BeautifulSoup

Executando o BeautifulSoup

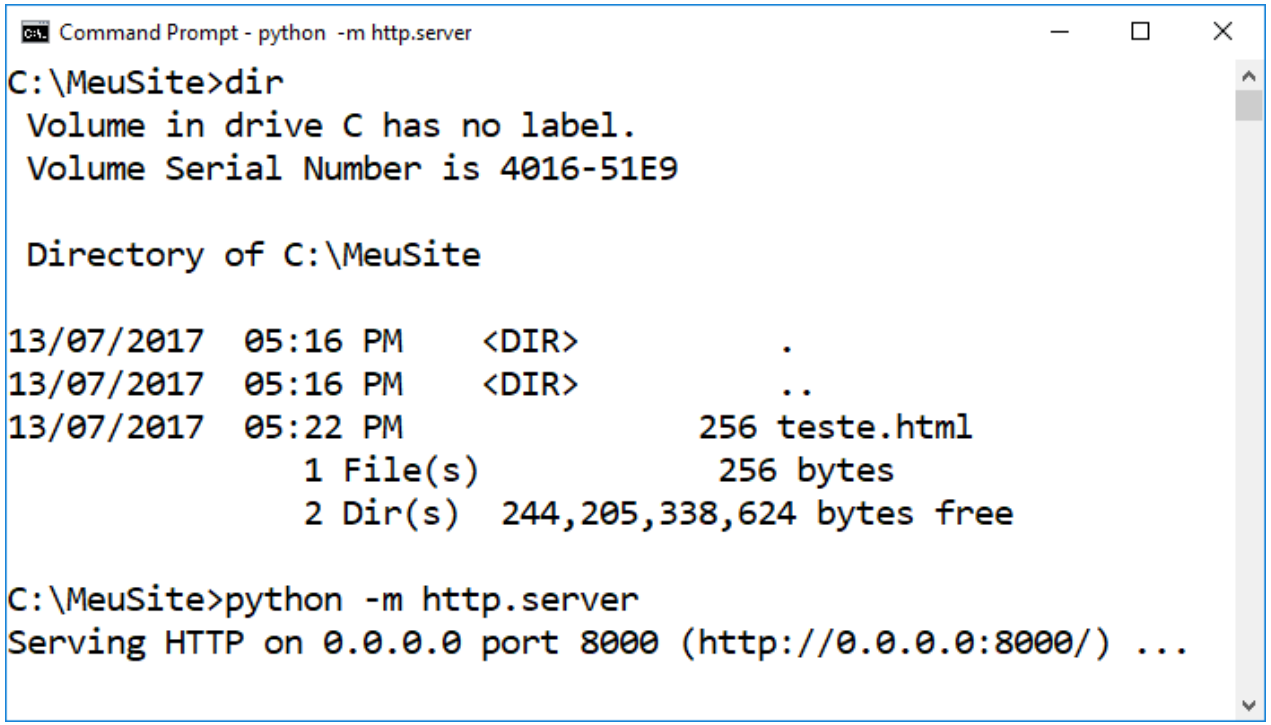
Executando o BeautifulSoup

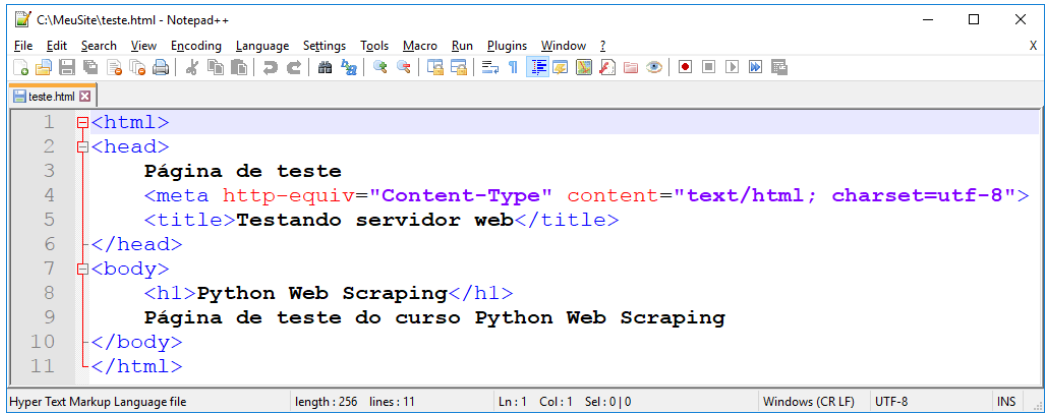
O objeto mais utilizado da biblioteca BeautifulSoup é o objeto BeautifulSoup.

Nesta aula mostrarei como utilizar este objeto para buscar informações em uma página *HTML*.

Executando o BeautifulSoup

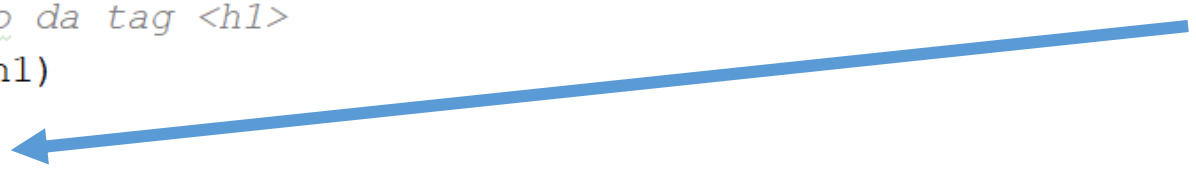
Primeiro inicie o servidor web conforme mostrado na aula anterior, contendo o arquivo *teste.html*.





Executando o BeautifulSoup

```
1  # Importando a função urlopen
2  from urllib.request import urlopen
3  # Importando a classe BeautifulSoup da biblioteca bs4
4  from bs4 import BeautifulSoup
5  # Acessando o conteúdo da página teste.html de nosso servidor web local
6  html = urlopen("http://127.0.0.1:8000/teste.html")
7  # Transformando o conteúdo HTML em um objeto BeautifulSoup
8  bsObj = BeautifulSoup(html.read(), "html.parser")
9  # Imprimindo o conteúdo da tag <h1>
10 print(bsObj.html.body.h1)
11 print(bsObj.body.h1)
12 print(bsObj.html.h1)
13 print(bsObj.h1)
```



Qualquer uma destas linhas retorna o conteúdo da tag “h1”

```
Teste
C:\Users\Evaldo\AppData\Local\Programs\Python\Python36\python.exe C:/Users/E
<h1>Python Web Scraping</h1>
<h1>Python Web Scraping</h1>
<h1>Python Web Scraping</h1>
<h1>Python Web Scraping</h1>
```

Executando o BeautifulSoup

Usando BeautifulSoup você pode, praticamente, extrair qualquer informação de arquivos *HTML* ou *XML*, contanto que esta informação tenha, ou esteja próxima a uma *tag* identificadora.

Usando objeto.h1, retornamos somente a primeira *tag* encontrada no documento. Se existirem mais *tags* h1?

Vamos mudar nosso *teste.hml* acrescentando mais uma *tag* h1.

Executando o BeautifulSoup

Mude o arquivo acrescentando mais uma *tag h1*.

```
<html>
<head>
        Página de teste
        <meta http-equiv="Content-Type"
content="text/html; charset=utf-8">
        <title>Testando servidor web</title>
</head>
<body>
        <h1>Python Web Scraping</h1>
        Página de teste do curso Python Web Scraping
        <h1>Aula 2</h1>
        Aprendendo a usar BeautifulSoup
</body>
</html>
```

Ao executar o programa novamente, teremos apenas o resultado abaixo:
<h1>Python Web Scraping</h1>

Será exibida a primeira ocorrência de h1.

Executando o BeautifulSoup

Para retornar uma lista com todas ocorrências das *tags* h1 podemos usar o método *find_all()*.

```
1  # Importando a função urlopen
2  from urllib.request import urlopen
3  # Importando a classe BeautifulSoup da biblioteca bs4
4  from bs4 import BeautifulSoup
5  # Acessando o conteúdo da página teste.html de nosso servidor
6  html = urlopen("http://127.0.0.1:8000/teste.html")
7  # Transformando o conteúdo HTML em um objeto BeautifulSoup
8  bsObj = BeautifulSoup(html.read(), "html.parser")
9  # Imprimindo o conteúdo da tag <h1>
10 print(bsObj.find_all('h1'))
```

```
Unittests in teste3.py
No tests were
[<h1>Python Web Scraping</h1>, <h1>Aula 2</h1>]
```

Executando o BeautifulSoup

Retornando todos os links de uma página usando o BeatifulSoup.

Um link em uma página *HTML* é identificado pela *tag*

Texto a ser exibido

Vamos colocar alguns links em nossa página *teste.html*.

Executando o BeautifulSoup

```
<html>
<head>
    Página de teste
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Testando servidor web</title>
</head>
<body>
    <h1>Python Web Scraping</h1>
    Página de teste do curso Python Web Scraping
    <h1>Aula 2</h1>
    Aprendendo a usar BeautifulSoup<br><br>


    Acesse os links abaixo:<br>
    <a href="http://www.google.com">Google</a><br>
    <a href="https://www.python.org">Python</a><br>
    <a href="http://globoesporte.globo.com/">Globo Esporte</a><br>
    <a href="http://evaldowolkers.wordpress.com">Página do Evaldo</a>
</body>
</html>
```

Executando o BeautifulSoup

Uma forma de “pegar” os links existentes na página é usando o *find_all*. Abaixo foi feito um *loop* no resultado do *find_all* para imprimir todas as *tags href*.

```
for link in bsObj.find_all('a'):
    print(link)
```

Veja o resultado a seguir.

Executando o BeautifulSoup

```
1   # Importando a função urlopen
2   from urllib.request import urlopen
3   # Importando a classe BeautifulSoup da biblioteca bs4
4   from bs4 import BeautifulSoup
5   # Acessando o conteúdo da página teste.html de nosso servidor web local
6   html = urlopen("http://127.0.0.1:8000/teste.html")
7   # Transformando o conteúdo HTML em um objeto BeautifulSoup
8   bsObj = BeautifulSoup(html.read(), "html.parser")
9   # Encontrando os links existentes na página
10  for link in bsObj.find_all('a'):
11      print(link)
```

```
Teste2
C:\Users\Evaldo\AppData\Local\Programs\Python\Python36\python.exe C:/Users/E
<a href="http://www.google.com">Google</a>
<a href="https://www.python.org">Python</a>
<a href="http://globoesporte.globo.com/">Globo Esporte</a>
<a href="http://evaldowolkers.wordpress.com">Página do Evaldo</a>
```

Executando o BeautifulSoup

Veja agora como fazer para pegar somente os links, sem o conteúdo inteiro da *tag* “<a>”.

Como as *tags* “<a>” possuem sempre o *href=“link”* informando o endereço. Veja abaixo:

```
<a href="http://www.google.com">Google</a>
<a href="https://www.python.org">Python</a>
<a href="http://globoesporte.globo.com/">Globo Esporte</a>
<a href="http://evaldowolkers.wordpress.com">Página do Evaldo</a>
```

Podemos usar o método *get* para pegar o conteúdo de *“href=”*.

Executando o BeautifulSoup

```
1   # Importando a função urlopen
2   from urllib.request import urlopen
3   # Importando a classe BeautifulSoup da biblioteca bs4
4   from bs4 import BeautifulSoup
5   # Acessando o conteúdo da página teste.html de nosso servidor web local
6   html = urlopen("http://127.0.0.1:8000/teste.html")
7   # Transformando o conteúdo HTML em um objeto BeautifulSoup
8   bsObj = BeautifulSoup(html.read(), "html.parser")
9   # Pegando somente os links
10  for link in bsObj.find_all('a'):
11      print(link.get('href'))
```

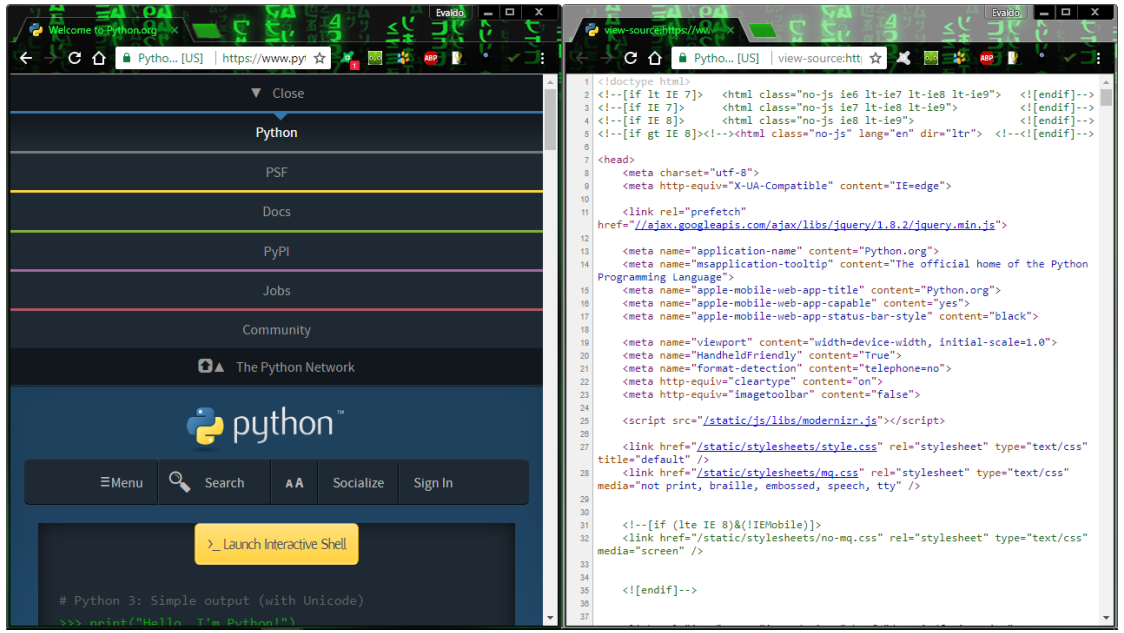
Teste2

```
C:\Users\Evaldo\AppData\Local\Programs\Python\Python36\python.exe C:/Users/E
http://www.google.com
https://www.python.org
http://globoesporte.globo.com/
http://evaldowolkers.wordpress.com

Process finished with exit code 0
```

Executando o BeautifulSoup

Teste o programa agora pegando todos os links do site *https://www.python.org* e em seguida teste com outros sites, você vai obter vários links existentes nos sites.



FIM