# Selenium Lab - WebDriver

## Introduction

Selenium is an extremely popular testing tool for web UI testing. Selenium is free & open source, making it a great option for most developers. There are three components offered in the Selenium Suite – WebDriver, IDE, and Grid. This tutorial will focus on WebDriver.

## Getting Started

1. You need to install the Selenium bindings for your desired language. Below is how you would use Pip if using Python. Open command prompt on your machine and type the following:
   1. pip install selenium
2. Open your IDE of choice. Import the required packages and start the session with WebDriver.

A screenshot of a computer

Description automatically generated

1. Navigate to a webpage

A black background with white text

Description automatically generated

1. Here, we’re going to test our code to make sure we’ve properly installed Selenium and are able to proceed.



Congrats! We’re now ready to move on to testing.

## Validation

1. First, let’s modularize our testing script using functions.



1. Here, we’re going to make sure that the page title is what we expect. What you’ll need is the expected page title as your input.



1. Next, we’ll compare the input to the value returned by the WebDriver using an if loop.

A black background with white text

Description automatically generated

## Adding Items

Our app allows users to add produce to a cart, much like any eCommerce application. Here, we’re going to test that.

1. Let’s create a function for this test and isolate the dropdown.



1. Next, we’re going to use the Select class to interact with the dropdown. Here, we’re selecting a visible option, you could instead use the select\_by\_value() if you wanted to input your own value.

A black background with white text

Description automatically generated

1. Now, we’ll confirm the value.

A black background with white text

Description automatically generated

1. Let’s add that to the cart.

## Input Testing

Perhaps the most useful feature of Selenium is the ability to automate tests. Here, we will teach you how to pass inputs to your web application to test. In this function, we will be testing the discount code functionality.

1. First, let’s create a function for this and gather your input. Let’s nest this in a loop for repeatability.

A black background with white text

Description automatically generated

1. Next, we will instruct the webdriver to find the inputs on the web application.



1. Before we can enter an input, we must first clear the text box.



1. Now, we can send inputs to the text box.



1. Let’s close the loop, allowing the user to test different inputs if they so choose.

A black screen with white text

Description automatically generated

## Removing Items

Our application allows for users to remove items from their cart, much in the same way that you can add them. Let’s review how to automate that process for testing.

1. First, let’s create a new function for this test.



1. Next, let’s find the dropdown responsible for removing items and select it.

A black screen with white text

Description automatically generated

1. Now, let’s select the first item, ‘Banana’.



1. Now, we’ll confirm the value was selected.



1. Now we can remove it from the cart.



## Finishing up

Alright, now that we have some tests we can run, let’s invoke all our functions and close out the webdriver.

1. Invoke all functions

A black background with white text

Description automatically generated

1. Close out the webdriver



1. Let’s put it all together

A screenshot of a computer program

Description automatically generated