev/documentation/en/selenium_installation/) and successfully use Selenium as a test automation tool, and scaling simple tests like this to run in large, distributed environments on multiple browsers, on several different operating systems.

Getting started

If you are new to Selenium, we have a few resources that can help you get up to speed right away.

* [Quick tour](https://selenium.dev/documentation/en/getting_started/quick/)
  + [WebDriver](https://selenium.dev/documentation/en/getting_started/quick/#webdriver)
  + [IDE](https://selenium.dev/documentation/en/getting_started/quick/#ide)
  + [Grid](https://selenium.dev/documentation/en/getting_started/quick/#grid)