**Python 3 projects**

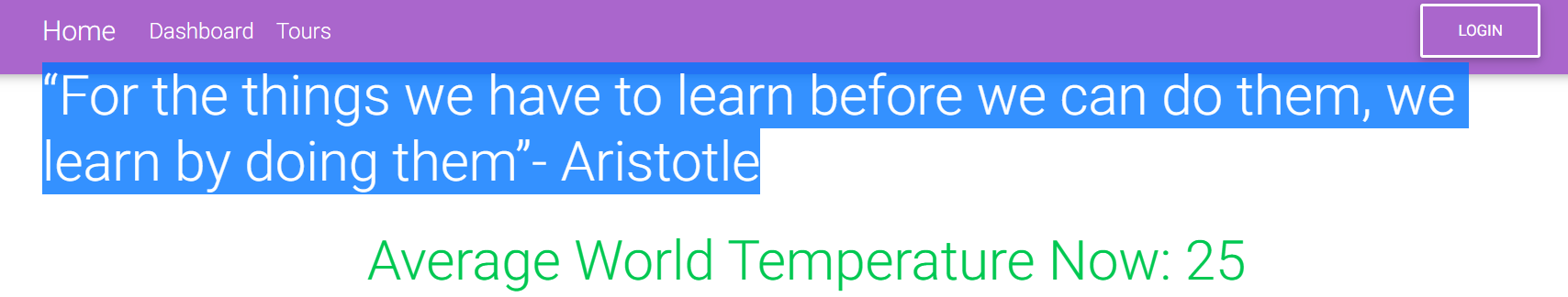
Contents

[ **Browser Automation and Web Scraping** 1](#_Toc137147751)

[ **References** 1](#_Toc137147752)

* **Browser Automation and Web Scraping**
* **Scraping Simple Text with Selenium**

Scrapping the selected text, means to get the text and load it in our Python program.



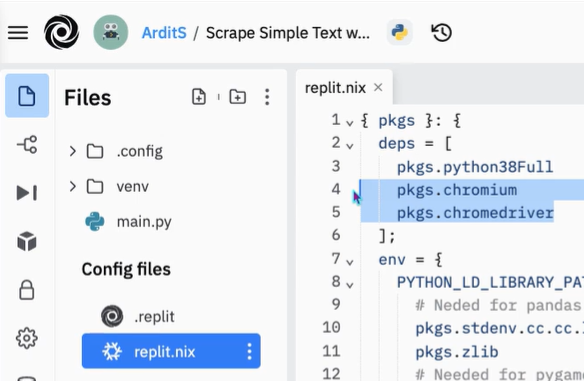
Using Online Editor Replit.com:

<https://replit.com/@ArditS/Scrape-Simple-Text-with-Selenium-done>

* Install packages:

Selenium

* Add this config in replit.nix to work with Chrome:



* To copy the tag of the web element, go to Developer Tools/Inspect/Copy/Copy XPath

A screenshot of a computer

Description automatically generated with low confidence

* **Scraping from Your Local IDE**

The same example **Scraping Simple Text with Selenium** but using **PyCharm IDE** local.

* Install **ChromeDriver:**

Get version Chrome:

<https://www.whatismybrowser.com/>

Download:

<https://chromedriver.chromium.org/downloads>

chromedriver\_win32.zip

* Run:

Photo1

* **Scraping Dynamic Value**

Scrapping the dynamic value (26):

A picture containing text, font, screenshot, design

Description automatically generated

Using Online Editor Replit.com:

<https://replit.com/@ArditS/Scrape-Dynamic-Value-done>

* To copy the tag of the web element, go to Developer Tools/Inspect/Copy/Copy XPath or Copy Full Path
* time — Time access and conversions

<https://docs.python.org/3/library/time.html>

* The Selenium Browser Automation Project

<https://www.selenium.dev/documentation/>

* **Automate Login Process**

The process is, go to this web <http://automated.pythonanywhere.com/login/>

A screenshot of a computer

Description automatically generated with medium confidence

Login:

Username = automated

Password = automated

Go to Home option from Menu

Using Online Editor Replit.com:

<https://replit.com/@ArditS/Log-in-and-Click-done>

* To find the element instead of using Copy XPath, use:

driver.find\_element(by="id", value="id\_username")

* **Log in, Click, and Scrape**

The same example **Automate Login** Process but add scrapping the dynamic value (26):

A picture containing text, font, screenshot, design

Description automatically generated

* **Log in, Click, Scrape, and Save in Text File**

The name of the file has this format:

A picture containing text, font, screenshot, design

Description automatically generated

And in that file save the dynamic value (26.0) from example **Log in, Click, and Scrape**

Using Online Editor Replit.com:

<https://replit.com/@ArditS/Scrape-Dynamic-Values-and-Save-to-File-done>

* datetime — Basic date and time types

<https://docs.python.org/3/library/datetime.html>

* **Log in and Click on Online Retailer**

The process is go to web page titan22.com, login, click in Sign In

Then click in Contact Us

Using Online Editor Replit.com:

<https://replit.com/@ArditS/Log-in-and-Click-on-Online-Retailer-done>

* **Download Stock Data for Any Company for Any Date**

Download historical stock data via Python

The process is, enter inputs from console, go to <https://finance.yahoo.com/>

Enter AAPL, click Search, click Historical Data

Click Time Period, click Apply, click Download and get CSV file.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Using Online Editor Replit.com:

<https://replit.com/@ArditS/Download-Stock-Data-done#data.csv>

* Requests Documentation

<https://docs.python-requests.org/en/latest/>

* **Scrape Real-Time Currency Rate with Beautiful Soup**

Scrape the real-time currency rates between any two currencies from x-rates.com.

<https://www.x-rates.com/calculator/?from=EUR&to=USD&amount=1>

A screenshot of a calculator

Description automatically generated with medium confidence

* Install libraries:

Beautiful Soup: pip install beautifulsoap4

Request: pip install requests

* **References**
* **Packages**
* **Selenium:** Is a Python library, used to automate browser actions.

<https://www.selenium.dev/documentation/webdriver/>

* **Python Libraries**

**Beautiful Soup**: only for **web scrapping**

**Selenium** is more target automation, automate actions on a browser (text boxes, clicking on a button, logging in and out), hybrid between **browser automation** and **web scraping**.

* **Replit Tips**
* **Deleting All Text Files: Log in and Click on Online Retailer**

Delete all text files from your Repl directory, go to the Shell tab and run:

rm \*.txt

* Creating a simple web app with Python and Flask

<https://www.youtube.com/watch?v=cjYxDFsSZio>

* Build a Python Facial Recognition App with Tensorflow and Kivy

<https://www.youtube.com/watch?v=LKispFFQ5GU>