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

Thassiana C. A. Muller

■ Web Scraping with Python - Beautiful Soup Crash Course

Para realizar a coleta de informações de uma página web com o python é necessário primeiro entender que o seu esqueleto é composto por tags html que delimitam as regiões de cada informação que irá definir a página. Após o entendimento geral das tags, é preciso saber o básico de manipulação de arquivos com python e do funcionamento das bibliotecas "resquests' e "BeaultifulSoup", conhecimentos de expressões regulares não são obrigatórios porém são desejados.

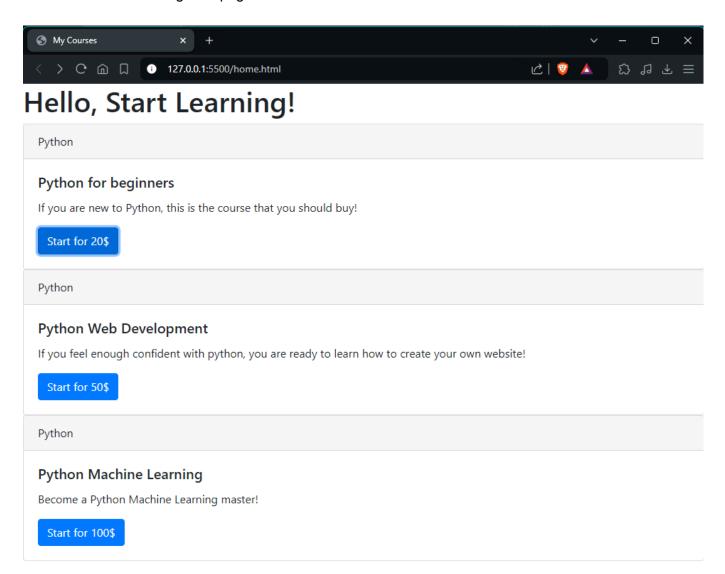
Elementos HTML

As principais tags html são:

- <html>: Tag raiz que envolve todo o conteúdo de uma página HTML.
- <head>: Contém metadados, links para arquivos externos (CSS, scripts) e outras informações não exibidas diretamente na página.
- <title>: Define o título da página mostrado na aba do navegador.
- <body>: Contém todo o conteúdo visível da página (texto, imagens, vídeos, etc.).
- 2. Cabeçalhos e Parágrafos
- <h1> a <h6>: Tags de cabeçalho, <h1> sendo o mais importante e <h6> o menos importante.
- : Define um parágrafo de texto.
- <a>: Define um hyperlink. Atributo href especifica o destino do link.
- : Insere uma imagem. Atributos src e alt especificam o caminho da imagem e o texto alternativo, respectivamente.
- <header>: Define um cabeçalho para um documento ou seção.
- <div>: Define uma divisão ou seção de página (bloco).
- : Define uma seção de texto (em linha).
- <script>: Define scripts de programação (JavaScript).
- link>: Define a relação entre o documento atual e um recurso externo (geralmente usado para CSS).

Web Scraping

Dado a seguinte página web:



Pode-se analisar a estrutura da página com o Python da seguinte forma:

```
with open('home.html', 'r') as html_file:
      content = html_file.read()
      print(content)
✓ 0.0s
<!doctype html>
<html lang="en">
  <head>
     <meta charset="utf-8">
     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" ir</pre>
     <title>My Courses</title>
  </head>
  <body>
     <h1>Hello, Start Learning!</h1>
     <div class="card" id="card-python-for-beginners">
        <div class="card-header">
           Python
        </div>
        <div class="card-body">
           <h5 class="card-title">Python for beginners</h5>
           If you are new to Python, this is the course that you should buy!
           <a href="#" class="btn btn-primary">Start for 20$</a>
        </div>
     </div>
     <div class="card" id="card-python-web-development">
```

Observa-se que, em um primeiro momento, os títulos dos cursos estão dentro das tags <h5></h5>. Então, para filtrar apenas os títulos dos cursos pode-se fazer algo como:

Para encontrar os valores de cada curso pode ser feito como abaixo, onde encontra-se, primeiramente, todas as tags relacionadas e após isso itera-se para printar seus valores

Conclusão

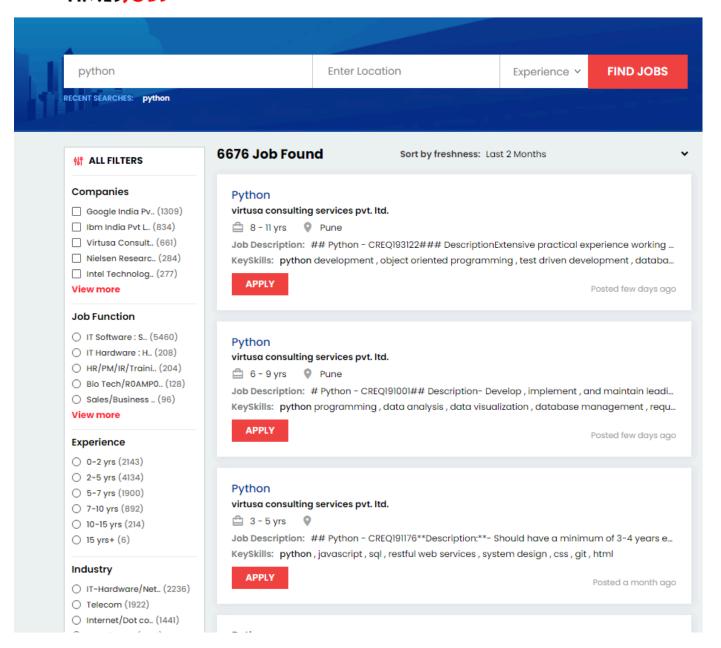
Assim, conclui-se que o principal fator para um scraping limpo é a inspeção adequada de padrões dos elementos das páginas que serão analisadas e um tratamento adequado em python a depender do escopo final da aplicação.

Referências

Web Scraping with Python - Beautiful Soup Crash Course

Para realizar o scraping de ur	m site de vagas de empre	ego como este:	

TIMESJOBS



Pode-se utilizar uma função para procurar e formatar as informações corretamente como abaixo, observa-se que a funções poderá receber ou não um vetor de palavras chaves para procurar skills específicas:

Obs: O código pode ser em encontrado em anexo ao card

```
from bs4 import BeautifulSoup
import requests
import re
import time
def find_jobs(user_skills=None):
   # fazer o request do site com o python
   html =
requests.get('https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=submit&searc
hTextSrc=&searchTextText=&txtKeywords=python&txtLocation=')
   soup = BeautifulSoup(html.text, 'lxml') # objeto da BeautifulSoup criado utilizando o parser "lxml"
   jobs = soup.find_all('li', class_='clearfix job-bx wht-shd-bx') #ocorrencias de todas as tags "li" no html
   for job in jobs: # é realizado o scraping em cada vaga de emprego
       posted = job.find('span', class_='sim-posted')
       if posted.span.text == 'Posted few days ago': # se é uma vaga recente
            company name = job.find('h3', class ='joblist-comp-name').text.strip() # agrupa titulos e remove
espaços em branco desnecessários
            skills = re.sub(r'\s+,',',', (job.find('span', class_='srp-skills').text.strip())) # filtro de regex
para melhorar a formataçao final da resposta
            skills_vector = [skill.strip() for skill in skills.split(',')] # list comprehension para separar as
skills em um vetor para, posteriormente, verificar se bate com o parametro de entrada
           more_info = job.find('a')['href']
            if user_skills is not None: # filtra apenas os trabalhos com as skills passadas para a função
                for user_skill in user_skills:
                    if any(user skill.lower() in skill.lower() for skill in skills vector):
                        print(f'Company Name: {company_name}\nRequired Skills: {skills}\nMore info:
{more_info}\n')
                       break
            else:
                print(f'Company Name: {company_name}\nRequired Skills: {skills}\nMore info: {more_info}\n')
if name ==' main ':
   user_skills = (input("Which skills do you have (comma separated):"))
   user_skills = re.sub(r'\s+', '', user_skills).split(',')
   find jobs(user skills)
   wait = 10
```

```
print(f"Waiting {wait} minutes...")
   time.sleep(wait*60)
   time.sleep(wait*60)
     O retorno será:
PS D:\Drive Google\UTFPR\Lamia\Atividade8> python .\main.py
Which skills do you have (comma separated):git
Company Name: virtusa consulting services pvt. ltd.
Required Skills: python development, object oriented programming, test driven development,
database connectors, linux environment, git, api
More info:
https://www.timesjobs.com/job-detail/python-virtusa-consulting-services-pvt-ltd-pune-8-to-11-yrs-jobid
-PRoGGHjEoVVzpSvf__PLUS__uAgZw==&source=srp
Company Name: LAKSH HUMAN RESOURCE
Required Skills: rest, python, django, git
More info:
https://www.timesjobs.com/job-detail/python-developer-laksh-human-resource-mumbai-1-to-3-yrs-jobid-uUE
qcx71MRdzpSvf__PLUS__uAgZw==&source=srp
Company Name: 3RI Technologies Pvt Ltd
Required Skills: python, database, security, django, git, mobile
More info:
https://www.timesjobs.com/job-detail/python-developer-3ri-technologies-pvt-ltd-pune-0-to-1-yrs-jobid-S
9R7triGKgpzpSvf__PLUS__uAgZw==&source=srp
```

Waiting 10 minutes...