Selenium Setup for Web Automation

To scrape data from a dynamically rendered webpage in Python, we need to install the selenium package (which is not included in Anaconda's default package collection). It provides an interface that allows us to write Python code to drive a browser to emulate human users' actions.

1. Install the selenium package

For most Python packages, we can use the graphical interface Anaconda Navigator to install them with just a few clicks.

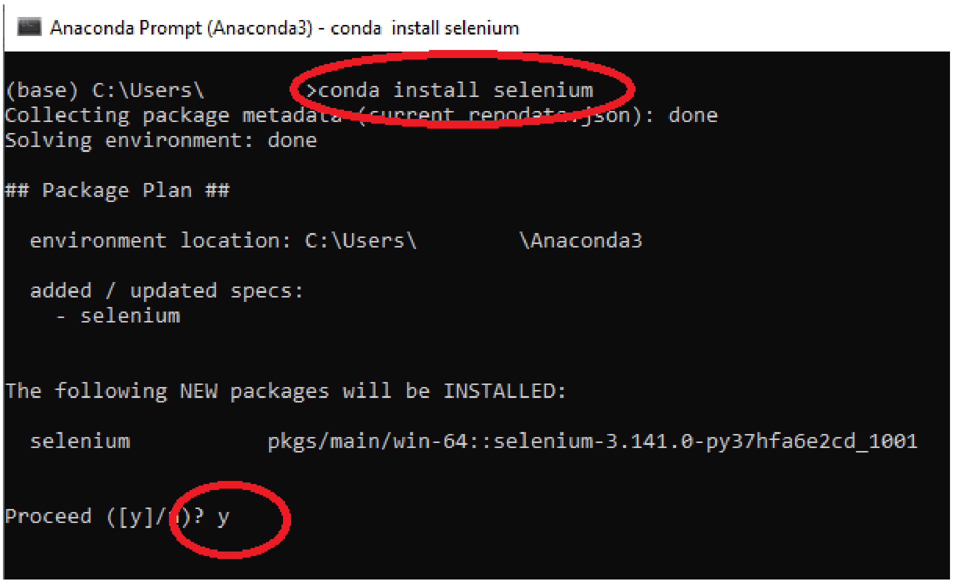
But for the selenium package, we have to use the conda package manager to install it from within a command prompt.

1) On Windows 10, go to **Start Menu**, right click **Anaconda Prompt (Anaconda3)** under **Anaconda3 (64-bit)**, choose **More**, and click **Run as administrator**.

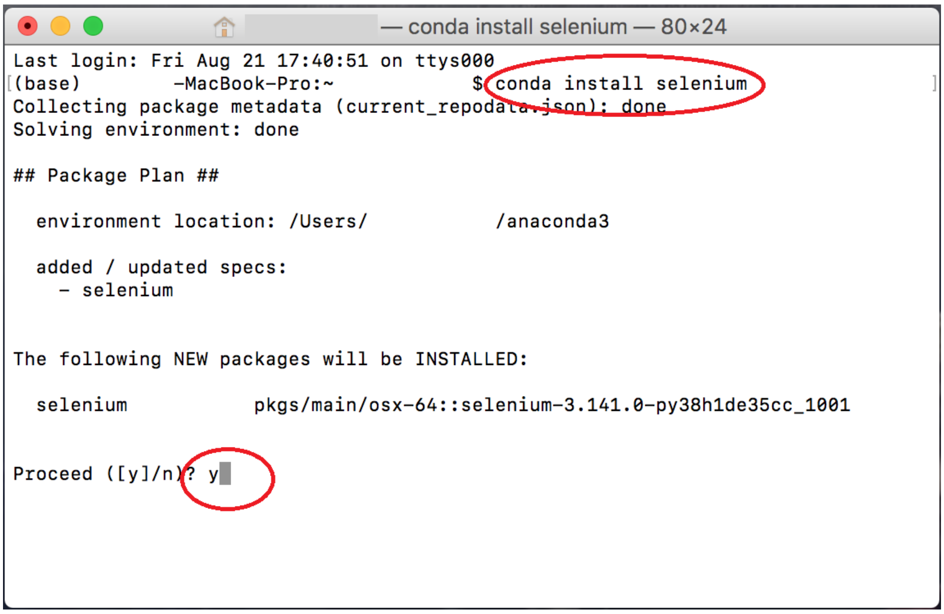
2) On macOS, launch **Spotlight Search** and open **Terminal**.

Type **conda install selenium** and press **Enter** in **Anaconda prompt/Terminal**.

Type **y** and press **Enter** to proceed. Wait for a while, and you will see the success message, indicating that the latest version of the selenium package has been installed on the computer.



```
(base) C:\Users\        >conda install selenium
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\        \Anaconda3

  added / updated specs:
    - selenium


The following NEW packages will be INSTALLED:

  selenium           pkgs/main/win-64::selenium-3.141.0-py37hfa6e2cd_1001


Proceed ([y]/n)? y
```



```
Last login: Fri Aug 21 17:40:51 on ttys000
(base)        -MacBook-Pro:~            $ conda install selenium
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /Users/           /anaconda3

  added / updated specs:
    - selenium


The following NEW packages will be INSTALLED:

  selenium           pkgs/main/osx-64::selenium-3.141.0-py38h1de35cc_1001


Proceed ([y]/n)? y
```

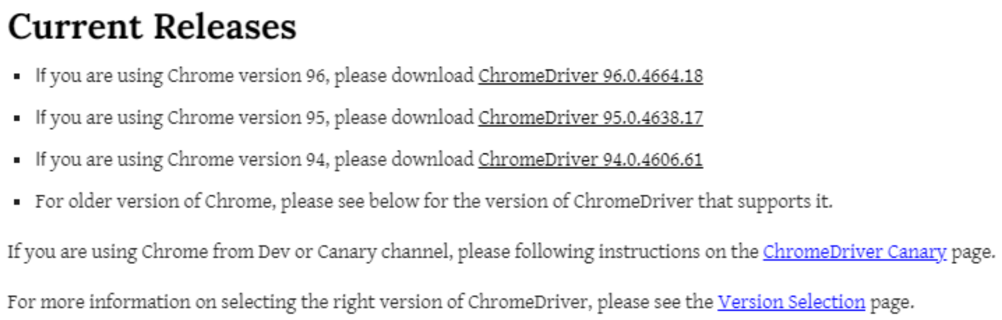
2. Install browser-specific WebDriver binaries

Selenium setup is different from the setup of other Python tools. In addition to the selenium package we just installed, we need to further install a browser-specific WebDriver binary file, an executable that will be controlled by our Python code to drive the browser we will use. So, what we are going to do next is to download the WebDriver binary file that matches the browser we choose and place it in a folder that is on your system's path.

Through WebDriver, selenium supports all major browsers on the market such as Chrome, Firefox, Internet Explorer, Opera, and Safari.

Since we are using Google Chrome for illustration in this tutorial, we will download and set up ChromeDriver.

1) Go to https://sites.google.com/chromium.org/driver/downloads



Current Releases

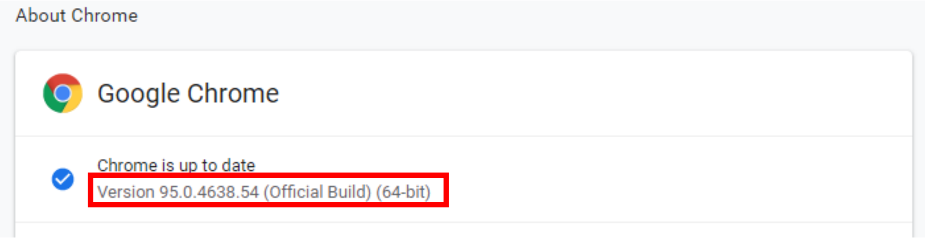
- If you are using Chrome version 96, please download ChromeDriver 96.0.4664.18
- If you are using Chrome version 95, please download ChromeDriver 95.0.4638.17
- If you are using Chrome version 94, please download ChromeDriver 94.0.4606.61
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

If you are using Chrome from Dev or Canary channel, please following instructions on the ChromeDriver Canary page.

For more information on selecting the right version of ChromeDriver, please see the Version Selection page.

What you can see from the screen capture above is that there are multiple versions of ChromeDriver. Which version of ChromeDriver to select depends on the version of Chrome you are using it with.

Let’s first check your version of Chrome. On your computer, open Chrome. At the top right, look at More ⋮ . Click Help > About Google Chrome. Then you can find the version number of Chrome you are using. In this example, it is **95**.



About Chrome

Google Chrome

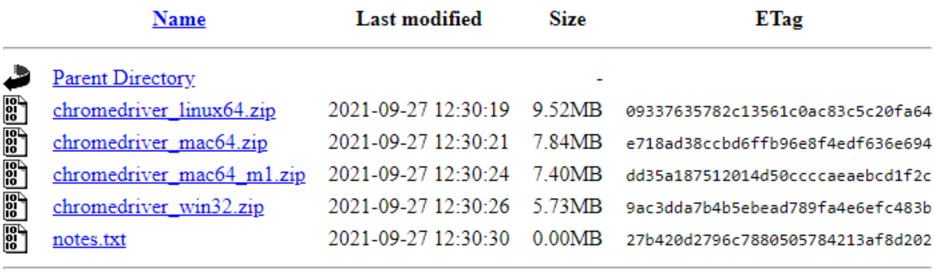
Chrome is up to date
Version 95.0.4638.54 (Official Build) (64-bit)

2) ChromeDriver uses the same version number scheme as Chrome. In this example, we should select ChromeDriver 95.0.4638.17, which supports all Chrome versions that start with 95.0.4638.

   For older version of Chrome, you can find the matching version of ChromeDriver below on the same webpage.

   Click on the ChromeDriver release that matches your Chrome version.

3) Download chromedriver_win32.zip for Windows and chromedriver_mac64.zip for macOS. Extract the binary file in it. The extracted file is named chromedriver.exe on Windows or chromedriver on Mac.



Name	Last modified	Size	ETag
Parent Directory		-	
chromedriver_linux64.zip	2021-09-27 12:30:19	9.52MB	09337635782c13561c0ac83c5c20fa64
chromedriver_mac64.zip	2021-09-27 12:30:21	7.84MB	e718ad38ccbd6ffb96e8f4edf636e694
chromedriver_mac64_m1.zip	2021-09-27 12:30:24	7.40MB	dd35a187512014d50ccccaeaebcd1f2c
chromedriver_win32.zip	2021-09-27 12:30:26	5.73MB	9ac3dda7b4b5ebead789fa4e6efc483b
notes.txt	2021-09-27 12:30:30	0.00MB	27b420d2796c7880505784213af8d202

4) To confirm that selenium has been successfully set up on your computer, you can open a Jupyter notebook file and run the following code:

```
from selenium.webdriver import Chrome
driver = Chrome(path_to_the_extracted_binary_file)
```

   where `path_to_the_extracted_binary_file` should be a string literal that specifies the full path of the extracted binary file.

   If you don’t know how to get the full path of a file on Windows 10 or macOS, refer to either of the two links that is applicable:

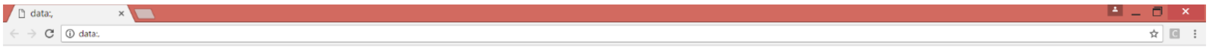
   https://www.howtogeek.com/670447/how-to-copy-the-full-path-of-a-file-on-windows-10/
   https://www.howtogeek.com/721126/3-ways-to-see-the-current-folder-path-on-mac/

If you put the extracted file in the root folder of the C drive of a Windows computer, the string you get should be `"C:\chromedriver.exe"`.

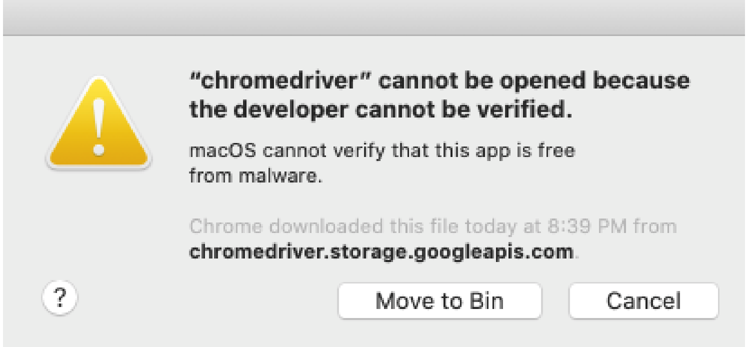
To make it a string Python can recognize, change all backslashes in it to slashes. Then the code with a valid path string should look like the following:

```
driver = Chrome("C:/chromedriver.exe")
```

Your setup is successful if you observe a new Chrome browser window is opened.



On macOS, if you observe a popup window like the following, press **Cancel** and follow this link provided by Apple to "verify" the file: right-click chromedriver, choose **Open** from the shortcut menu, and then click **Open** on the popup window.



"chromedriver" cannot be opened because the developer cannot be verified.

macOS cannot verify that this app is free from malware.

Chrome downloaded this file today at 8:39 PM from **chromedriver.storage.googleapis.com**.