web_scraping_wiki

May 17, 2019

Table of Contents

Autor: Matheus de Vasconcellos Barroso

1 Web scraping com Python

1.1 Introdução

Aprenderemos alguns conceitos básicos de **web scraping** e como utilizar o Python para essa tarefa. Mais precisamente esse material foi preparado utilizando um Jupyter Notebook e algumas extensões úteis (`nbextensions`). O exemplo que utilizaremos será a definição de **web scraping** pela Wikipedia:

In [48]:
```
#from IPython.display import display, HTML
from IPython.display import IFrame

IFrame(src = 'https://en.wikipedia.org/wiki/Web_scraping', width = 900, height=650)
```

Out[48]:
```
<IPython.lib.display.IFrame at 0x902e0f0>
```

Imagine como seria útil desenvolver uma técnica para ler o primeiro ou **n** primeiros parágrafos de uma página da Wikipedia para testar algorítmos de sumarização? Utilizando as técnicas de raspagem de dados podemos obter de forma simples os três primeiros parágrafos da url:

In [50]:
```
import requests
from bs4 import BeautifulSoup

page = requests.get("https://en.wikipedia.org/wiki/Web_scraping")
soup = BeautifulSoup(page.content, 'html.parser')
for item in soup.find_all('p')[:3]: print(item.get_text())
```

```
Web scraping, web harvesting, or web data extraction is data scraping used for extracting data

Web scraping a web page involves fetching it and extracting from it.[1][2] Fetching is the dow

Web scraping is used for contact scraping, and as a component of applications used for web ind
```

Para alcançar esse objetivo final serão abordados alguns tópicos necessários para compreender como funciona esse processo: + HTML + CSS (Não será abordado) + JS (Não será abordado)

Após um melhor entedimento teórico utilizaremos algumas bibliotecas do Python para realizar essa tarefa como `requests` e `bs4`. Algumas importante, mas que não serão utilizadas: `scrapy` e `selenium`

2 Conceitos básicos

2.1 HTML

HyperText Markup Language (HTML) ou Linguagem de Marcação de Hipertexto é uma linguaguem que server para a construção de páginas web. Ela serve para informar ao navegador como exibir o conteúdo da página.

O HTML é formado por tags, é importante compreendê-las e reconhecê-las já que fascilitam o trabalho de raspagem de dados na web. Algumas tags principais: - **html**: informar ao navegador aonde temos código em HTML - **head**: contêm informações como o título da página - **body**: é aonde o conteúdo principal da página está inserido, usualmente é onde o scraping ocorre. - **p**: delimita um parágrafo - **a**: para links - **div**: aponta uma região na página, útil para dividir o conteúdo - **b**: texto em negrito - **i**: texto em itálico - **table**: cria uma tablea - **form**: cria um formulário

2.2 CSS

Cascading Style Sheets (CSS) é um mecanismo para adicionar estilos (cores, fontes, espaçamento, etc.) a um documento web

2.3 JS

JavaScript (JS) é uma linguagem de programação que adiciona interatividade às paginas web.

2.4 Hands On

2.4.1 Explicando como extraimos a explicação da Wikipedia

Primeiro precisamos importas as bibliotecas:

```
In [3]: import requests
        from bs4 import BeautifulSoup
```

Posteriormente, precisamos obter o conteúdo da página (HTML, CSS, JS). Podemos utilizar o método `get`do módulo **requests** para a *url* desejada e atribuí-lo à variável 'page':

```
In [4]: page = requests.get("https://en.wikipedia.org/wiki/Web_scraping")
```

Se a conexão for bem sucedida teremos um status 200 para a página:

```
In [52]: page.status_code
```

```
Out[52]: 200
```

Podemos obter o conteúdo da página:

```
In [10]: page.content[:1000]
```

```
Out[10]: b'<!DOCTYPE html>\n<html class="client-nojs" lang="en" dir="ltr">\n<head>\n<meta chars
```

Observe que essa forma de exibição não é muito fácil e podemos utilizar o BeautifulSoup para nos auxiliar:

```
In [17]: from bs4 import BeautifulSoup

         soup = BeautifulSoup(page.content, 'html.parser')
         #soup
```

É possível melhorar ainda mais usando o método `prettify`

```
In [18]: #print(soup.prettify())
```

Podemos utilizar um seletor CSS para extrair somente o corpo do HTML, veja que agora já está mais fácil encontrar o primeiro parágrafo, basta localizar o < p> a esquerda:

```
In [21]: #print(soup.find('body').prettify()) # é uma outra opção via find
         #soup.select("html body")
```

Podemos selecionar o primeiro parágrafo utilizando o método `find` para localizar uma *tag* **HTML** e o `get_text` para extrair somente o texto:

```
In [71]: print(soup.find('p').get_text())
```

```
Web scraping, web harvesting, or web data extraction is data scraping used for extracting data
```

Veja que estamos próximos do nosso objetivo final, basta usar o método `find_all` que conseguiremos econtrat todas as tags **p**'s. Atente que os resultados serão retornados em uma lista:

```
In [73]: for item in soup.find_all('p')[:3]: print(item.get_text())
```

```
Web scraping, web harvesting, or web data extraction is data scraping used for extracting data

Web scraping a web page involves fetching it and extracting from it.[1][2] Fetching is the dow

Web scraping is used for contact scraping, and as a component of applications used for web inde
```

Resumindo, poderíamos ter utilizado somente os seguintes comandos:

```
In [74]: import requests
         from bs4 import BeautifulSoup

         page = requests.get("https://en.wikipedia.org/wiki/Web_scraping")
         soup = BeautifulSoup(page.content, 'html.parser')
         for item in soup.find_all('p')[:3]: print(item.get_text())
```

```
Web scraping, web harvesting, or web data extraction is data scraping used for extracting data

Web scraping a web page involves fetching it and extracting from it.[1][2] Fetching is the dow

Web scraping is used for contact scraping, and as a component of applications used for web ind
```

Para fixar as ideias veja como seria simples retornar os dois primeiros parágrafo na wikipédia sobre HTML, CSS e JS em três passos: + Lista de *url*'s + Função para retornar o parágrafo + Loop para aplicar a função a cada item na lista

Passo 1:

```
In [105]: paginas = [
              'https://pt.wikipedia.org/wiki/HTML',
              'https://pt.wikipedia.org/wiki/Cascading_Style_Sheets',
              'https://pt.wikipedia.org/wiki/JavaScript'
          ]

          paginas

Out[105]: ['https://pt.wikipedia.org/wiki/HTML',
           'https://pt.wikipedia.org/wiki/Cascading_Style_Sheets',
           'https://pt.wikipedia.org/wiki/JavaScript']
```

Passo 2:

```
In [106]: def retorna_n_paragrafos(url, n = 1):
              '''
              Essa função retorna os n primeiros parágrafos da url selecionada.
              Possui os seguintes argumentos:
              @url: a url desejada
              @n: número de parágrafos para retornar, default = 1
              '''

              page = requests.get(url)
              soup = BeautifulSoup(page.content, 'html.parser')
              paragrafos = soup.find_all('p')[:n]
              paragrafos_texto = []
              for item in paragrafos: paragrafos_texto.append(item.get_text())
```

```
            return '/n'.join(paragrafos_texto) #concatenar elementos e separar por nova linh

        print(retorna_n_paragrafos('https://pt.wikipedia.org/wiki/HTML', n = 2))
```

```
HTML (abreviação para a expressão inglesa HyperText Markup Language, que significa Linguagem de
/nHyTime é um padrão para a representação estruturada de hipermídia e conteúdo baseado em tempo
```

Vemos que a função funciona, o primeiro /n ocorre porque o primeiro parágrafo não tinha conteúdo.

Passo 3 (criando o loop):

```
In [107]: for pagina in paginas:
              out = retorna_n_paragrafos(pagina, n = 2)
              print(pagina + '\n' + out)
```

```
https://pt.wikipedia.org/wiki/HTML
HTML (abreviação para a expressão inglesa HyperText Markup Language, que significa Linguagem de
/nHyTime é um padrão para a representação estruturada de hipermídia e conteúdo baseado em tempo

https://pt.wikipedia.org/wiki/Cascading_Style_Sheets
Cascading Style Sheets (CSS) é um mecanismo para adicionar estilo (cores, fontes, espaçamento,
/nO código CSS pode ser aplicado diretamente nas tags ou ficar contido dentro das tags <style>

https://pt.wikipedia.org/wiki/JavaScript
JavaScript , frequentemente abreviado como JS, é uma linguagem de programação interpretada de a
/nÉ atualmente a principal linguagem para programação client-side em navegadores web. É também
```

```
In [ ]: Fim.
```