



Curso de Web Scraping com Python

Rafael Alves Ribeiro

21 de Julho de 2018

Configurando o Ambiente de Trabalho

Instalando o Python

Web Scraping

O que é Web Scraping

O que é um Scraper

Requests

HTTP for Humans

Configurando o Ambiente

Instalando o Python

Utilizaremos a versão 3.6 do Python:

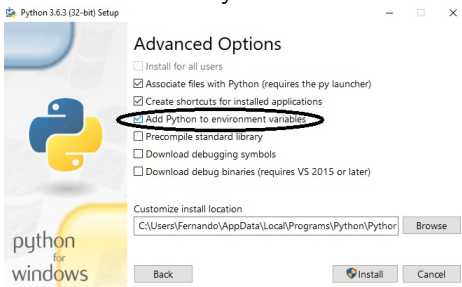
<https://www.python.org/downloads/release/python-366/>

Configurando o Ambiente

Instalando o Python - Windows



Adicionar o Python ao PATH:



Configurando o Ambiente

Instalando o Python - Linux



Python3 já está deve estar instalado. Instalar o idle:

```
sudo apt-get install idle3
```

Abrir o terminal e executar o comando: *python3*

Configurando o Ambiente

Instalando o Python - Mac



MacOS

Abrir o terminal e executar o comando:

brew install python3

Em caso de dúvidas:

<https://docs.python-guide.org/starting/install3/osx/>

Configurando o Ambiente

Instalando as Bibliotecas

Na linha de comando (cmd) executar o comando abaixo:

```
pip install requests beautifulsoup4 pdfminer3k selenium ipython  
jupyter pandas matplotlib lxml xlrd PyPDF2 jupyter Pillow  
pytesseract
```

ou

```
pip install -r requirements.txt
```

*Linux e Mac: utilizar pip3

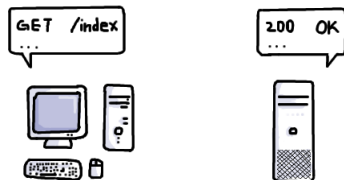
Web Scraping, ou raspagem de dados, consiste em um processo que se utiliza de técnicas de programação para a coleta automatizada de dados provenientes da Web.

Web Scraping

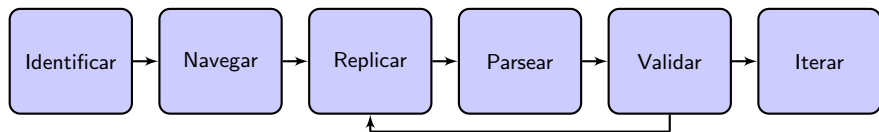
O que é um Scraper

Scraper é um software que simula a interação realizada entre um browser operado por um humano e um Web Site. Possui 3 funções básicas:

- ▶ **Acesso ao Web Site**
- ▶ **Parsing e Extração de conteúdo**
- ▶ **Estruturação dos resultados**



6 Etapas para o desenvolvimento de um Web Scraper



Web Scrapping

Desenvolvendo um Scraper - Identificar

No primeiro passo do processo de desenvolvimento de um scraper precisamos **entender qual é a estrutura das páginas que queremos raspar** e traçar um plano para extrair tudo que precisamos.

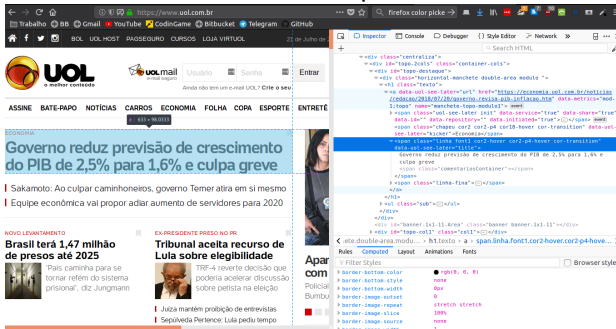
The image displays three side-by-side screenshots of Wikipedia pages, illustrating the structure of web pages for scraping. Each page has a consistent layout with a sidebar on the left, a main content area, and a navigation bar at the top.

- Left Screenshot (R):** The page title is "R (programming language)". The main content area contains a large blue "R" logo and text describing R as a programming language and free software environment for statistical computing and graphics. The sidebar on the left includes a "Main page" link and a "Contents" table of contents.
- Middle Screenshot (Python):** The page title is "Python". The main content area contains a large blue "Python" logo and text describing Python as a high-level, interpreted, object-oriented, and general-purpose programming language. The sidebar on the left includes a "Main page" link and a "Contents" table of contents.
- Right Screenshot (JavaScript):** The page title is "JavaScript". The main content area contains a large blue "JavaScript" logo and text describing JavaScript as a high-level, interpreted, object-oriented, and general-purpose programming language. The sidebar on the left includes a "Main page" link and a "Contents" table of contents.

Web Scraping

Desenvolvendo um Scraper - Navegar

Precisamos entender **como localizar o dado que queremos extrair dentro do HTML da página**. Esse passo pode ser extremamente simples, mas de vez em quando ele se tornará algo bastante complexo.



O Dev Tools é conjunto de ferramentas integradas ao browser construídas para facilitar o desenvolvimento de Web Sites. Permite analisar o código, o tráfego de rede e a performance de uma página. É a principal ferramenta de apoio ao desenvolvimento de Scrapers.

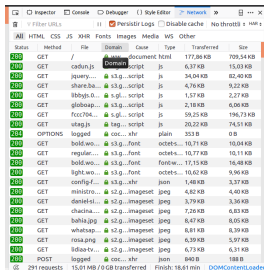
Como acessar:

Abra o Firefox ou o Google Chrome e tecle F12.

Web Scraping

Desenvolvendo um Scraper - Replicar

Neste passo é importante compreender as várias requisições HTTP que a página está realizando para trazer o conteúdo até você, assim poderemos replicar as requisições com nosso Scraper. Utilizaremos a **aba Network do Dev Tools** para este trabalho.



Status	Method	File	Domain	Cause	Type	Transferred	Size
200	GET	/	www.document.html		177.86 KB	709.54 KB	
200	GET	cadun.js	Domain	script	js	6.37 KB	15.03 KB
200	GET	jquery...	s3.g...script	js	34.04 KB	82.40 KB	
200	GET	share.ba...	s3.g...script	js	4.76 KB	9.22 KB	
200	GET	libbys.0...	s3.g...script	js	1.57 KB	2.27 KB	
200	GET	gleboap...	s3.g...script	js	2.18 KB	6.06 KB	
200	GET	fccc04...	s3.g...script	js	59.25 KB	196.73 KB	
200	GET	vlog.js	img...script	js	20.22 KB	74.51 KB	
200	OPTIONS	logged	coc...xhr	plain	353 B	0 B	
200	GET	bold.wo...	s3.g...font	octets...	10.71 KB	10.04 KB	
200	GET	regular...	s3.g...font	octets...	10.77 KB	10.11 KB	
200	GET	bold.wo...	s3.g...font	fontw...	17.15 KB	16.48 KB	
200	GET	light.wo...	s3.g...font	octets...	10.62 KB	9.96 KB	
200	GET	config-f...	s3.g...xhr	json	1.48 KB	5.37 KB	
200	GET	minisro...	s2.g...image	jpeg	4.82 KB	4.40 KB	
200	GET	daniel-s...	s2.g...image	jpeg	3.79 KB	3.36 KB	
200	GET	chacina...	s2.g...image	jpeg	7.26 KB	6.83 KB	
200	GET	bahia.jpg	s2.g...image	jpeg	8.47 KB	8.05 KB	
200	GET	whatsapp...	s2.g...image	jpeg	8.81 KB	8.39 KB	
200	GET	rosa.png	s2.g...image	jpeg	6.39 KB	5.97 KB	
200	GET	idbaotv...	s2.g...image	jpeg	6.73 KB	6.31 KB	
200	POST	logged	coc...xhr	json	840 B	188 B	


291 requests 15.61 MB / 0.68 GB transferred Finish: 18.61 min DOMContentLoaded

<http://testing-ground.scraping.pro/login>

Web Scraping

Desenvolvendo um Scraper - Replicar

Qual a classe da imagem da logomarca da Inferir?
Qual o tipo da requisição HTTP?

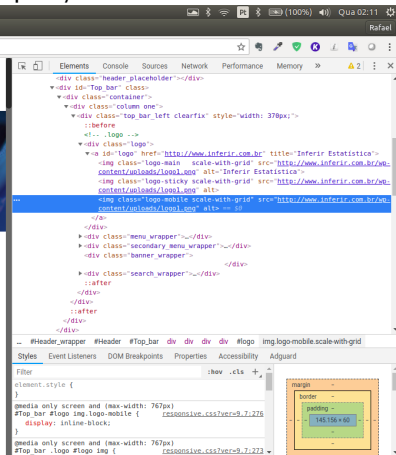


The screenshot shows the Inferir Estatística website. The logo is a blue square with the word "inferir" in white lowercase letters and a green line graph icon. Below the logo is a banner for a course titled "CURSO WEB SCRAPING" with dates "21 e 28 de julho - das 9h às 18h". A yellow button on the banner says "Saiba como extrair dados de sites de forma automatizada".

Below the banner is the word "Empresa" with a green line graph icon above it.

A Inferir® Estatística iniciou sua atividade em 2013, aproveitando a experiência de seus fundadores, Fioravante e Diogo que, desde 2002, trabalham com estatística em diversas empresas nas áreas financeiras, de saúde ou de educação.

A Inferir® Estatística é uma empresa que atua nos segmentos de consultoria, treinamento e assessoria em



The screenshot shows the Chrome DevTools Elements panel. The HTML structure is as follows:

```
<div class="header_placeholder"></div>
<div id="Top_bar" class="...>
  <div class="container">
    <div class="column one">
      <div class="top_bar_left clearfix" style="width: 378px;">
        <div class="logo">
          <a id="logo" href="http://www.inferir.com.br" title="Inferir Estatística">
            
            
            
          </a>
        </div>
        <div class="menu_wrapper"></div>
        <div class="secondary_menu_wrapper"></div>
        <div class="banner_wrapper"></div>
        <div class="search_wrapper"></div>
      </div>
    </div>
  </div>
</div>
```

The selected element is the logo image with the class "img logo-mobile scale-with-grid" and the attribute "src="http://www.inferir.com.br/wp-content/uploads/logo.png".

Web Scraping

Desenvolvendo um Scraper - Parsear

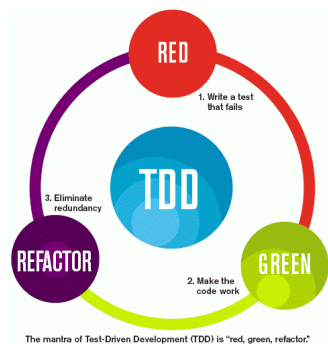
O anglicismo parsear vem do verbo *to parse*, que quer dizer algo como analisar ou estudar, mas que, no contexto do Web Scraping, significa extrair os dados desejados de um arquivo HTML.

```
1 from bs4 import BeautifulSoup
2 import requests
3
4 # enviamos uma requisição HTTP utilizando o método GET
5 response = requests.get('http://www.inferir.com.br')
6 html = response.text
7
8 # realizamos o parse do HTML utilizando BeautifulSoup
9 soup = BeautifulSoup(html, 'html.parser')
10
11 # buscamos a imagem utilizando a tag e a classe
12 # encontradas na Ferramenta de Desenvolvedor
13 classe_logo = 'logo-mobile scale-with-grid'
14 logo_inferir = soup.find('img', {'class': classe_logo})
15 print(logo_inferir)
```

Web Scrapping

Desenvolvendo um Scraper - Validar

Se tivermos feito tudo certo até agora, validar os resultados será uma tarefa simples. Precisamos apenas reproduzir o procedimento descrito até agora para algumas outras páginas de modo verificar se estamos de fato extraíndo corretamente tudo o que queremos.



Web Scrapping

Desenvolvendo um Scraper - Iterar

O último passo consiste em colocar o nosso scraper em produção. Aqui, ele já deve estar funcionando corretamente para todos os casos desejados.

Na maior parte dos casos isso consiste em encapsular o scraper em uma função que recebe uma série de links e aplica o mesmo procedimento em cada um. Se quisermos aumentar a eficiência desse processo, podemos paralelizar ou distribuir o nosso raspador.



Requests

HTTP for Humans



Requests
http for humans

Requests é um pacote Python que permite o acesso à serviços web, sem a necessidade de conhecimento avançados sobre o protocolo de comunicação HTTP.

Empresas como Amazon, Google, Mozilla, PayPal, The Washington Post e Twitter utilizam Requests, que é **um dos pacotes Python mais baixados de todos os tempos, com mais de 11 milhões de downloads mensais**.

Para importar o pacote utilize o seguinte código:

```
1 import requests
```

Requests

Baixando um arquivo com 4 linhas de código

No exemplo abaixo, fazemos o download das informações da Base de Dados do Comércio Exterior Brasileiro disponibilizadas no site do Ministério da Indústria, Comércio Exterior e Serviços e imprimimos na tela a primeira linha do arquivo csv.

```
1     >>> import requests
2     >>>
3     >>> url = "http://www.mdic.gov.br/balanca/bd/ncm/EXP_2018.csv"
4     >>> # envia uma requisição HTTP utilizando o método GET
5     >>> response = requests.get(url)
6     >>> # acessa o texto da resposta enviada pelo servidor,
7     >>> # seleciona a primeira linha e imprime na tela
8     >>> print(response.text.splitlines()[0])
9     "CO\_UNID";"CO\_PAIS";"CO\_UF";"CO\_PORTO";"CO\_VIA";"QT\_ESTAT";
10    "KG\_LIQUIDO";"VL\_FOB"''
```

Modifique o código anterior, selecionando um arquivo .csv encontrado em uma busca no Google. Imprima todo o conteúdo do arquivo na tela.

Requests

Manipulando a resposta do servidor

Após receber e interpretar a requisição enviada, o servidor web retorna uma resposta HTTP contendo diversas informações úteis. O Requests realiza o tratamento destes dados e os armazena em um objeto chamado **Response**.

Em nosso exemplo, utilizamos apenas o texto da resposta. Realizaremos uma nova requisição, desta vez solicitando uma imagem, para explorar as informações contidas no objeto Response.

Requests

Manipulando a resposta do servidor



ex02_requests2.py

Bônus: from PIL import Image

Manipulando a resposta do servidor - Exercício

#ficadica
<https://wiki.python.org/moin/UsingAssertionsEffectively>
<https://www.freeformatter.com/mime-types-list.html>

Requests

Primeira Função Python - ex04_requests3.py

Agora que conseguimos analisar as repostas do servidor e salvar arquivos, vamos escrever nossa primeira função Python para consolidar o conhecimento.

https://docs.python.org/3/reference/compound_stmts.html#grammar-token-funcdef

Requests

Estudo de Caso - CVM - ex05_requests4.py



http://dados.cvm.gov.br/dataset/fi-doc-inf_diario

Requests

Estudo de Caso - Mais Bolão - ex07_requests6.py



<http://www.maisbolao.com.br/>

Requests

Desafio - Web Scraper Testing Ground



`http://testing-ground.scraping.pro/login`

Requests

Consumindo API's - ex08_requests7.py

O consumo de API's não faz parte do conceito de Web Scraping, uma vez que os dados estão estruturados.



BANCO CENTRAL DO BRASIL

<https://olinda.bcb.gov.br/olinda/servico/PTAX/versao/v1/swagger-ui3#/>