

Scrapy

primeiros passos



Agenda

- Conceitos de Web Scraping
- Instalando o Scrapy
- Arquitetura de um Spider
- Extraindo dados
- Exercícios

Olá!

- Python Developer na **Scrapinghub**
- Laboratório Hacker de Campinas
- Grupy-Campinas
- **renne@rennerocha.com**
- **@rennerocha**

Telegram / Twitter / Github / Instagram / ...



We are Scrapinghub - We embrace work, challenges, and appreciate diversity, we welcome innovation, we are an eclectic team who delivers and have fun.

Scrapinghub is hiring!

scrapinghub.com/jobs



O que é Web Scraping?

Extrair dados **estruturados** de
fontes de dados **não estruturadas**
(tipicamente páginas web)

O que é Web Scraping?

Casos de uso (alguns):

- Pesquisas com dados governamentais
- Monitorar o que estão falando do meu produto
- Monitorar os produtos dos concorrentes
- Ofertas de emprego, imóveis, bens de consumo
- Análise de redes sociais

Scrapy!

- Framework especializado em web scraping
- Baterias Incluídas
Sessões/Cookies, Redirecionamentos, Caches, seletores XPath e CSS, exportação de dados, etc
- Extensível
Pipelines, Middlewares, Signals



<https://scrapy.org/>

Instalando o Scrapy

```
renne@capivara:~$ mkdir tutorial
renne@capivara:~$ cd tutorial
renne@capivara:tutorial$ python3 -m venv .venv
renne@capivara:tutorial$ source .venv/bin/activate
(.venv) renne@capivara:tutorial$ pip install scrapy
.
.
.
(.venv) renne@capivara:code$ scrapy version
Scrapy 1.5.1
```

Instalando o Scrapy (Win)

<https://doc.scrapy.org/en/1.5/intro/install.html#windows>

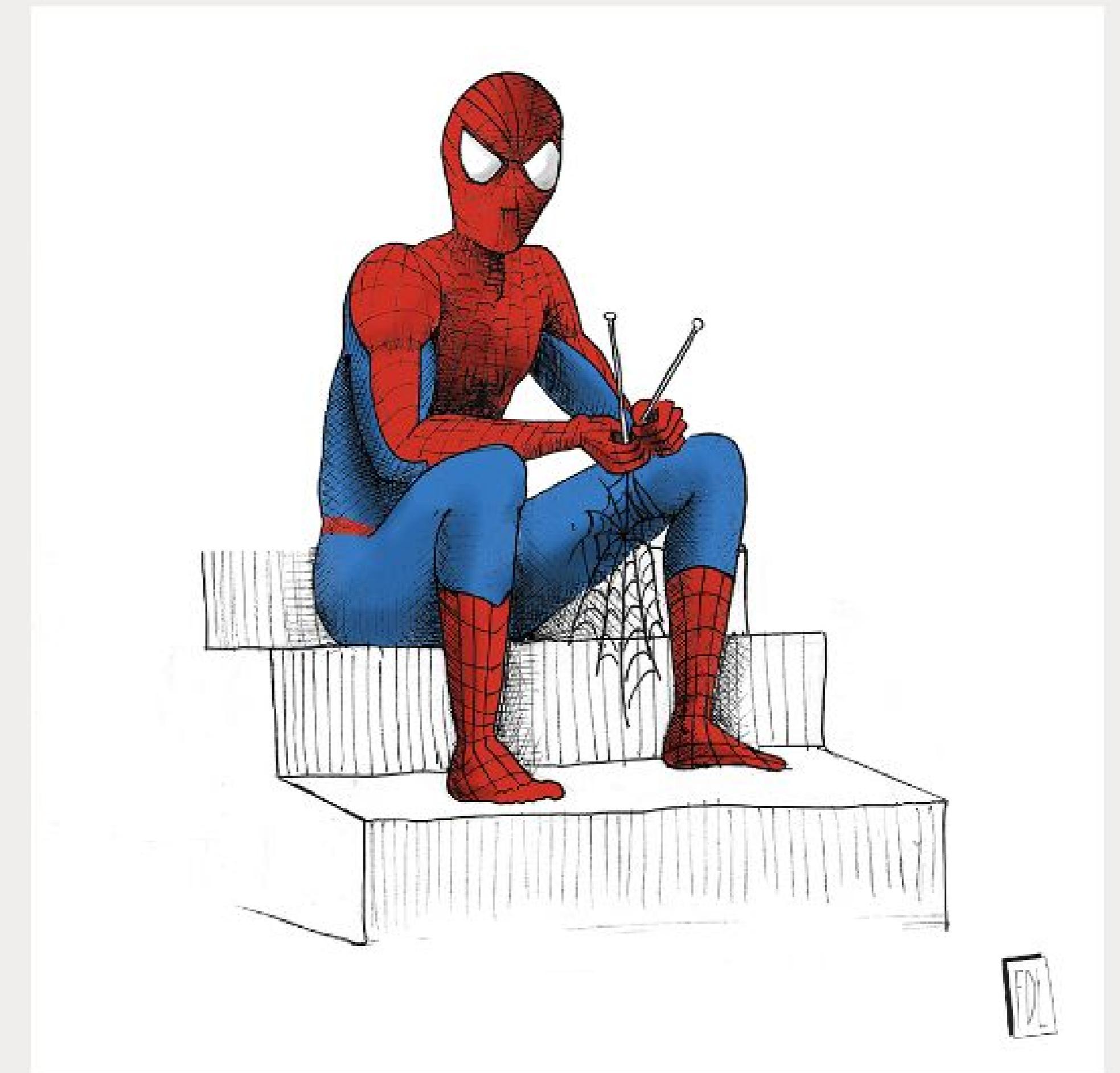
Apesar de ser possível instalar o Scrapy no Windows utilizando o pip, a recomendação é utilizar o Anaconda ou o Miniconda e depois usar o pacote do conda-forge. Isso evitará a maioria dos problemas de instalação.

Depois de instalar o Anaconda ou o Miniconda, instale o Scrapy usando:

```
conda install -c conda-forge scrapy
```

Arquitetura de um Spider

- Classe Python que contém a lógica de:
 - Extrair dados das páginas
 - Navegar pelas páginas
- É uma classe herdada de `scrapy.Spider`
- Um projeto pode ter **diversos** spiders



Arquitetura de um Spider

```
1 import scrapy
2
3 class MyFirstSpider(scrapy.Spider):
4     name = 'my-first-spider'
5
6     start_urls = [
7         'http://quotes.toscrape.com/page/1/',
8         'http://quotes.toscrape.com/page/2/',
9     ]
10
11    def parse(self, response):
12        self.logger.info(
13            'Just parsing {}'.format(response.url))
```

Arquitetura de um Spider

```
(.venv) renne@capivara ~/tutorial> scrapy runspider myfirstspider.py
2018-10-11 17:01:12 [scrapy.utils.log] INFO: Scrapy 1.5.1 started (bot: scrapybot)
(...)
2018-10-11 17:01:13 [scrapy.extensions.telnet] DEBUG: Telnet console listening on 127.0.0.1:6023
2018-10-11 17:01:13 [scrapy.core.engine] DEBUG: Crawled (200) <GET http://quotes.toscrape.com/page/2/> (referer: None)
2018-10-11 17:01:13 [scrapy.core.engine] DEBUG: Crawled (200) <GET http://quotes.toscrape.com/page/1/> (referer: None)
2018-10-11 17:01:13 [my-first-spider] INFO: Just parsing http://quotes.toscrape.com/page/2/
2018-10-11 17:01:13 [my-first-spider] INFO: Just parsing http://quotes.toscrape.com/page/1/
2018-10-11 17:01:13 [scrapy.core.engine] INFO: Closing spider (finished)
(...)
2018-10-11 17:01:13 [scrapy.core.engine] INFO: Spider closed (finished)
(.venv) renne@capivara ~/tutorial>
```

Arquitetura de um Spider

```
1 import scrapy
2
3 class MyFirstSpider(scrapy.Spider):
4     name = 'my-first-spider'
5
6     def start_requests(self):
7         urls = [
8             'http://quotes.toscrape.com/page/1/',
9             'http://quotes.toscrape.com/page/2/',
10        ]
11        requests = []
12        for url in urls:
13            requests.append(
14                scrapy.Request(url=url, callback=self.parse))
15
16        return requests
17
18    def parse(self, response):
19        self.logger.info('Just parsing {}'.format(response.url))
20
```

Extraindo Dados

Quotes to Scrape

[Login](#)

"The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking."

by [Albert Einstein](#) (about)

Tags: [change](#) [deep-thoughts](#) [thinking](#) [world](#)

"It is our choices, Harry, that show what we truly are, far more than our abilities."

by [J.K. Rowling](#) (about)

Tags: [abilities](#) [choices](#)

"There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle."

by [Albert Einstein](#) (about)

Tags: [inspirational](#) [life](#) [live](#) [miracle](#) [miracles](#)

<http://quotes.toscrape.com/>

Extraindo Dados

```
1 # quotes-1.py
2 import scrapy
3
4
5 class QuotesSpider(scrapy.Spider):
6     name = 'quotes'
7
8     start_urls = ['http://quotes.toscrape.com/']
9
10    def parse(self, response):
11        quotes = response.css('.quote')
12        for quote in quotes:
13            yield {
14                'quote': quote.css('.text::text').get(),
15                'author': quote.css('.author::text').get()
16            }
17
```

Extraindo Dados

```
(.venv) renne@capivara ~/tutorial> scrapy runspider quotes-1.py
2018-10-11 17:12:46 [scrapy.utils.log] INFO: Scrapy 1.5.1 started (bot: scrapybot)
(...)
2018-10-11 17:12:46 [scrapy.core.scrape] DEBUG: Scraped from <200 http://quotes.toscrape.com/>
{'quote': '“It is our choices, Harry, that show what we truly are, far more than our abilities.”',
'author': 'J.K. Rowling'}
2018-10-11 17:12:46 [scrapy.core.scrape] DEBUG: Scraped from <200 http://quotes.toscrape.com/>
{'quote': '“There are only two ways to live your life. One is as though nothing is a miracle.
The other is as though everything is a miracle.”', 'author': 'Albert Einstein'}
(...)
2018-10-11 17:12:58 [scrapy.core.engine] INFO: Spider closed (finished)
```

Extraindo Dados

```
(.venv) renne@capivara ~/tutorial> scrapy runspider quotes-1.py -o quotes.csv
2018-10-11 17:12:46 [scrapy.utils.log] INFO: Scrapy 1.5.1 started (bot: scrapybot)
(...)
2018-10-11 17:12:46 [scrapy.core.scrape] DEBUG: Scraped from <200 http://quotes.toscrape.com/>
{'quote': '“It is our choices, Harry, that show what we truly are, far more than our abilities.”',
'author': 'J.K. Rowling'}
2018-10-11 17:12:46 [scrapy.core.scrape] DEBUG: Scraped from <200 http://quotes.toscrape.com/>
{'quote': '“There are only two ways to live your life. One is as though nothing is a miracle.
The other is as though everything is a miracle.”', 'author': 'Albert Einstein'}
(...)
2018-10-11 17:12:58 [scrapy.extensions.feedexport] INFO: Stored csv feed (10 items) in: quotes.csv
2018-10-11 17:12:58 [scrapy.core.engine] INFO: Spider closed (finished)
```

Extraindo Dados

```
(.venv) renne@capivara ~/tutorial> scrapy runspider quotes-1.py -o quotes.csv
2018-10-11 17:12:46 [scrapy.utils.log] INFO: Scrapy 1.5.1 started (bot: scrapybot)
(...)
2018-10-11 17:12:46 [scrapy.core.scrape] DEBUG: Scraped from <200 http://quotes.toscrape.com/>
{'quote': '“It is our choices, Harry, that show what we truly are, far more than our abilities.”',
'author': 'J.K. Rowling'}
2018-10-11 17:12:46 [scrapy.core.scrape] DEBUG: Scraped from <200 http://quotes.toscrape.com/>
{'quote': '“There are only two ways to live your life. One is as though nothing is a miracle.
The other is as though everything is a miracle.”', 'author': 'Albert Einstein'}
(...)
2018-10-11 17:12:58 [scrapy.extensions.feedexport] INFO: Stored csv feed (10 items) in: quotes.csv
2018-10-11 17:12:58 [scrapy.core.engine] INFO: Spider closed (finished)
```

Outros formatos: JSON, JL, XML

Callbacks

```
1 #simplespider.py
2 import scrapy
3
4 class SimpleSpider(scrapy.Spider):
5     name = 'simplespider'
6     start_urls = ['http://scrapy.org/']
7
8     def parse(self, response):
9         self.logger.debug('Site visited: {}'.format(response.url))
10        yield {'url': response.url, 'size': len(response.body)}
11
12     next_url = 'http://python.org/'
13     self.logger.debug('Next site: {}'.format(next_url))
14
15     yield scrapy.Request(next_url, self.handle_python)
16
17     def handle_python(self, response):
18         self.logger.debug('Python site visited: {}'.format(response.url))
```

Extraindo Dados

Seletores CSS

<https://www.w3.org/TR/selectors/>

Seletores XPath

<https://www.w3.org/TR/xpath/all/>

Extraindo Dados

```
1 <html>
2   <body>
3     <h1>Last Offers</h1>
4     <ul id="offers">
5       <li class="product">
6         <a href="http://mystore.com/product-1">Product 1</a>
7         <p>I am a great product! Buy me!</p>
8       </li>
9       <li class="product bestseller">
10        <a href="http://mystore.com/product-2">Product 2</a>
11        <p><ul><li>abc</li></ul>
12        I am a better! Buy me!</p>
13      </li>
14      <li class="ad">
15        <a href="http://otherstore.com/product-2">Ad Product 2</a>
16        <p>I am an ad product! I paid to be here!</p>
17      </li>
18      <li class="product">
19        <a href="http://mystore.com/product-2">Product 3</a>
20        <p>Ok, you won't buy me anyway :-(</p>
21      </li>
22    </ul>
23
24    <p class="bestseller">Teste</p>
25
26    <h1>You may like</h1>
27    <ul id="recommendations">
28      <li class="product">
29        <a href="http://recommendation.com/recommendations-product-1">Recommended Product 1</a>
30        <p>Probably you will like me too.</p>
31      </li>
32      <li class="product">
33        <a href="http://recommendation.com/recommendations-product-2">Recommended Product 2</a>
34        <p>Probably you will like me too (2).</p>
35      </li>
36    </ul>
37  </body>
38 </html>
```

Extraindo Dados

```
(.venv) renne@capivara:code$ pip install ipython
(.venv) renne@capivara:code$ ipython
Python 3.6.4 (default, Mar 26 2018, 15:25:21)
Type 'copyright', 'credits' or 'license' for more infor
IPython 6.5.0 -- An enhanced Interactive Python. Type '
```

```
In [1]: from parsel import Selector
```

```
In [2]: with open('product_list.html') as code:
...:     response = Selector(text=code.read())
...:
```

```
In [3]:
```

Extraindo Dados

```
# Exemplo de Seletores CSS

response.css('h1')

response.css('ul#offers')

response.css('.product')

response.css('ul#offers .product')

response.css('ul#offers .product a::attr(href)')

response.css('ul#offers .product *::text')

response.css('ul#offers .product p::text')
```

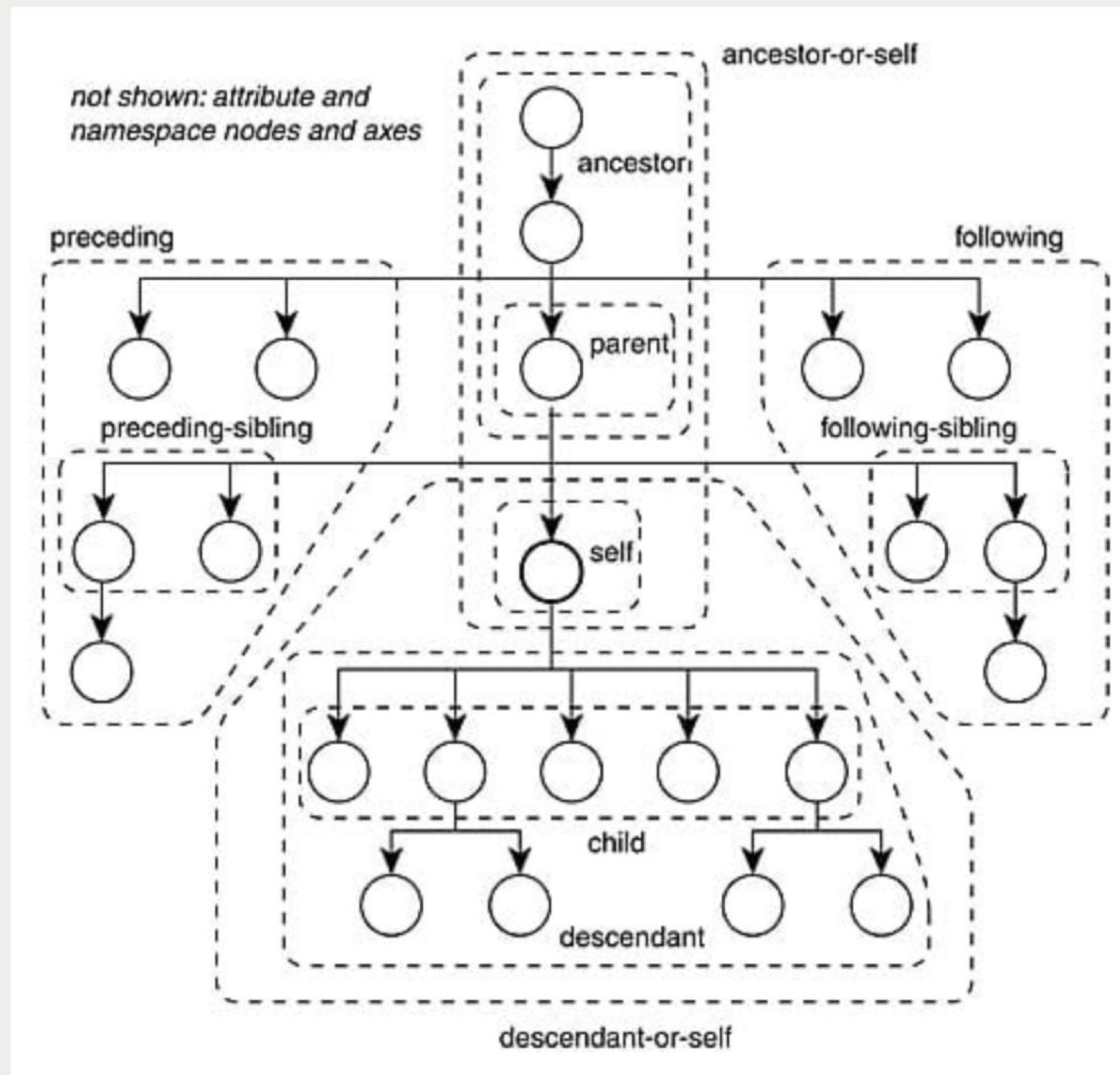
Extraindo Dados

```
1 # Exemplo de seletores XPath  
2  
3 response.xpath('//h1')  
4  
5 response.xpath('//h1[2]')  
6  
7 response.css('ul[@id="offers"]')  
8  
9 response.xpath('//li/a/@href')  
10  
11 response.xpath('//li/text()')  
12  
13 response.xpath('//li//text()')  
14  
15 response.xpath('//p/text()')
```

Extraindo Dados

```
1 response.xpath(  
2     '//ul[@id="offers"]//li[@class="product"]'  
3 )  
4  
5 response.xpath(  
6     '//ul[@id="offers"]//li[contains(@class, "product")]'  
7 )  
8  
9 response.xpath(  
10    '//li[@class="ad"]/following-sibling::li'  
11    '[@class="product"]').getall()
```

Extraindo Dados



<http://www.informit.com/articles/article.aspx?p=29844&seqNum=3>

Hora de escrever código



Exercício 1

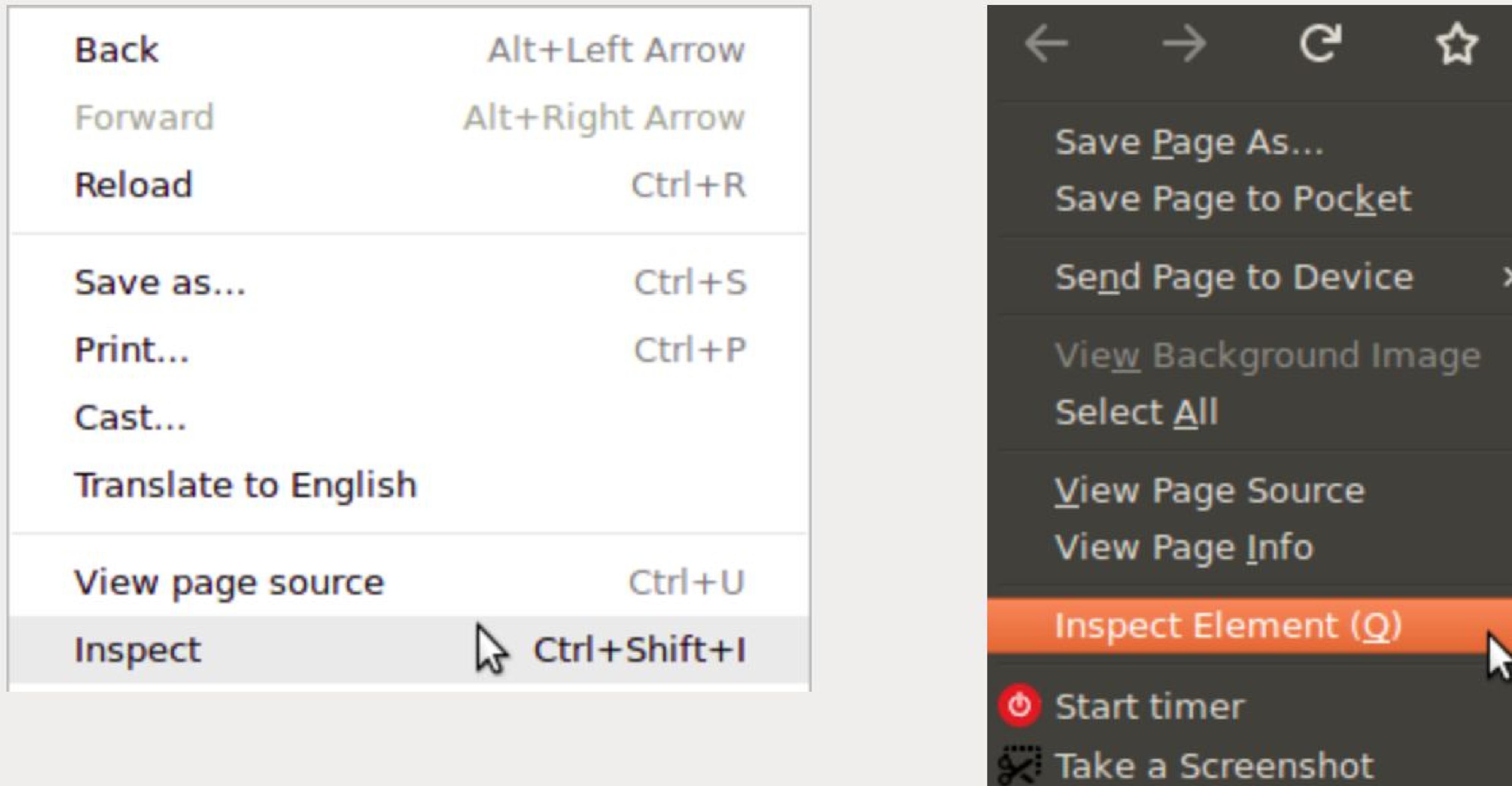
<http://quotes.toscrape.com/>

Nesse site temos uma lista de citações com seus autores, um link para uma página com mais informações sobre o autor e um conjunto de tags.

Queremos extrair as seguintes informações:

- Nome do Autor
- URL da página de detalhes do autor
- A citação
- A lista de tags

Exercício 1



CUIDADO com a opção "Inspecionar elemento". Diferentemente do resultado da opção "Exibir código-fonte", o código que você vê representa as estruturas que o navegador cria para a página, e nem sempre é o mesmo código HTML que veio na requisição HTTP (que é o que você obtém quando usa o Scrapy). Isso é muito importante ao trabalhar com páginas que usam muito Javascript ou chamadas AJAX.

Exercício 1

<http://quotes.toscrape.com/>

Nesse site temos uma lista de citações com seus autores, um link para uma página com mais informações sobre o autor e um conjunto de tags.

Queremos extrair as seguintes informações:

- Nome do Autor
- URL da página de detalhes do autor
- A citação
- A lista de tags

Exercício 2

<http://quotes.toscrape.com/scroll>

Houve uma mudança na página de citações.
Agora ao invés de uma paginação simples,
temos uma página com scroll infinito.

Dica: ao abrir a opção "Inspecionar Elemento" no seu navegador, escolha a aba "Network" e veja o que acontece quando você navega até o fim da página.

Exercício 3

<http://quotes.toscrape.com/login>

Para acessar a página, é necessário fazer login (utilize qualquer valor para usuário e senha).

Exercício 3

http://quotes.toscrape.com/login

Para acessar a página, é necessário fazer login (utilize qualquer valor para usuário e senha).

```
1 scrapy FormRequest(  
2     url="http://www.example.com/post/action/login",  
3     formdata={  
4         'login': 'myusername',  
5         'password': 'mypassword'  
6     },  
7     callback=self.after_post  
8 )
```

Exercício 4

<http://quotes.toscrape.com/js/>

Houve uma mudança na página de citações.
Agora o conteúdo é gerado por um código
Javascript.

Próximos passos?

Assistam a palestra:

Scrapy beyond the first steps

Com Eugenio Lacuesta neste sábado (20/10) às 13h40