

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
Selenium
<div><div></div>What is Selenium?</div>
<div><div></div>Selenium IDE</div>
<div><div></div>Webdrivers</div>
<div><div></div>Opening A Web Browser With Selenium</div>
<div><div></div>Browser manipulation</div>
<div><div></div>Interacting With Elements In Selenium</div>
<div><div></div>The POM Design Pattern</div>
<div><div></div>Actions</div>
<div><div></div>Waits</div>
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security

Interacting With Elements In Selenium

Contents

- [Overview](#)
 - [Finding Elements](#)
 - [Interactions](#)
- [Tutorial](#)
- [Exercises](#)

Overview

Each tag on a Web page is an element which can be found and interacted with. The **By** class is used to select an element, and the **WebElement** interface has a set of actions which can be performed on every element.

Finding Elements

Elements can be found by running the `webdriver.findElement(By)` method. The **By** class has a selection of methods which can be used to select different types of tags:

By method	Example code	HTML element it could select
By.id	<code>driver.findElement(By.id("profile-picture"))</code>	<code></code>
By.cssSelector	<code>driver.findElement(By.cssSelector("body > div:nth-child(2)"))</code>	<code><body></code> <ul style="list-style-type: none">• <code><div></div></code>• <code><div></div></code> <code><--</code>
By.xpath	<code>driver.findElement(By.xpath("/html/body/div[2]"))</code>	<code><body></code> <ul style="list-style-type: none">• <code><div></div></code>• <code><div></div></code> <code><--</code>
By.name	<code>driver.findElement(By.name("searchbar"))</code>	<code><input name="searchbar"/</code>
By.className	<code>driver.findElement(By.className("btn"))</code>	<code><input class="btn btn-primary"></code>
By.tagName	<code>driver.findElement(By.tagName("header"))</code>	<code><header> Welcome to QA </header></code>
By.linkText	<code>driver.findElement(By.linkText("About"))</code>	<code>About</code>
By.partialLinkText	<code>driver.findElement(By.partialLinkText("out"))</code>	<code>About</code>

id is the best selector to use as it will always be unique.

Selectors such as **className** are likely to find more than one element, in which case `driver.findElements(By.className("btn"))` can be used to return an array of found elements instead of just one.

Interactions

Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

Interaction	Example code	What interaction does
element.findElement(By)	<code>element.findElements(By.tagName("li"));</code>	Finds sub element in selected element. For instance if the selected element was a list, might want to find all list items in that list
element.click()	<code>element.click();</code>	Clicks on the element
element.getText()	<code>String innerText = element.getText();</code>	Grabs the text from within the element and sub-elements
element.sendKeys(String)	<code>element.sendKeys("username") or element.sendKeys(Keys.chord(Keys.SHIFT, "hello "), "world");</code>	Sends a string, or a sequence of strings to the selected element. Keys.chord() can be used to simulate pressing many keys at once
element.submit()	<code>element.submit()</code>	If the element is in a form, it will submit the form
element.clear()	<code>element.clear()</code>	If the element is an input it will clear all text in the textbox

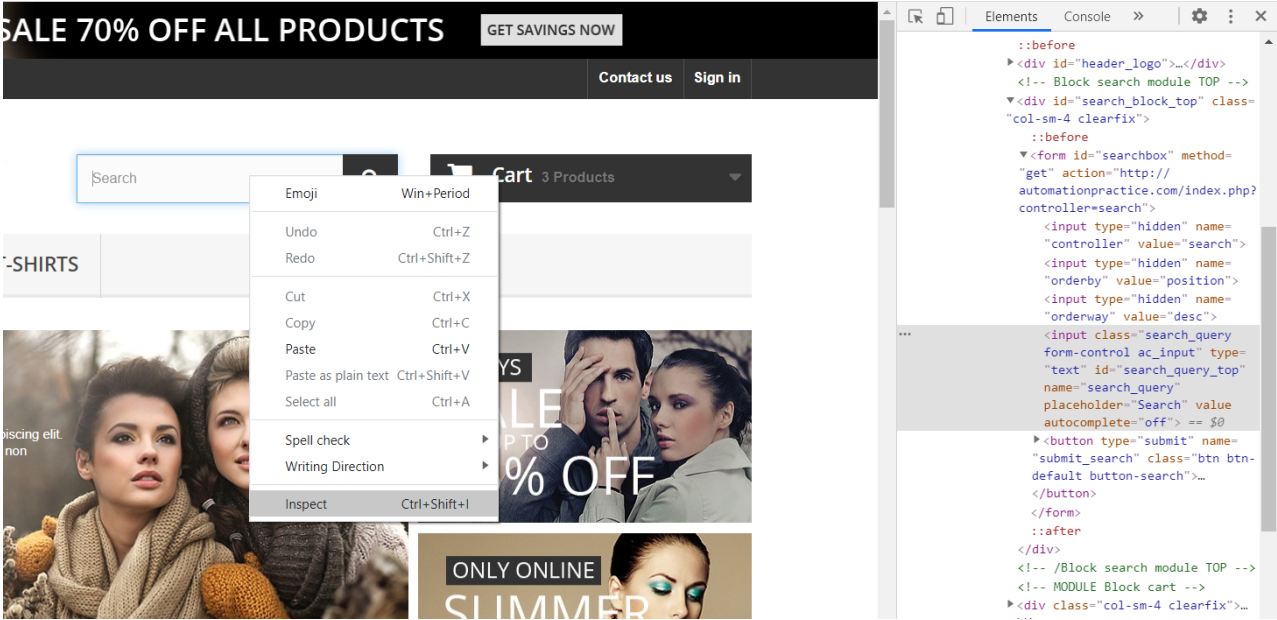
Interaction	Example code	What interaction does
element.getAttribute(String)	<code>element.getAttribute("src")</code>	Returns the value of a selected attribute of the element. Useful when automatically testing whether images are displaying by grabbing the src location, and then making a direct request to that location and then checking for a 200 response.
element.getLocation()	<code>element.getLocation()</code> or <code>element.getLocation().getX()</code>	Returns the x and y position of the top-left corner of the element. Useful for checking responsive design
element.getSize()	<code>element.getSize()</code> or <code>element.getSize().getHeight()</code>	Returns the height and width of the element. Useful for checking responsive design

Tutorial

For this tutorial we will be running a search query in an input field. We will then check that the returned results are relevant.

First manually navigate to the [automation practice Web site](#) in your browser.

Then right click and select "Inspect" on the element you want to select with Selenium:



This will select the element in the Web page DOM. The element can then be inspected for appropriate selectors. This element in particular has an `id`, which is the preferred way of selecting code (as we won't get any conflicts with other elements with the same name).

In this case, the `id` for the search function is `"search_-_query_-_top"`.

In our test, we can use the `sendKeys()` function to pass a search term (e.g. `"Printed"`) to the search bar before submitting.

We can retrieve a list of the products which appear on the resulting search results page by using `driver.findElements(By)` and using the class selector (in this case it's `"ajax_block_product"`).

We then check if the name inside each product in the list matches our search term:

► AutomationPracticeIndexPageTest

Exercises

Automate the following process on the same website:

1. Go to the [automation practice Web site](#)
2. Click T-shirts
3. Click the first T-shirt item
4. On the new page save the name of the product into a variable and click "Add to Cart"
5. On the modal that appears select "continue to checkout"
6. Check that the item clicked matches that which is in the basket