👩‍🏫 Lesson 1: Introduction to Web Scraping with BeautifulSoup & Requests

Welcome to the first lesson of our beginner-friendly web scraping series! In this lesson, we explore two essential Python libraries: **Requests** and **BeautifulSoup**. These tools are perfect for beginners who want to start extracting data from websites in a simple and efficient way.

🌐 What is Web Scraping?

Web scraping is the process of automatically extracting information from websites. It's a powerful way to gather data for analysis, automation, or personal projects. In this series, we're focusing on scraping **static web pages**—pages that don't change dynamically.

📦 Tools You'll Learn

- **Requests**: Allows us to access the content of a web page by sending an HTTP request and getting a response.
- **BeautifulSoup**: Helps us organize and parse messy HTML code into a format that is easier to work with.

🛠️ What You'll Do in This Lesson

1. **Import the Tools**
   You'll begin by importing the necessary libraries—Requests and BeautifulSoup.
2. **Access a Web Page**
   You'll choose a website and send a request to it to fetch the entire HTML content.
3. **Understand the Response**
   After sending the request, you'll check the response status (like 200 for success, 404 for not found, etc.) to ensure everything is working.
4. **Create a "Soup" Object**
   Once the raw HTML is fetched, you'll use BeautifulSoup to convert it into a navigable format. This cleaned-up structure is referred to as "soup."

5. **Visualize the HTML**
   You'll use a method to neatly display the HTML in a more readable, indented format. This helps in understanding the layout and preparing for the next step—extracting the data you want.

💡 Key Concepts Introduced

- The idea of **parsing HTML** from a website.
- How a website's HTML structure looks and how we can view it.
- Using a tool that simplifies messy code and prepares it for data extraction.

🔍 What's Coming Next

In the following lessons, you'll learn how to search through the HTML using **tags**, **classes**, and **attributes**. You'll also dive into tools like `find()` and `find_all()` to pull out exactly what you need.

And at the end of this beginner series, there will be a **mini project** where you extract table data from a web page and organize it into a **pandas DataFrame** for analysis.