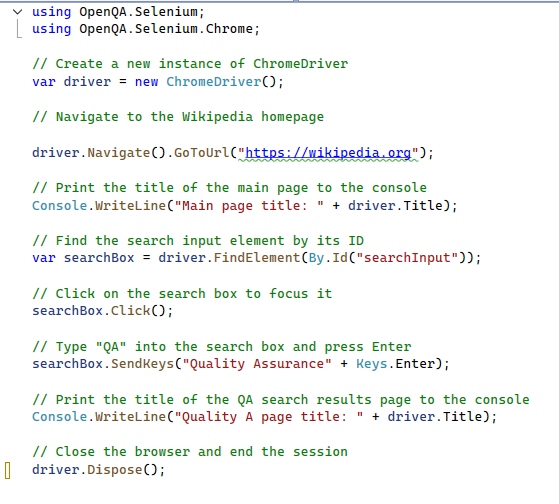
# **Lab: Selenium WebDriver**

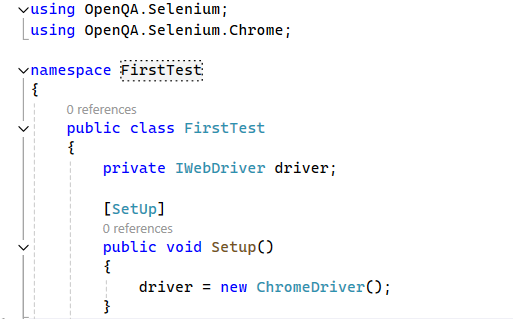
## Search for "Quality Assurance" in Wikipedia

* Select Create a new Console App (.NET Core) project in Visual Studio.
* Install Necessary NuGet Packages:
  + Selenium WebDriver
  + ChromeDriver
* Import necessary namespaces.
* Inside the Main method, create a new instance of ChromeDriver.
* Navigate to the Wikipedia homepage.
* Print the title of the main page to the console.
* Find the search input element by its ID.
* Click on the search box to focus it.
* Type "Quality Assurance" into the search box and press Enter.
* Print the title of the QA search results page to the console.
* Close the browser and end the session.

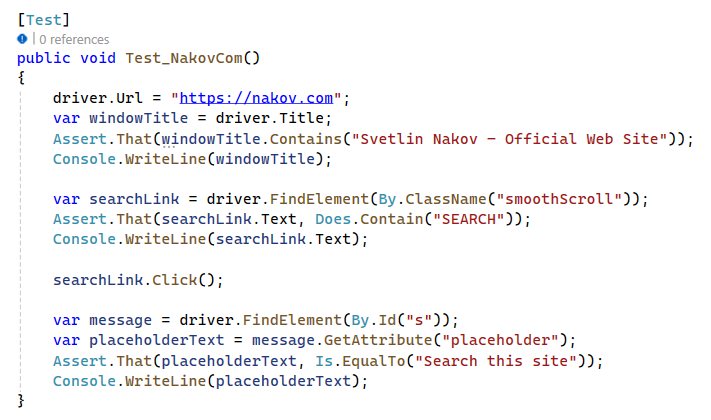
a

## First Test

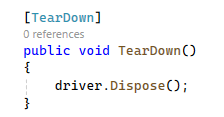
* Create a new NUnit Test Project (.NET Core) project.
* Install Necessary NuGet Packages.
* Import necessary namespaces.
* Define the test class.
* Inside the class, declare a private IWebDriver variable named driver.
* Create a Setup method to initialize the ChromeDriver before each test.



* Write the test method.
* In the method, navigate to https://nakov.com.
* Retrieve the page title and verify it contains "Svetlin Nakov – Official Web Site".
* Find the search link by its class name and verify its text contains "SEARCH".
* Click on the search link.
* Find the search input element by its ID.
* Retrieve and verify the placeholder attribute of the search input is "Search this site".

a

* Define the teardown method to close the browser after each test.

 a

## Practice Locators

You will receive two HTML files: **Locators.html** and **Thankyou.html**. These files will be in a folder called **SimpleForm**. Ensure both **Locators.html** and **Thankyou.html** are in the **same directory on your local machine**.

For example, place them in **C:/Users/YourUsername/Desktop/SimpleForm/**.

* Create a New Project:
* Name your project (e.g., LocatorsPractice) and click Create.
* Install Necessary NuGet Packages:
* Install Selenium WebDriver:
* Search for Selenium.WebDriver.
* Install Selenium WebDriver for Chrome:
* In your test file, add the necessary using statements at the top.
* Initialize the WebDriver and navigate to the Locators.html file.

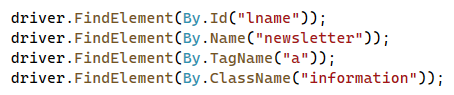


* Write **OneTimeSetup** and **OneTimeTearDown** Method

### **Basic Locators**

Practice locating elements by ID, Name, Tag Name, and Class Name, for example:

* Find the LastName by ID
* Find the Newsletter by Name
* Find the Official Softuni Page by TagName
* Find the information fields by ClassName



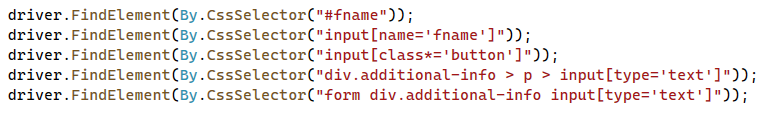
### **Text Link Locators**

Practice locating elements by Link Text and Partial Link Text, for example:



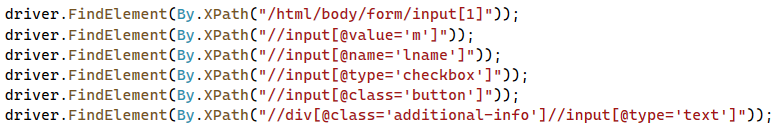
### **CSS selectors**

Practice locating elements using various CSS selectors, for example:



### **XPath Locators**

Practice locating elements using various XPath expressions, for example:

a

## Practice Locators Extended

### **Basic Locators**

**Extend the previous task. Add assertions to verify the located elements.**

* Locate the "Last name" input field and verify its value
* Locate the "Newsletter" checkbox and verify it is not selected
* Locate the anchor tag and verify its text
* Locate the element with class name "information" and verify its background color

### **Text Link Locators**

* Locate the link by its full text and verify its href attribute
* Locate the link by partial text and verify its displayed text

### **CSS selectors**

* Locate the "First name" input field by ID and verify its value
* Locate the "First name" input field by name attribute and verify its value
* Locate the submit button by class name and verify its value attribute
* Locate the "Phone Number" input field by CSS selector and verify it is displayed
* Locate the "Phone Number" input field using a more specific CSS selector and verify it is displayed

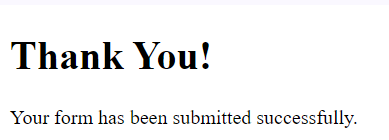
### **XPath Locators**

* Locate the male radio button using absolute XPath and verify its value attribute
* Locate the male radio button using relative XPath and verify its value attribute
* Locate the last name input field using relative XPath and verify its value
* Locate the newsletter checkbox using relative XPath and verify its type attribute
* Locate the submit button using relative XPath and verify its value attribute
* Locate the phone number input field within additional info using relative XPath and verify it is displayed

## Test Form Submission

Write a Test that does the following:

* Asserts the "Contact Form" title.
* Selects the male radio button and asserts its selection.
* Enters "Butch" as the first name and asserts the entered value.
* Enters "Coolidge" as the last name and asserts the entered value.
* Asserts the presence of the "Additional Information" section.
* Enters "0888999777" as the phone number and asserts the entered value.
* Selects the newsletter checkbox and asserts its selection.
* Clicks the submit button.
* Asserts the "Thank You!" message on the next page.

a

## Headless Mode

Run some of your Tests in a headless browser. Check if there is difference in times.

