# Relatório do bootcamp - 8

Thassiana C. A. Muller

■ Web Scraping with Python - Beautiful Soup Crash Course

#### Elementos HTML

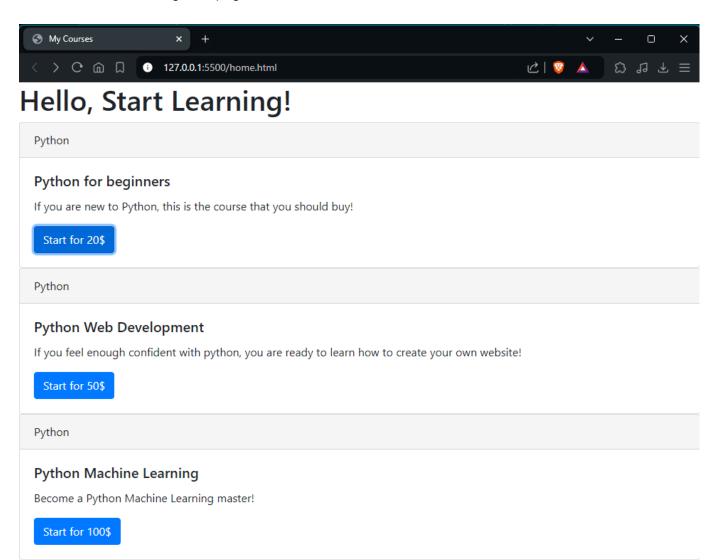
Para realizar a coleta de informações de uma página web é necessário primeiro entender que o seu esqueleto é composto por tags que delimitam as regiões de cada informação que irá definir a página.

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.
- <img>: 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).
- <span>: 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:

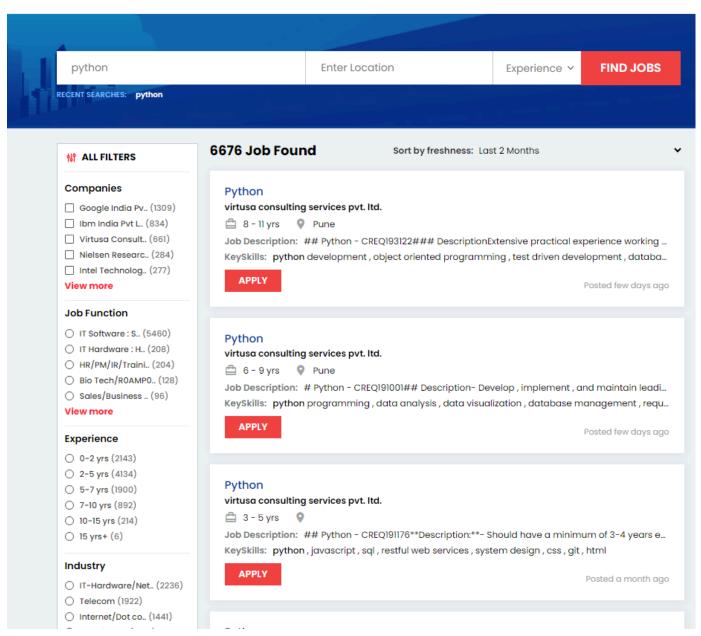
```
with open('home.html', 'r') as html_file:
      content = html_file.read()
      soup = BeautifulSoup(content, 'lxml')
      print(soup.prettify())
✓ 0.0s
                                                                                                       Python
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8"/>
 <meta content="width=device-width, initial-scale=1, shrink-to-fit=no" name="viewport"/>
 <link crossorigin="anonymous" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css</pre>
 <title>
  My Courses
 </title>
</head>
<body>
 <h1>
  Hello, Start Learning!
 <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!
```

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 realizar o scraping de um site de vagas de emprego como este:

TimesJobs JobBuzz Gulf Jobs Career Insight

### **TIMESJOBS**



Pode-se utilizar uma função para procurar e formatar as informações corretamente como abaixo:

```
from bs4 import BeautifulSoup
import requests
import re
import time
def find_jobs(user_skills=None):
   html =
requests.get('https://www.timesjobs.com/candidate/job-search.html?searchType=personalizedSearch&from=s
ubmit&searchTextSrc=&searchTextText=&txtKeywords=python&txtLocation=')
   soup = BeautifulSoup(html.text, 'lxml')
   jobs = soup.find all('li', class = 'clearfix job-bx wht-shd-bx')
   for job in jobs:
        posted = job.find('span', class_='sim-posted')
        if posted.span.text == 'Posted few days ago':
            company_name = job.find('h3', class_='joblist-comp-name').text.strip()
            skills = re.sub(r'\s+,',',', (job.find('span',class_='srp-skills').text.strip()))
            skills_vector = [skill.strip() for skill in skills.split(',')]
            more info = job.find('a')['href']
            if user_skills is not None:
                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)
```

#### 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...
```