VOLTAR

Info Gather

Apresentar um recurso em Python, o Beautiful Soup, para coletar informações da WEB.

NESTE TÓPICO

Marcar tópico

Olá alunos,

Vamos tomar uma maravilhosa sopa? (do jeito que está frio hoje, até que é uma boa ideia) mas, na realidade a Beautiful Soup (Sopa Bonita) é um recurso utilizado para coletar informações na WEB. É um recurso do Python muito utilizado na ciência de dados.

O módulo Beautiful Soup está relacionado com a técnica de WEB Scraping (raspando a WEB). Os próprios fabricantes de navegadores, como Google Chrome® e Firefox®, por exemplo, também possuem estes recursos que realizam esta técnica em seus navegadores.

BEAUTIFUL SOUP

Beauiful Soup permite que você “raspe” um determinado site e extraia informações, podendo utilizar filtros. Vamos ver como podemos realizar este processo em Python..

Primeiramente, consideraremos a construção de páginas em HTML, lembrando que o HTML não é uma linguagem procedural e sim uma linguagem de marcação, com tags. E as tags são iniciadas e finalizadas com marcações, como por exemplo: (início) e (finalizar), < a > e < /a >.

O módulo Beautiful Soup consegue “parsear” (interpretar) código HTML e assim utilizar parâmetros para filtrar informações da página.

Teremos que utilizar as bibliotecas: **requests** (solicitações) e a **BeautifulSoup**.

```
1. import requests
2. from bs4 import BeautifulSoup
```

Vamos utilizar como exemplo, o site da Uninove:

```
1. import requests
2. from bs4 import BeautifulSoup
3.
4. pagina = requests.get('http://www.uninove.br')
5.
6. sopa = BeautifulSoup(pagina.text, 'html.parser')
```

Como a BeautifulSoup só rastreia (parser) e não pega a url do site, então antes utilizamos o método **get** da biblioteca **requests (requests.get)**, passamos a url da Uninove e atribuímos à variável **pagina (linha 4).**

Ai sim, em seguida utilizamos a **BeautifulSoup** para "parsear" o código HTML do site da Uninove **(pagina.text, 'html.parser')** e atribuímos na variável **sopa (linha 6)**.

Agora, vamos criar um **script** com um nome, por exemplo, **links_uninove.py** e acrescentar as seguintes linhas:

```
1. import requests
2. from bs4 import BeautifulSoup
3.
4. pagina = requests.get('http://www.uninove.br')
5.
6. sopa = BeautifulSoup(pagina.text, 'html.parser')
7.
8. # encontra todos os links
9. uninove = sopa.find_all('a')
10.
11. for i in uninove:
12.     print(i.prettify())
13.
14. input('Tecle ENTER para sair...')
```

Os links de uma página em HTML são definidos com a tag < **a** > por isso utilizamos a função **find_all** passando o parâmetro **('a')** para encontrar todos os links da página da Uninove e atribuímos à variável **uninove (linha 9).**

Utilizamos um loop com **for** e com a função **prettify()** verifica a estrutura do código HTML, no caso os links e com a função **print** mostramos todos os links da página Uninove.

Ao rodarmos este script, todos os links da página da Uninove e mais outras informações foram mostradas, mas poderemos melhorar esta busca:

```
1.  import requests
2.  from bs4 import BeautifulSoup
3.
4.  pagina = requests.get('http://www.uninove.br')
5.
6.  sopa = BeautifulSoup(pagina.text, 'html.parser')
7.
8.  # encontra todos os links
9.  uninove = sopa.find_all('a')
10.
11. for i in uninove:
12.     print(i.get("href"))
13.
14. input('Tecle ENTER para sair...')
```

Mudamos o bloco **for** utilizando o método **get** para apanhar somente os links do site por causa do parâmetro: **("href") (linha 12)**. O resultado seria algo assim:

```
1.  http://www.uninove.br/unidade/memorial/
2.  http://www.uninove.br/unidade/vergueiro/
3.  http://www.uninove.br/unidade/vila-maria/
4.  http://www.uninove.br/unidade/vila-prudente/
5.  http://www.uninove.br/unidade/santo-amaro/
6.  http://www.uninove.br/unidade/maua/
7.  http://www.uninove.br/unidade/sao-bernardo-do-campo-2/
8.  http://www.uninove.br/unidade/osasco-2/
9.  http://www.uninove.br/unidade/guarulhos/
10. http://www.uninove.br/unidade/cotia/
```

Vamos apanhar outras informações do site, criando outro script com o nome: **paragrafo_9.py**:

```
1.  import requests
2.  from bs4 import BeautifulSoup
3.
4.  pagina = requests.get('http://www.uninove.br')
5.
6.  sopa = BeautifulSoup(pagina.text, 'html.parser')
7.
8.  # encontra todos os paragrafos
9.  uninove = sopa.find_all('p')
10.
11. for i in uninove:
12.     print(i.prettify())
13.
14. input('Tecle ENTER para sair...')
```

Utilizamos novamente o método **find_all**, porém com o parâmetro **('p')** para apanhar todos os parágrafos da página, pois a tag **< p >** são os parágrafos da página. E o resultado foi esse:

```
<p>
 Informação e conhecimento disponíveis o tempo todo para você!
</p>

<p>
 <a href="http://www.uninove.br/biblioteca/sobre-a-biblioteca/apresentacao/">
  Acesse.
 </a>
</p>

<p>
 Laboratórios das áreas de exatas e biológicas, com recursos modernos que dinamizam o aprendizado.
</p>

<p>
 <a href="http://www.uninove.br/conheca-a-uninove/estrutura/laboratorios/">
  Visite.
 </a>
</p>

<p>
 Primeira Universidade a oferecer wifi ilimitado para mais de 150 mil alunos.
</p>

Tecle ENTER para sair...
```

Podemos alterar o script para o seguinte código:

```
import requests
from bs4 import BeautifulSoup

pagina = requests.get("http://www.uninove.br")

sopa = BeautifulSoup(pagina.content, 'html.parser')

print(sopa.find_all('p')[2].get_text())

print(sopa.find_all('p')[4].get_text())

input('Tecle ENTER para sair...')
```

Agora utilizamos a função **find_all** com a função **get_text** para pegar somente o texto do parágrafo **('p')** com um índice **[2] e [4]** que representam a posição do parágrafo na página, lembrando que começa pelo número 0 (zero), então **[2]** representa o terceiro parágrafo e a posição **[4]** representa o quinto parágrafo. E o resultado foi este:

```
Laboratórios das áreas de exatas e biológicas, com recursos modernos que dinamizam o aprendizado.
Primeira Universidade a oferecer wifi ilimitado para mais de 150 mil alunos.
Tecle ENTER para sair...
```

Agora vamos criar outro **script** com um nome: **titulo_uni9.py**:

```
1.  import requests
2.  from bs4 import BeautifulSoup
3.
4.  pagina = requests.get('http://www.uninove.br')
5.
6.  sopa = BeautifulSoup(pagina.text, 'html.parser')
7.
8.  # encontra todos os titulos
9.  uninove = sopa.find_all('h1')
10.
11. for i in uninove:
12.     print(i.prettify())
13.
14. input('Tecle ENTER para sair...')
```

Como o parâmetro **('h1') (linha 9)** representa a tag **< h1 >** que são todos os títulos da página. E, por último, vamos ver este outro exemplo:

```
1.  import requests
2.  from bs4 import BeautifulSoup
3.
4.  pagina = requests.get('http://www.uninove.br')
5.
6.  sopa = BeautifulSoup(pagina.text, 'html.parser')
7.
8.  # encontra todos os icones
9.  uninove = sopa.find_all('i')
10.
11. for i in uninove:
12.     print(i.prettify())
13.
14. input('Tecle ENTER para sair...')
```

Neste exemplo, utilizamos o parâmetro **('i')** que é a tag **< i >** do HTML e mostra todos os ícones que existem na página da Uninove.

Estes são apenas alguns exemplos de como podemos utilizar o módulo Beautiful Soup para varrer e coletar informações de uma página da WEB.

SAIBA MAIS...

Dê uma olhada nos links abaixo para saber mais sobre a linguagem Python:

https://www.python.org/doc/ (https://www.python.org/doc/)

https://wiki.python.org/moin/PythonBooks (https://wiki.python.org/moin/PythonBooks)

Neste tópico vimos como utilizar o módulo Beautiful Soup e a técnica de WEB Scraping, criando scripts como exemplos.

Quiz

Exercício Final

Info Gather

INICIAR

Referências

SUMMERFIELD, M. *Programação em Python 3*: Uma introdução completa à linguagem Python. Rio de Janeiro Alta Books, 2012. 495 p.

MENEZES, N. N. C. *Introdução à programação com Python:* algoritmos e lógica de programação para iniciantes. 2. ed. São Paulo: Novatec, 2014. 328 p.

SWEIGART, AL. *Automatize tarefas maçantes com Python:* programação prática para verdadeiros iniciantes. São Paulo: Novatec, 2015. 568 p.

PYTHON, doc. Disponível em: <https://www.python.org/doc/>. Acesso em: Junho/2018.

PYTHON, books. Disponível em: <https://wiki.python.org/moin/PythonBooks>. Acesso em: Junho/2018.

Avalie este tópico

ANTERIOR
Packet Analyzer: criação de Packet Sniffer

Índice

PRÓXIMO
Multi-threading

Ajuda?
(https://ava.uninove.br/seu/AVA/topico/topico.php?idCurso=)

Biblioteca
(https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)
Portal Uninove
(http://www.uninove.br)
Mapa do Site