**Xolani Keith Mpala (444463)**

**Ivan Alexander Moz Prez (440130)**

# Obtaining data for the top 100 cryptocurrencies based on their market capitalization through web scraping techniques.

A cryptocurrency is a form of digital currency that operates as a medium of exchange within a computer network. Unlike traditional currencies, cryptocurrencies do not rely on a central authority like a government or bank to regulate or maintain them. They are based on blockchain technology, which is a secure and transparent method of recording online transactions through encryption and authentication.

Cryptocurrencies derive their name from the use of encryption to verify and safeguard transactions. Currently, there are over a thousand distinct cryptocurrencies worldwide. In recent years, the prices of cryptocurrencies have experienced fluctuations, rising and falling. It's important to note that crypto marketplaces do not guarantee the optimal price for investors when making purchases or trades. Consequently, some investors engage in arbitrage, taking advantage of price differences across various markets. The first decentralized cryptocurrency, Bitcoin, was introduced in 2009 as open-source software. Since then, the growing interest in using cryptocurrencies has led to the creation of numerous other digital currencies.

The primary focus of our project revolves around conducting web scraping activities on the website (https://www.coingecko.com) using suitable tools and frameworks.

The tools we will employ include:

- Beautiful Soup
- Scrapy
- Selenium

**About coingecko.com:**

CoinGecko is a platform that focuses on providing digital currency price and information data to enable users to quantitatively assess and rank their coins. Established in April 2014 by TM Lee and Bobby Ong, it is headquartered in Singapore and aims to democratize access to cryptocurrency data, equipping users with actionable insights.

As the world's largest independent cryptocurrency data aggregator, CoinGecko tracks over 13,000 different crypto assets from more than 600 exchanges worldwide. In addition to monitoring price, volume, and market capitalization, CoinGecko offers fundamental analysis of the crypto market. Users can view cryptocurrency prices based on market cap, 24-hour trading volume, and price charts for various time periods.

Forex and cryptocurrency traders frequently rely on CoinGecko to access the prices of cryptocurrencies they are interested in before making buying or selling decisions. Furthermore, CoinGecko provides information on crypto exchanges, NFTs, learning resources, and other related products on their website.

**Description that cuts across all the frameworks**

**Progress Bar:** A visual element utilized to represent the advancement of the data scraping operation.

**Charging Bar:** This feature aids in tracking the percentage of progress achieved during program execution.

**Error Handling:** When making a request, if any bugs are encountered, it displays the corresponding exception message. In the event of a program failure, it indicates the specific location where the error occurred.

**Elapsed Time:** This printed information indicates the duration it took for the program to complete the website scraping process.

### 1. Beautiful soup

To begin, we import the necessary libraries for our project. We use the "pip install request" command to install the library required for parsing Beautiful Soup, which enables us to extract the desired details during the scraping process.

We also import the "time" library, which allows us to incorporate delays between requests. By doing so, we ensure that the program does not overwhelm the server with excessive requests, potentially leading to IP address blocking.

Additionally, we create headers to provide identification information to the browser. This step helps prevent the browser from mistaking our program for a bot and allows for smoother interactions with the website.

In this stage, we begin utilizing Beautiful Soup. We parse the HTTP response, which is in string format, using the lxml parser. We locate all the table bodies and, under each table body, we find all the table rows.

To store our scraped data, we create containers called "containers" to which we will append the relevant information. The "slugs" serve as unique identifiers to obtain the 100 links for the cryptocurrencies. These slugs are obtained by inspecting the website and extracting their class and "href" attributes.

We proceed to scrape the data, updating our "COINS" dictionary with the scraped information. The scraped data is then appended to the "crypto-coins" container created earlier. We loop through the slugs and append them to the base URL, which corresponds to the URL for coingecko.com.

Next, we create a session to make the request with the necessary headers and scrape the URLs. We locate the class for the name and URLs, and using the "strip" method, we remove any whitespace from the scraped data. At this point, we append a tuple containing the names and URLs to a "bar" variable, which helps us scrape the subsequent data in the code's desired order.

Finally, we create a CSV file to store our data. We open the file, write the "COINS" data to it, and instruct the program to print the data.

### 2. Scrapy

In order to inherit the Spider class from Scrapy, a custom spider was created by subclassing the Spider class. Both scripts include a function to open a file for reading and writing purposes. A new variable called "crypto_body" is created using the CSS method to extract information about the table body and its table rows. The "COINS" dictionary is appended with the relevant data. By looping through the table rows within the table, the script identifies the location of the actual data. This involves stripping each element and appending it to a string as a new URL. The URLs and Coins are then appended to "crypto_slugs". To ensure the printed data (COINS) is displayed properly, the "yield print()" statement is used. This allows the information to be printed without any formatting issues.

### 3. Selenium

To initiate the scraping process, we first import the necessary libraries. Options are added to handle errors, allowing us to comment out or ignore certain errors while still displaying level 3 errors (fatal errors) during code execution. These options are effectively appended to the driver. To prevent excessive requests and potential IP address blocking, the sleep function is incorporated to introduce delays between requests. A function is created to identify the tag name for each table body (tbody) and retrieve the corresponding table rows. A slug is generated to capture the XPath of elements and their hrefs, which are then appended to the crypto URLs.

The index of the desired elements, such as NAME and PRICE, is determined. JSON files, such as bitcoin.org, are dumped to extract the official URLs of the cryptocurrencies. The slug obtained is specifically for coingecko.com and not the official websites. Using the "Driver.get" command, the class under a div is inspected, with ".text" indicating that the scraped data should be in text format. The tag name for the names is also obtained.

Temporary lists are created to store the URLs. The href is extracted and split into separate lines. A new array is then generated based on the split data. The temporary lists are appended, and subsequently, the official websites are appended. The URLs are written to a data frame and saved in a CSV file before printing the official URLs and names of the cryptocurrencies in chronological order as per the code. Finally, all the pages are scraped, and the program is closed and quit.

**Technical Description of Output.**

**RANK:** the position of the crypto in terms of market capitalization

**SYMBOL:** symbol for the cryptocurrency

**PRICE:** the prevailing price of the cryptocurrency. This normally changes from time to time. Unlike forex, cryptocurrencies are continuously traded on weekends.

**MARKET CAPITALISATION**: refers to the total value of all a company's shares of stock. The price of the crypto is the share price of the stock of that particular cryptocurrency.

**FDV (Fully Diluted Valuation):** refers to a measure of a cryptocurrency's total value if all of its tokens or coins were in circulation and fully available on the market.

**Market Cap/FDV (Market Cap to Fully Diluted Valuation ratio)** refers to a financial metric used in cryptocurrency analysis. It compares the current market capitalization of a cryptocurrency to its fully diluted valuation.

**24h Volume**: refers to the total trading volume or total amount of a specific cryptocurrency that has been traded within the last 24 hours.

**1h:** refers to the past one hour.

**24h**: represents the past 24 hours or one day.

**7d:** It stands for the past seven days or one week.

For each scrapper, csv files was created.

It scraped the name of the cryptocurrency, its sign, the price, FDV, market cap, 24 Volume, 1h, 24h, 7d

**Performance of Scrapers**

Below are the recorded running time for the three(3) scrapers used:

- Beautiful Soup: Execution duration: 0:00:01.292009
- Selenium: Execution duration: 0:00:09.366997
- Scrapy: Execution duration: 0:00:09.547461

From the elapse time recorded for each tools above, we can deduce that Beautiful soup is the fastest, followed by Selenium while Scrapy happens to be the slowest among the three.

**Work Description**

Ivan Alexander Moz Prez - Selenium

Xolani Keith Mpala – Beautiful Soup

Xolani Keith Mpala & Ivan Alexander Moz Prez – Scrapy

Xolani Keith Mpala & Ivan Alexander Moz Prez – Description pdf

The whole program was done standardizing the words used in the codes such as coins, message such as print('---successfully created URLS---') so that one can easily follow up easily on the codes from selenium, beautiful soup and scrapy