CODING NINJAS

Data Science

Web Scraping

Web Scraping

- Web scraping is a technique for extracting information from the internet automatically using our script that simulates human web surfing.
- Web scraping helps us extract large volumes from different websites

Scraping Rules

- Check a website’s Terms and Conditions before you scrape it.
- Do not spam the website by making a lot of requests to a specific web page.
- Update your code time to time

Libraries Used

- BeautifulSoup
- Selenium
- Scrapy

Process

- Find the URL that you want to scrape
- Send an HTTP request to that URL and get the HTML as response
- Parse the HTML content
- Inspect the web page and find data that we want to extract
- Extract required data and store it data in the required format

Web Page

Web Page Structure

- HTML
- CSS
- JavaScript
- Media content

HTML Tour

HTML Tags

- <html>
- <head> and <title>
- <body>
- Heading tags <h1><h2>....<h6>
- <p>
- <a>
- <img>
- <table>

HTML - Relative Tag Names

- Child
- Parent
- Sibling

HTML

- Class
- ID

BeautifulSoup

Steps

- Load HTML
- Parse HTML
- Locate and extract the desired data

Methods & Attributes

- prettify()
- page.tag
  - page.tag.name
  - page.tag.string
  - page.tag.attrs
    - Using get()
    - Access like dictionary
- get_text()

Methods & Attributes

- find()
- find_all()

Navigate Tree

- Searching Parse Tree
- Going up
- Going down
- Going sideways
- Going back & forth

Searching Parse Tree

- find_all()
  - A string
  - A list
  - True
  - Using id
  - Using class
  - Using CSS selector

Going down

- Navigating using tag names
  - We can use nested tag names also
- .string
- .strings and .stripped_strings
- .contents and .children
- .descendants

Going Up

- .parent
- .parents

Going sideways

- .next_sibling and .previous_sibling
- .next_siblings and .previous_siblings

Going Back & forth

- .next_element and .previous_element
- .next_elements and .previous_elements