

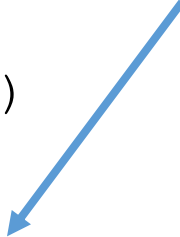
# BeautifulSoup

Um pouco mais de BeautifulSoup

# Um pouco mais de BeautifulSoup

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
```


*#Localizar qualquer tag que tenha a propriedade class igual a comments-link*



```
html = urlopen("https://evaldowolkers.wordpress.com")
bsobj = BeautifulSoup(html, "html.parser")
```

```
teste = bsobj.findAll("", {"class": "comments-link"})
for a in teste:
    print(a)
```

```
python = bsobj.findAll(text="Python")
for a in python:
    print(a)
```



*#Localizar o texto Python*

# Um pouco mais de BeautifulSoup

## Veja o resultado

`<span class="comments-link"><a href="https://evaldowolkers.wordpress.com/2017/03/10/curso-de-python-aula-9/#respond">Deixe um comentário</a></span>`

`<span class="comments-link"><a href="https://evaldowolkers.wordpress.com/2017/02/10/curso-de-python-aula-8/#respond">Deixe um comentário</a></span>`

`<span class="comments-link"><a href="https://evaldowolkers.wordpress.com/2017/01/17/curso-de-python-aula-7/#respond">Deixe um comentário</a></span>`

`<span class="comments-link"><a href="https://evaldowolkers.wordpress.com/2017/01/10/curso-de-python-aula-6/#respond">Deixe um comentário</a></span>`

`<span class="comments-link"><a href="https://evaldowolkers.wordpress.com/2017/01/06/curso-de-python-aula-5/#respond">Deixe um comentário</a></span>`

`<span class="comments-link"><a href="https://evaldowolkers.wordpress.com/2016/12/30/curso-de-python-aula-4/#respond">Deixe um comentário</a></span>`

`<span class="comments-link"><a href="https://evaldowolkers.wordpress.com/2016/12/27/curso-de-python-aula-3/#respond">Deixe um comentário</a></span>`

Python

Python

Python

Python

Python

# Um pouco mais de BeautifulSoup

## Conferindo as informações no site

```
youtube.com/evaldowolkers</p>
</div>
<footer class="entry-footer">
  <span class="comments-link"><a href="https://evaldowolkers.wordpress.com/2017/02/10/curso-
</article><!-- #post-## -->
<article id="post-114" class="post-114 post type-post status-publish format-standard hentry catego:
```

Veja uma das tags span retornadas.

```
ategorized">
```

Veja uma das ocorrências da palavra Python.

```
okmark">Curso de Python &#8211; Aula&nbsp;7</a></h2>
<div c:
="bookmark"><time class="entry-date published" datetime="2017-01-17T10:19:!!

'youtube-player' type='text/html' width='685' height='416' src='https://ww
```

# Um pouco mais de BeautifulSoup

## Explorando as tags *HTML* com BeautifulSoup

```
# .get_text() - Retorna todo o texto da página  
print(bsObj.get_text())
```

```
# .tag - Retorna a primeira ocorrência da tag informada  
print(bsObj.title)
```

```
# .tag.name - Retorna o nome da primeira ocorrência da tag informada  
print(bsObj.title.name)
```

```
# .tag.string - Retorna o texto da primeira ocorrência da tag informada  
print(bsObj.title.string)
```

# Um pouco mais de BeautifulSoup

## Explorando as tags *HTML* com BeautifulSoup

```
# .tag.parent - Retorna a tag externa à tag atual (tag pai/mãe)  
print(bsObj.title.parent)
```

```
# .tag.parent.name - Retorna o nome da tag externa à tag atual (tag pai/mãe)  
print(bsObj.title.parent.name)
```

```
# .tag['atributo'] - retorna todos os valores do atributo informado  
print(bsObj.body['class'])  
print(bsObj.button['aria-controls'])
```

```
# .find(id="descricao") - retorna a tag que possua o id informado  
print(bsObj.find(id="menu-item-147"))
```

# Um pouco mais de BeautifulSoup

Vamos fazer mais um exercício:

1 – Crie um arquivo com o conteúdo abaixo e salve em uma pasta qualquer como “site.html”.

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Página HTML de exemplo</title>
  </head>
  <body>
    <p class="title"><b>Página de teste</b></p>
    <p class="nome_classe">Olá, este é um teste do uso de BeautifulSoup<br>
    <a href="http://www.google.com" class="google" id="link1">Google</a><br>
    <a href="http://ubuntu.com" class="ubuntu" id="link2">Ubuntu</a><br>
    <a href="http://python.org" class="python" id="link3">Python</a><br>
  </body>
</html>
```

# Um pouco mais de BeautifulSoup

2 – Acesse a pasta onde se encontra o arquivo via linha de comando (prompt de comando ou Terminal) e inicie o servidor web com Python.



```
Command Prompt - python -m http.server

C:\Temp>python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
-
```

3 – Execute o arquivo `beautiful_aula5.py` e veja o resultado.



# Um pouco mais de BeautifulSoup

Vamos implementar um programa para buscar links em que conste o nome de um dos seis primeiros times da tabela do brasileiro 2017. Tabela de 22/08/2017. Iniciando no site [globoesporte.globo.com](http://globoesporte.globo.com).

```
def getLinks(url_da_pagina):  
    global paginas  
    try:  
        if url_da_pagina not in paginas_invalidas:  
            html = urlopen(url_da_pagina)  
            bsObj = BeautifulSoup(html, "html.parser")  
            serie_a_g6_2017 = ('.corinthians.|.gremio.|.santos.|.palmeiras.|.flamengo.|.cruzeiro.')  
  
            for link in bsObj.findAll("a", href=re.compile(serie_a_g6_2017)):  
                if "href" in link.attrs:  
                    if link.attrs['href'] not in paginas and link.attrs['href'] not in paginas_invalidas:  
                        nova_pagina = link.attrs['href']  
                        print(nova_pagina)  
                        paginas.add(nova_pagina)  
                        getLinks(nova_pagina)  
  
    except:  
        paginas_invalidas.add(nova_pagina)  
  
getLinks("http://globoesporte.globo.com")
```

```
from urllib.request import urlopen  
from bs4 import BeautifulSoup  
import re
```

```
paginas = set()  
paginas_invalidas = set()  
nova_pagina = ""
```

# FIM