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

Descrição da Atividade

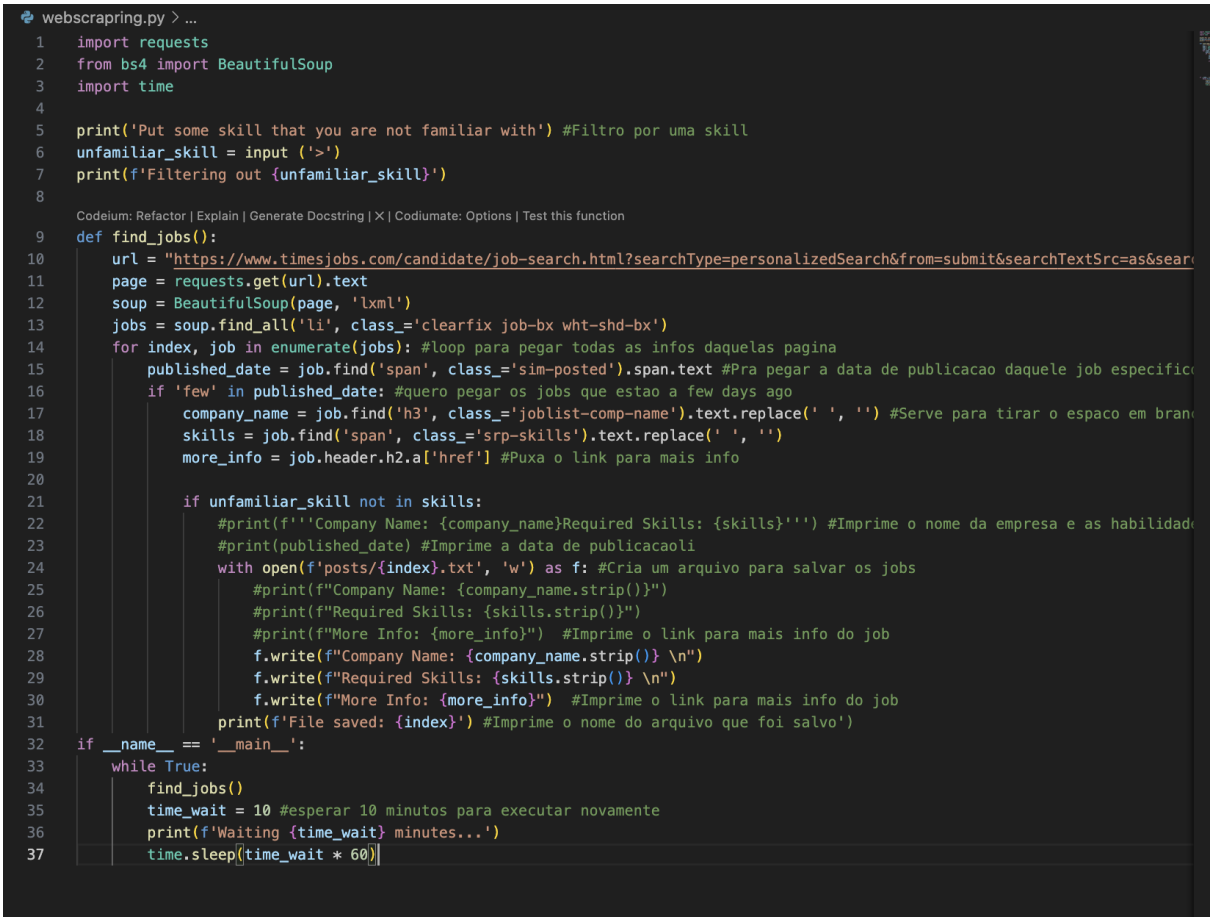
Realização de WebScraping, consiste em uma técnica para captar e armazenar dados que estão na Web, nesse caso do vídeo foi retirado dados do site 'time jobs'. Para ler a página foi utilizado a biblioteca BeautifulSoup.

Nas linhas 5,6,7 do código adicionamos um filtro para o usuário inserir se necessário.

Na função Find_Jobs seguimos esse passo a passo:

- Ler Url e a página;
- Identificar qual classe do Html queremos pegar;
- Fizemos um loop para pegar percorrer todas as informações que queremos:
  - Data de Publicação;
  - Filtramos apenas os jobs que as datas que estão como 'few days ago';
  - Nome da Empresa;
  - Habilidades Requisitadas;
  - Mais Informações;
  - Obs: Para pegar essas informações basta indicar em qual parte do Html contém ela.
- Condicional para filtrar alguma Skill que meu usuário colocou la em cima;
- Criação de arquivo para salvar as informações capturadas (neste caso em arquivo txt);
- Escreve no arquivo Txt, Nome da Companhia, Skills Requisitadas e Mais Infos;
- Salvo o Arquivo.

Chama a função principal para rodar o código e adiciona um tempo de espera para re-executar o código.



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

print('Put some skill that you are not familiar with') #Filtro por uma skill
unfamiliar_skill = input ('>')
print(f'Filtering out {unfamiliar_skill}')

def find_jobs():
    url = "https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&searchTextSrc=as&sear
    page = requests.get(url).text
    soup = BeautifulSoup(page, 'lxml')
    jobs = soup.find_all('li', class_='clearfix job-bx wht-shd-bx')
    for index, job in enumerate(jobs): #loop para pegar todas as infos daquelas pagina
        published_date = job.find('span', class_='sim-posted').span.text #Pra pegar a data de publicacao daquele job especific
        if 'few' in published_date: #quero pegar os jobs que estao a few days ago
            company_name = job.find('h3', class_='joblist-comp-name').text.replace(' ', '') #Serve para tirar o espaco em bran
            skills = job.find('span', class_='srp-skills').text.replace(' ', '')
            more_info = job.header.h2.a['href'] #Puxa o link para mais info

            if unfamiliar_skill not in skills:
                #print(f'''Company Name: {company_name}Required Skills: {skills}''') #Imprime o nome da empresa e as habilidad
                #print(published_date) #Imprime a data de publicacaoli
                with open(f'posts/{index}.txt', 'w') as f: #Cria um arquivo para salvar os jobs
                    #print(f"Company Name: {company_name.strip()}")
                    #print(f"Required Skills: {skills.strip()}")
                    #print(f"More Info: {more_info}")  #Imprime o link para mais info do job
                    f.write(f"Company Name: {company_name.strip()} \n")
                    f.write(f"Required Skills: {skills.strip()} \n")
                    f.write(f"More Info: {more_info}")  #Imprime o link para mais info do job
                print(f'File saved: {index}') #Imprime o nome do arquivo que foi salvo')
if __name__ == '__main__':
    while True:
        find_jobs()
        time_wait = 10 #esperar 10 minutos para executar novamente
        print(f'Waiting {time_wait} minutes...')
        time.sleep(time_wait * 60)
```

Conclusões

Portanto, WebScraping é uma forma de retirar e armazenar informações que estão na Web de forma automatizada, neste caso utiliza-se python para viabilizar.

Referências

Vídeo: Web Scraping with Python - Beautiful Soup Crash Course. Disponível em <https://www.youtube.com/watch?app=desktop&v=XVv6mJpFOb0> Acessado em: 09/09/2024