Curso Webscrapping

Michel Alves

11/03/2022

Unidade 1

O que é Webscrapping?

O *web scraping* (raspagem de rede, em tradução livre), também conhecido como extração de dados da web, é o nome dado ao processo de coleta de dados estruturados da web de maneira automatizada. Os dados são então exportados para um formato mais útil ao usuário.

Em geral, esse método é usado por pessoas, empresas e, preocupantemente, criminosos, que desejam usar a vasta quantidade de dados da web disponíveis publicamente para tomar decisões mais inteligentes ou cometer crimes. Quando usado para o bem, pode ser aplicado em monitoramento de preços, monitoramento de notícias e pesquisa de marketing, entre outros.

O processo básico de raspagem, na verdade, é realizado diariamente por boa parte da população mundial, no ato de copiar e colar informações de um site para outro meio, a diferença é que a raspagem de rede faz isso em uma escala macroscópica e com automação inteligente, para extrair milhões de dados de páginas da internet.

No entanto, o processo de webscrapping não é uma tarefa simples. Os websites se apresentam em muitas formas e, como resultado, os webscrappers variam em funcionalidades e características. Também pode acontecer de encontrarmos captchas (aquele teste para verificar se você é humano) quando tentando realizar a raspagem de dados. Esses casos, infelizmente, não serão tratados nesse curso.

Para realizar a raspagem de dados, existem duas abordagens diferentes, que dependem de como o site estrutura suas informações

Abordagem 1: Se o site armazena todas as suas informações no *front end* do HTML, você pode baixar o conteúdo do HTML diretamente e extrair as informações úteis. São baiscamente 5 passos:

1. Identificar o website desejado;
2. Coletar as URLs das páginas das quais você deseja extrair os dados;
3. Fazer uma requisição para essas URLs para obter o HTML da página;
4. Usar os localizadores para encontrar os dados no HTML;
5. Salvar os dados em formato CSV ou XLSX, entre outros.

Abordagem 2: Se o site armazena os dados em um API e o site faz uma requisição para a API a cada vez que o usuário deseja obter os dados, então você pode simular a requisição e fazer o pedido diretamente pela API. Os passos são:

1. Inspecionar o seção de rede da URL da qual você deseja obter os dados;
2. Descobrir a forma da requisição-resposta da URL da qual você deseja obter os dados;
3. Dependendo do tipo de requisição (POST ou GET) e também do *header* e *payload*, simular a requisição no seu código para obter os dados;
4. Extrair as informações desejadas;
5. Salvar os dados em formato CSV ou XLSX, entre outros.

Quando é preciso realizar webscrapping?

O processo de webscrapping é necessário quando os dados desejados estão disponíveis dentro de um código HTML. Em outros casos, pode ser usado para obter arquivos que contém os dados desejados.

Estrutura de uma página web

Páginas são criadas usando HTML (HyperText Markup Language). O HTML descreve os elementos de uma página por meio de *tags* marcadas por `< >`. Exemplo:

```
<html>
<head>
<body>
<h1>Minha página</h1>
<h2>Segundo título</h2>
<h3>Terceiro Título</h2>
</body>
</head>
</html>
```

No código HTML acima, os principais elementos são: * O documento sempre inicia e termina usando `<html>` e `</html>`; * A parte visível do documento pe marcada por `<body></body>`; * As tags `<h1>` até `<h3>` são usadas para títulos.

Podemos também adicionar um parágrafo abaixo do segundo título, usando a tag `<p>`:

```
<html>
<head>
<body>
<h1>Minha página</h1>
<h2>Segundo título</h2>
<p>"Duas coisas são infinitas. O universo e a estupidez humana." Albert Einstein</p>
<h3>Terceiro Título</h2>
</body>
</head>
</html>
```

Podemos adicionar um link na palavra “Albert Einstein” para direcionar os usuários para uma página da wikipédia, por exemplo, Para isso, usamos a tag `<a>`. Ela tem o atributo `href` que especifica o link:

```
<html>
<head>
<body>
<h1>Minha página</h1>
<h2>Segundo título</h2>
<p>"Duas coisas são infinitas. O universo e a estupidez humana." <a href="https://it.wikipedia.org/wiki/Albert_E
<h3>Terceiro Título</h2>
</body>
</head>
</html>
```

Para melhorarmos ainda mais a nossa página, podemos adicionar uma imagem de Albert Einstein. Para isso usamos a tag `<img>`, que por sua vez tem o atributo `src` para especificarmos a URL da imagem.

```
<html>
<head>
<body>
<h1>Minha página</h1>
<h2>Segundo título</h2>
<p>"Duas coisas são infinitas. O universo e a estupidez humana." <a href="https://it.wikipedia.org/wiki/Albert_E
<img src="https://upload.wikimedia.org/wikipedia/commons/thumb/d/d3/Albert_Einstein_Head.jpg/800px-Albert_Einste
<h3>Terceiro Título</h2>
</body>
</head>
</html>
```

Classes e ID

- O **id** é um atributo para especificar um único elemento e dar características específicas a ele;
- A **class** é um atributo para definir diferentes elementos com o mesmo nome class. Isso é útil quando queremos dar as mesmas características, como fonte, cor e tamanho, para vários elementos de uma página.

Ambos são definidos na tag `<style>` e as propriedades são definidas usando chaves (`{}`). A sintaxe para `class` precisa do ponto (`.`) seguido pelo nome da classe, enquanto o `id` precisa da hashtag (`#`) seguido pelo nome do `id`. Uma vez criados o `id` e `class` na tag `<style>`, temos que passá-los para os elementos que queremos. Nesse caso o `id` é chamado na tag `<h1>` enquanto a `class` é chamada nas tags `<p>`. Além disso, as tags `<i>` e `<b>` são usadas para exibir o texto em itálico e negrito, respectivamente.

```
<html>
<head>
<style>
#mytitle {
  font-size: 60;
  color: blue;
}
.cit {
  background-color: pink;
  color: white;
  font-family: Arial, Helvetica, sans-serif;
}
</style>
<body>
<h1 id="mytitle">Minha página</h1>
<h2>Segundo título</h2>
<p class='cit'><i>"Duas coisas são infinitas. O universo e a estupidez humana.<i>" <a href="https://it.wikipedia
<p class='cit'>"<i>Imaginação é mais importante do que conhecimento<i>." <b>Albert Einstein</b></p>
</body>
</head>
</html>
```

Tabelas

Outro recurso importante do HTML são as tabelas, que são definidas pela tag `<table>`. Dentro de `<table>`, existem três principais tags que são utilizadas:

- A tag `<tr>` é usada para construir cada linha da tabela;
- A tag `<th>` é usada para definir o cabeçalho;
- A tag `<td>` é usada para definir a célula dentro da linha.

Exemplo:

```
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
<body>
<h1 id="mytitle">Minha página</h1>
<h2>Segundo título</h2>
<p class='cit'><i>"Duas coisas são infinitas. O universo e a estupidez humana.<i>" <a href="https://it.wikipedia
<table>
<tr><th>Descobertas</th><tr>
<tr><td>Relatividade Especial</td></tr>
```

```
<tr><td>Relatividade Geral</td></tr>
</table>
</body>
</head>
</html>
```

Na tag `<style>` foram definidas as propriedades para as tags `<table>`, `<th>` e `<td>`.

Listas

Existem dois tipos de listas que podem ser definidas em HTML. A primeira é uma lista não ordenada que começa pela tag `<ul>`, enquanto o outro tipo é uma lista ordenada especificada pela tag `<ol>`. Cada item em cada um dos tipos de lista é especificado pela tag `<li>`. A seguir temos um exemplo.

```
<html>
<head>
<body>
<h1>Minha página</h1>
<h2>Segundo título</h2>
<p>"<i>Duas coisas são infinitas. O universo e a estupidez humana<i>." <a href="https://it.wikipedia.org/wiki/Al
<p>Descobertas</p>
<ul>
<li>Relatividade Especial</li>
<li>Relatividade Geral</li>
</ul>
<p>Prêmios</p>
<ol>
<li>Medalha Max Planck</li>
<li>Nobel de Física</li>
</ol>
</body>
</head>
</html>
```

Blocos

Um dos elementos mais comuns em um página é chamado de *Block* ou *Container*. Ele é útil para agrupar differentes elementos a aplicar a eles as mesmas propriedades.

Por exemplo, suponha que queremos dividir a página em duas partes. Para criar esses dois blocos, usamos a tag `<div>`. No exemplo, definimos a tag "linha" para definir a estrutura das duas partes na mesma linha e uma classe com o nome coluna para especificar as propriedades de cada parte da página.

Além disso, usamos o seletor `*` para selecionar todos os elementos e aplicar a propriedade que define o tamanho da caixa.

```
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

/* Cria duas colunas que são dispostas lado a lado */
.column {
  float: left;
  width: 50%;
  padding: 10px;
  height: 300px;
}
```

```
.row:after {
  content: "";
  display: table;
  clear: both;
}
</style>
<body>

<h2>Glossário</h2>

<div class="row">
  <div class="column" style="background-color:#e6e6ff;">
    <h2>Relatividade Geral</h2>
    <p>A relatividade geral, também conhecida como a teoria geral da relatividade, é a teoria geométrica da grav
  </div>
  <div class="column" style="background-color:#9999ff;">
    <h2>Relatividade Especial</h2>
    <p>Na física, a teoria da relatividade especial, ou relatividade especial, é uma teoria científica sobre a r
  </div>
</div>

</body>
</head>
</html>
```

Hierarquia

Os elementos HTML possuem uma hierarquia entre si. Conhecer a relação entre os elementos pode ser útil para realizar a busca de um elemento dentro do HTML.

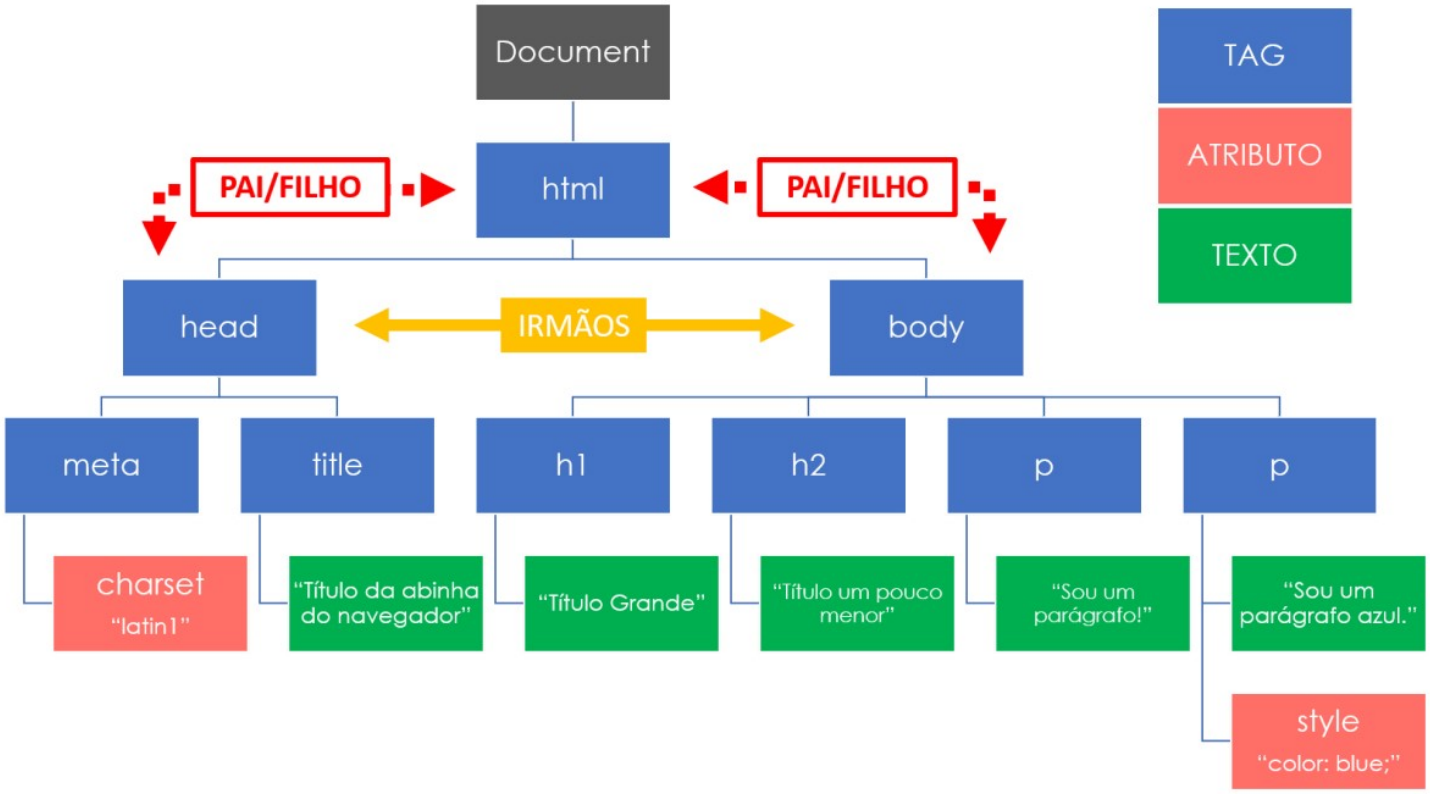

Figure 1: Hierarquia dos elementos HTML

XML

XML é uma ferramenta para armazenar e transporta informações. Algumas características

- XML é a sigla de *eXtensible Markup Language*
- XML é parecido com HTML
- XML foi desenvolvido para armazenar e transportar dados
- XML foi desenvolvido de forma a ser auto-descritivo

Abaixo temos um exemplo do uso do XML:

```
<note>
  <to>João</to>
  <from>Maria</from>
  <heading>Lembrete</heading>
  <body>Não me esqueça nesse fim de semana!</body>
</note>
```

Apesar de conter informações, o conteúdo em XML não faz nada por si só. É necessário que alguém escreva algum software para receber, enviar, armazenar or exibir.

Como visualizar o HTML de uma página

A maioria dos navegadores atuais possuem um recurso conhecido como **Ferramentas para Desenvolvedores**. Embora a maioria de suas funcionalidades sejam úteis para desenvolvedores da web, algumas delas são essenciais para realizar o webscrapping. Para abrir a ferramenta, basta clicar com o botão direito do mouse em algum elemento da página e em seguida escolher a opção Inspecionar. Outra opção é apertar a tecla F12 no teclado ou o atalho CTRL+SHIFT+I.

Algumas Dicas

Edição de uma página Ir na guia Console e digitar o seguinte comando:

```
document.body.contentEditable = true
```

Isso torna a página completamente editável. No entanto, as modificações desaparecem quando a página é atualizada.

Visualização de senhas Se você se esqueceu de alguma senha e ela está salva no navegador, basta clicar com o botão direito no campo de senha e escolha "Inspecionar item". Isso abrirá a janela Inspecionar documento e tudo o que você precisa fazer é substituir "senha" por "texto" no campo de entrada do tipo de senha. Isso deve revelar a senha oculta com asteriscos.

Bibliotecas no R para webscrapping

Existem várias possibilidades de usos de ferramentas para webscrapping em R. Veremos algumas delas.

```
library("xml2")

url <- "http://www.fjp.mg.gov.br"

pagina <- xml2::read_html(url)
```

Referências

- https://canaltech.com.br/seguranca/o-que-e-web-scraping/
- https://www.zyte.com/learn/what-is-web-scraping/
- https://ubc-library-rc.github.io/intro-web-scraping/content/understanding-a-website.html
- https://betterprogramming.pub/understanding-html-basics-for-web-scraping-ae351ee0b3f9
- https://www.w3schools.com/xml/xml_whatis.asp
- https://www.w3schools.com/css/css_intro.asp
- https://lente.dev/webscraping.pdf