**Python Project for Data Science**

# **Crowdsourcing Short squeeze Dashboard**

In this module, you will demonstrate your skills in Python - the language of choice for Data Science and Data Analysis. You will apply Python fundamentals, Python data structures, and work with data in Python. By working on a real project, you will model a Data Scientist or Data Analyst's role, and build a dashboard using Python and popular Python libraries using Jupyter notebook.

**Learning Objectives**

* Perform webscraping in Python to obtain data.
* Extract data by using a Python library.

## **Web Scrapping: A Key tool in Data Science:**

**Web Scraping: A Key Tool in Data Science**

**Estimated Effort: 5 mins**

**Introduction**

Web scraping, also known as web harvesting or web data extraction, is a technique used to extract large amounts of data from websites. The data on websites is unstructured, and web scraping enables us to convert it into a structured form.

**Importance of Web Scraping in Data Science**

In the field of data science, web scraping plays an integral role. It is used for various purposes such as:

1. **Data Collection:** Web scraping is a primary method of collecting data from the internet. This data can be used for analysis, research, etc.
2. **Real-time Application:** Web scraping is used for real-time applications like weather updates, price comparison, etc.
3. **Machine Learning:** Web scraping provides the data needed to train machine learning models.

**Web Scraping with Python**

Python provides several libraries for web scraping. Here are some of them:

1. **BeautifulSoup:** BeautifulSoup is a Python library used for web scraping purposes to pull the data out of HTML and XML files. It creates a parse tree from page source code that can be used to extract data in a hierarchical and more readable manner.
2. from bs4 import BeautifulSoup
3. import requests
4. URL = "http://www.example.com"
5. page = requests.get(URL)
6. soup = BeautifulSoup(page.content, "html.parser")

Copied!

1. **Scrapy:** Scrapy is an open-source and collaborative web crawling framework for Python. It is used to extract the data from the website.
2. import scrapy
3. class QuotesSpider(scrapy.Spider):
4. name = "quotes"
5. start\_urls = ['http://quotes.toscrape.com/tag/humor/',]
6. def parse(self, response):
7. for quote in response.css('div.quote'):
8. yield {'quote': quote.css('span.text::text').get()}

Copied!

1. **Selenium:** Selenium is a tool used for controlling web browsers through programs and automating browser tasks.
2. from selenium import webdriver
3. driver = webdriver.Firefox()
4. driver.get("http://www.example.com")

Copied!

**Applications of Web Scraping**

Web scraping is used in various fields and has many applications:

1. **Price Comparison:** Services such as ParseHub use web scraping to collect data from online shopping websites and use it to compare the prices of products.
2. **Email address gathering:** Many companies that use email as a medium for marketing, use web scraping to collect email ID and then send bulk emails.
3. **Social Media Scraping:** Web scraping is used to collect data from Social Media websites such as Twitter to find out what's trending.

**Conclusion**

Web scraping is an essential skill in the fast-growing world of data science. It provides the ability to turn the web into a source of data that can be analyzed, processed, and used for a variety of applications. However, it's important to remember that one should use web scraping responsibly and ethically, respecting the terms of use or robots.txt files of the websites being scraped.

## **HTML for Web Scraping**

Simple Explanation:

* **HTML Basics**: The video introduces **HTML (Hypertext Markup Language)**, which is the language used to create web pages. It helps in displaying information on the internet.
* **HTML Tags**: Web pages are made up of **tags** (like <h3> for headings and <p> for paragraphs) that tell the browser how to show the content. For example, <h3> makes text larger and bold.
* **HTML Structure**: The video explains that an HTML document has a structure similar to a tree, where:
  + The **root** is the <html> tag.
  + Inside it, there are two main parts: the <head> (which contains information about the page) and the <body> (which shows the content).
* **HTML Tables**: It also covers how to create tables using tags like <table>, <tr> (for table rows), and <td> (for table cells).

Summary:

The video provides an overview of HTML for web scraping, explaining how to read and understand the structure of a web page. It covers:

* The role of HTML tags in displaying content.
* The hierarchical structure of HTML documents, likening it to a family tree.
* How to create tables in HTML.

This foundational knowledge is essential for extracting data from web pages using Python.

## **Web Scrapping**

Key Points:

* **Definition**: Web scraping is the process of extracting data from websites without having to copy and paste manually.
* **Tools Used**: It involves using Python code along with two libraries: **Requests** (to download the webpage) and **BeautifulSoup** (to parse the HTML).
* **BeautifulSoup Objects**: This library helps represent the HTML structure as a tree, allowing you to navigate and extract specific data easily.
* **Methods**:
  + The **find\_all** method is used to filter and retrieve specific tags from the HTML.
  + You can access elements like tags, attributes, and text using this method.
* **Example**: If you wanted to find player names and salaries from a sports website, you would use these tools to scrape that information efficiently.

Summary:

The video teaches you how to use web scraping to gather data from websites quickly and efficiently using Python, Requests, and BeautifulSoup. It explains how to navigate the HTML structure and extract specific information, making data collection much easier than manual methods.