Relatório 8 - Prática: Web Scraping com Python p/ Ciência de Dados (II)

Higor Miller Grassi

No vídeo é tratado sobre o web scraping, que é o método de coletar dados de sites da internet, seja de forma manual ou através de ferramentas e bibliotecas que automatizam esse processo, permitindo a extração de informações diretamente de páginas da web. Durante o processo, utiliza-se código que aproveita tags HTML para navegar e identificar os elementos desejados.

Para realizar web scraping em sites reais, é fundamental ter conhecimento em Python e familiaridade com suas bibliotecas relacionadas:

- Biblioteca Requests: Responsável por fazer requisições HTTP e acessar o conteúdo da página desejada, onde o método requests.get(url) é amplamente utilizado para obter o código-fonte HTML do site.
- Biblioteca BeautifulSoup: Usada para analisar o conteúdo HTML recebido, permitindo navegar na estrutura do documento e localizar as informações de interesse, normalmente, o HTML é analisado com o lxml para interpretar o código.

A coleta de dados geralmente envolve a identificação das tags que contêm as informações relevantes e a extração delas com base em atributos como class, id etc

Primeira atividade prática em vídeo:

```python
from bs4 import BeautifulSoup
import requests
import time

#abrir o arquivo html apenas para leitura e apos, e impresso o q esta escrito no arquivo html
with open("home.html","r") as html_file:
content = html_file.read()
print(content)

#imprimirá a mesma coisa que a anterior, mas o lxml e mais eficiente para documentos maiores
soup = BeautifulSoup(content,"lxml")
print(soup.prettify())

#print de todos o h5
tags = soup.find_all("h5")
print(tags)

#Imprime o nome dos cursos
courses_html_tags = soup.find_all("h5")
for course in courses_html_tags:
    print(course.text)

#Encontra todos os elementos <div> com a classe "card" na página HTML
course_cards = soup.find_all("div", class_="card")
for course in course_cards:
    #Extrai o nome do curso, que está dentro da tag <h5>
    course_name = course.h5.text
    #Faz a mesma coisa que o anterior, mas divide o texto para pegar o ultimo elemento(neste caso o preco)
    course_price = course.a.text.split()[-1]

    print(f"{course_name} costs {course_price}")
```

Segunda Atividade:

```python
def find_jobs():
    html_text = requests.get("https://g1.globo.com/").text #Vai no site
    soup = BeautifulSoup(html_text, "lxml") #Junta o HTML com a biblioteca BeutifulSouop
    #jobs = soup.find_all("div", class_="feed-post-body") #procura todos os div q tiver a class com este nome
    #print(jobs)

    jobs = soup.find_all("div", class_="glb-grid")
    for job in jobs:
        print('Put some skill that you are not familiar with')
        unfamiliar_skill = input('>')
        print(f'Filtering out {unfamiliar_skill}')
        job_date = job.find("span", class_="feed-post-datetime").text.strip()
        if 'few' in job_date:
            company_name = job.find("div", class_="feed-post-header with-post-chapeu").text.replace("","")
            notice = job.find("div", class_="feed-post-body-title gui-color-primary gui-color-hover").text.replace("","")
            #procura o link da materia
            links = job.h2.a["href"]
            if unfamiliar_skill not in notice:
                with open("posts/{index}.txt', 'w'") as f:
                    f.write(f"Company Name: {company_name.strip()}\n - Notice: {notice.strip()}\n - Publication Date: {job_date}\n - More Info: {links}\n")

                print(f'file saved {index}')

if __name__  == "__main__":
    while True:
        find_jobs()
        time_wait = 10
        print(f"Waiting{time_wait} seconds...")
        time.sleep(time_wait*60)
```

Conclusão: O web scraping é uma técnica essencial para coletar dados diretamente da internet, sendo amplamente utilizada em ciência de dados, pesquisa de mercado e análise de tendências, juntamente com as ferramentas como BeautifulSoup e Requests tornam o processo acessível e eficiente, permitindo que grandes volumes de informações sejam coletados, estruturados e analisados, após aprender a adicionar funcionalidades ao projeto, como filtros, automação de notificações e armazenamento eficiente,o processo fica bem mais automatizado.