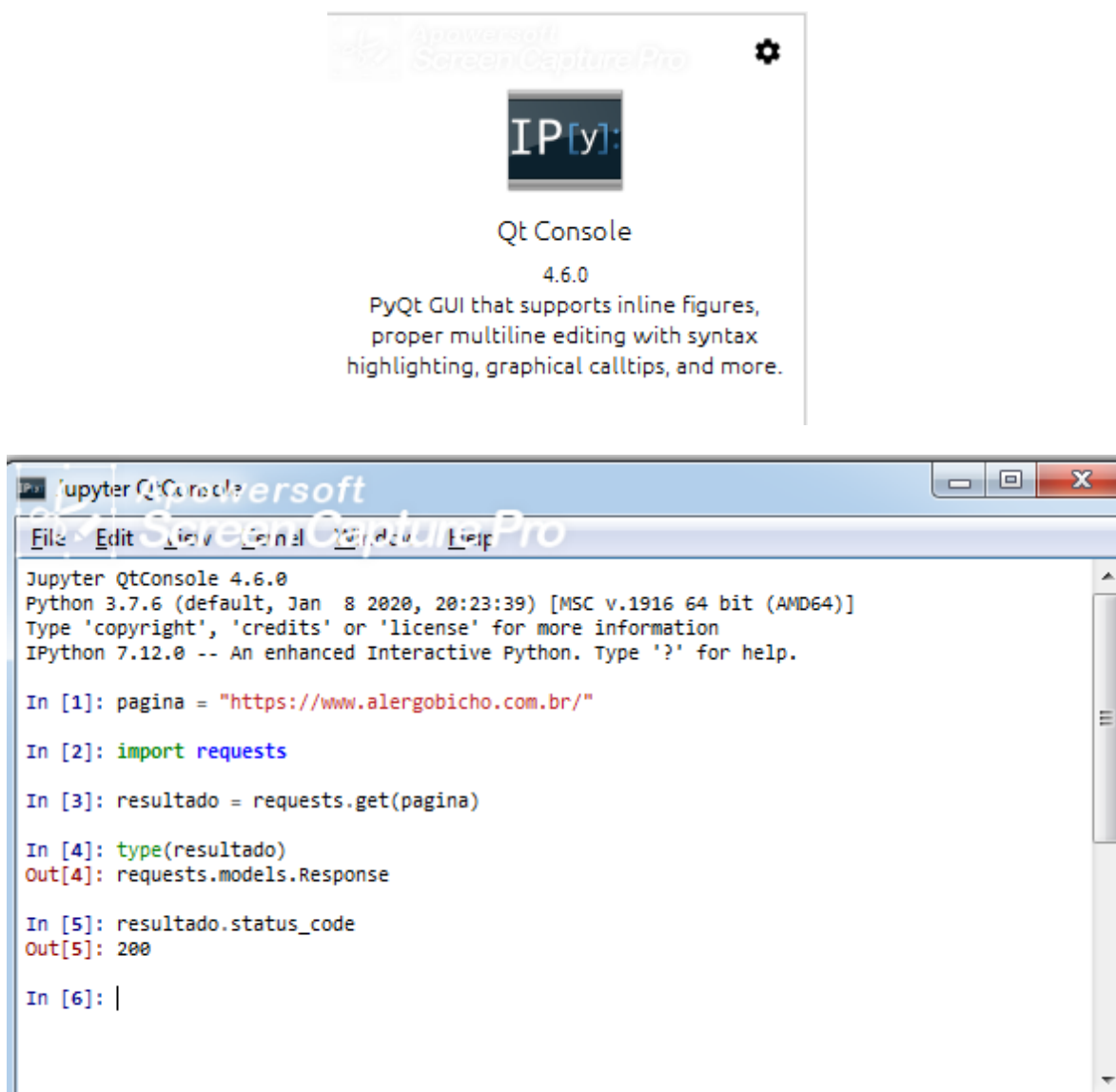


# WEB SCRAPING

Passo a Passo de como é realizado o WEBSCRAPING. Utilizei o Anaconda esse aplicativo para a realização dessa atividade



# Verifico se está ok para a utilização

Jupyter QtConsole 4.6.0

Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]

Type 'copyright', 'credits' or 'license' for more information

IPython 7.12.0 -- An enhanced Interactive Python. Type '?' for help.

# Defino a página que eu desejo analisar. Nesse caso eu optei pela minha página. Lembrando que tudo o que eu faço fica armazenado no servidor do cliente. Utilizei para esse exercício o site da alergobicho.

pagina = " https://www.alergobicho.com.br/ "

#Pacotes Python que permitem manipular paginas web

```
import requests
```

# Defino a página que eu desejo analisar. Get ( método). Nesse momento eu conecto a página.  
Estando ok eu recebo uma mensagem

```
resultado = requests.get(pagina)
```

```
type(resultado)
```

```
Out[4]: requests.models.Response
```

# código 200 indica que a conexão foi feita com sucesso.

# código 404 indica que a conexão não foi feita com sucesso.Verificamos se a pagina é válida

```
resultado.status_code
```

```
Out[5]: 200
```

# retorna os métodos e atributos do resultado, que podemos manipular o objeto.

resultado. (apertar a tecla TABLE)

```
In [4]: resultado
```

apparent_encoding	history	ok
close()	is_permanent_redirect	raise_for_status()
connection	is_redirect	raw
content	iter_content()	reason
cookies	iter_lines()	request
elapsed	json()	status_code
encoding	links	text
headers	next	url

# o tipo de resultado que ele retorna é bytes. Para poder manipular todo o conteúdo como HTML devemos transformar em texto, logo uma string ..“str”.

```
type(resultado.content)
```

```
Out[7]: bytes
```

```
fonte = resultado.text
```

```
type(fonte)
```

```
Out[9]: str
```

**# o pacote Request do Python serve apenas para obter os dados.**

2- Temos que saber o que é código HTML dentro da nossa página. Para isso utilizamos o “Parse HTML” pacote BeautifulSoup.

```
from bs4 import BeautifulSoup
```

```
soup = BeautifulSoup(fonte, 'html.parser')
```

#Para ver se o resultado se deu certo. Ele mostra todo o código.

```
print(soup.prettify)
```

Mostra toda a pagina em HTML a partir desse comando acima.

```
        </strong>
</div>
</div>
</div>
</div>
<div class="acoes-produto hidden-phone">
<a class="botao botao-comprar principal" href="https://www.alergobicho.com.br/none-58022379" title="Ver
detalhes do produto">
<i class="icon-search"></i>Comprar
    </a>
</div>
<div class="acoes-produto-responsiva visible-phone">
<a class="tag-comprar fundo-principal" href="https://www.alergobicho.com.br/none-58022379" title="Ver
detalhes do produto">
```

## Exemplo do código do Professor

```
Type 'copyright', 'credits' or 'license' for more information
IPython 7.4.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: pagina = "http://lscallhos.com/none-58022379.html"

In [2]: import requests

In [3]: resultado = requests.get(pagina)

In [4]: resultado.content[:15]
Out[4]: b'<!DOCTYPE html>'

In [5]: type(resultado)
Out[5]: requests.models.Response

In [6]: type(resultado.content)
Out[6]: bytes

In [7]: fonte = resultado.text

In [8]: type(fonte)
Out[8]: str

In [9]: from bs4 import BeautifulSoup

In [10]: soup = BeautifulSoup(fonte, 'html.parser')

In [11]: soup.prettify()
```

---

Para poder executar o webscrapping se faz necessário o entendimento mínimo de HTM e CSS (código de marcação). Um bom site para conhecimento é W3schools.com

Como localizar o código fonte de um site. Com o mouse direito dentro do site (Inspecionar o código fonte ou ver o código fonte da página)

### Contruindo um webscrapper

Para esse exercício, iremos utilizar o modelo do professor. Extrair a (pasta 3 –Parte2)

Como transformar o meu computador em local host para a realização do exercício.

Na minha máquina executar o cmd (como administrador da máquina).

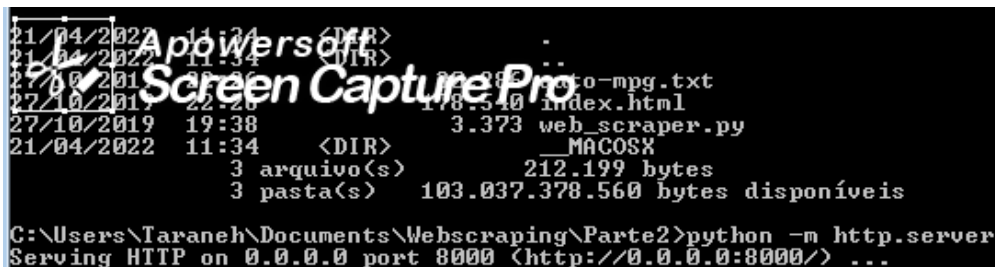
**Comandos do cmd : cd ../ ( volta um nível pasta)**

No cmd da minha máquina eu digito onde está o documento que é esse caminho que está abaixo: C:\Users\Taraneh\Documents\Webscraping\Parte2



```
Pasta de C:\Users\Taraneh\Documents\Webscraping\Parte2
21/04/2022 11:34 <DIR>
21/04/2022 11:34 <DIR>
27/10/2019 22:26      30.286 auto-mpg.txt
27/10/2019 22:26     178.540 index.html
27/10/2019 19:38      3.373 web_scraper.py
21/04/2022 11:34 <DIR>                _MACOSX
                3 arquivo(s)        212.199 bytes
                3 pasta(s)    103.037.378.560 bytes disponíveis
C:\Users\Taraneh\Documents\Webscraping\Parte2>
```

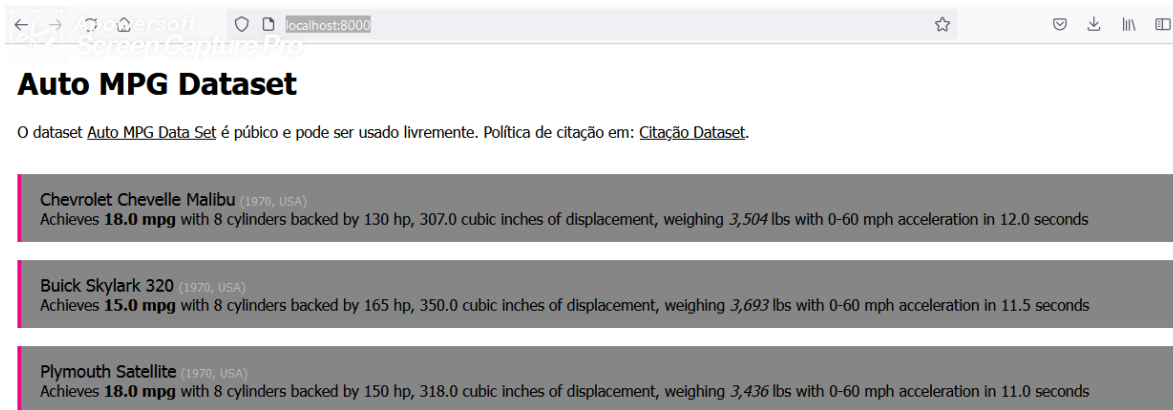
Quando eu chegar nesse nível de pasta eu terei que inicializar o servidor web pelo computador dentro dessa pasta (cmd) utilizando o seguinte caminho: **python -m http.server**



```
21/04/2022 11:34 <DIR>
21/04/2022 11:34 <DIR>
27/10/2019 22:26      30.286 auto-mpg.txt
27/10/2019 22:26     178.540 index.html
27/10/2019 19:38      3.373 web_scraper.py
21/04/2022 11:34 <DIR>                _MACOSX
                3 arquivo(s)        212.199 bytes
                3 pasta(s)    103.037.378.560 bytes disponíveis
C:\Users\Taraneh\Documents\Webscraping\Parte2>python -m http.server
Servicing HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Isso mostra que o servidor foi inicializado. Agora é só abrir a pasta e digitar:

<http://localhost:8000/>



Isso mostra que o passo foi executado com sucesso.

Agora vamos abrir uma nova janela CMD (abrir como adm) e chegar até onde está a pasta index.

Digitar : **python web\_scraper.py**

```
21/04/2022 11:34 <DIR> .
21/04/2022 11:34 <DIR> ..
27/10/2019 22:26 30.280 auto-mpg.txt
27/10/2019 18:50 8.610 index.html
27/10/2019 19:38 3.373 web_scraper.py
21/04/2022 11:34 <DIR> _MACOSX
3 arquivo(s) 212.199 bytes
3 pasta(s) 103.035.211.776 bytes disponíveis

C:\Users\Taraneh\Documents\Webscraping\Parte2>python web_scraper.py
Copiando dados da página http://localhost:8000/index.html.
Gravando o cache em dados_copiados_v1.pickle
Temos 406 linhas de dados retornadas do scraping da página!
Primeira linha copiada:
{'name': 'Chevrolet Chevelle Malibu', 'cylinders': 8, 'weight': 3504, 'acceleration': 12.0}
última linha copiada:
{'name': 'Chevy S-10', 'cylinders': 4, 'weight': 2720, 'acceleration': 19.4}

C:\Users\Taraneh\Documents\Webscraping\Parte2>
```

Isso mostra o resultado da etapa executada com sucesso.

1-) A cópia dessa página da internet gerou um arquivo.csv.

2-) A quantidade de linhas que foram copiadas.(406)

3-) A primeira e última linha também copiadas.

Gera um dicionário {} com chave e valor.

Resultando em um arquivo chamado **dados\_copiados\_v1** na mesma pasta (PARTE2)

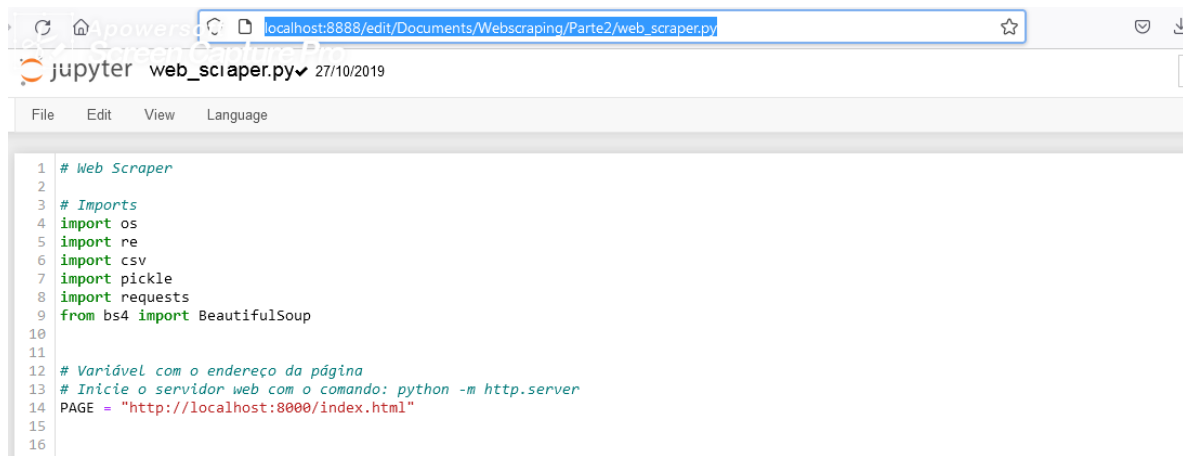
O arquivo

📁 _MACOSX	21/04/2021
📄 auto-mpg	27/10/2019
📄 dados_copiados_v1	21/04/2021
📄 dados_copiados_v1.pickle	21/04/2021
📄 index	27/10/2019
📄 web_scraper	27/10/2019

O arquivo **web\_scraper.py** mostra como foi feito esse script para a cópia desse documento.

Vamos abrir agora um Jupyter Notebook e abrir o arquivo web\_scraper.py

[http://localhost:8888/edit/Documents/Webscraping/Parte2/web\\_scraper.py](http://localhost:8888/edit/Documents/Webscraping/Parte2/web_scraper.py)



```

1 # Web Scraper
2
3 # Imports
4 import os
5 import re
6 import csv
7 import pickle
8 import requests
9 from bs4 import BeautifulSoup
10
11
12 # Variável com o endereço da página
13 # Inicie o servidor web com o comando: python -m http.server
14 PAGE = "http://localhost:8000/index.html"
15
16

```