DataScientest • com

Web scraping with Python

Inspect a web page

Context

What is a web?

World Wide Web, or more simply the web is a tool for consulting websites via a browser (Firefox, Chrome, Safari, ...).
This web interacts with servers to share the hypertext documents they host. These files are identified by their URL which is an IP address, actually addressing the computer on which the site is hosted.
A web page consists of a hypertext document and various files. Thus, a website is a set of web pages linked to each other by links to easily navigate from one document to another.

What is the web scraping and what is its use?

Web scraping is a technique to automate data recovery from web pages. It collects the content of the sites and extract the relevant data.

In general, scraping is widely used in the marketing and insurance sector. Small as large companies use it to monitor and compare the prices of online offers, ensure their presence and reputation on the web and much more.

In the field of artificial intelligence (AI), scraping is an essential tool. The AI depends on the data and thus, automating the extraction of the data makes it possible to facilitate and optimize automatic learning. Therefore, this tool is widely used to create or complete databases for all types of projects.

Scraping challenges

At first, it is necessary to access the source code of the web pages that we want to scrap. However, this source code can constantly change and make scraping obsolete. It is therefore necessary, once the data has been recovered, save them in a file on a local computer.

Scraping is not necessarily an obvious practice. The two main obstacles that a scraper may meet are as follows.

- As we have just mentioned, data collection is carried out by accessing the source code of web pages, so any modification of the structure of a web page can make scraping obsolete. Thus, the updates of websites are real challenges at web scraping.
- In addition, some sites try to protect themselves from scrapers by implementing different strategies to make recovery of data tedious. Anti-scraping technologies or invisible traps (called HoneyPots) can thus slip into the web page and block the actions of the scraper.

Tool right

Collecting data using scraping web tools is completely legal. On the other hand, it is the use of the data later that can be problematic. Each site has its own conditions of use and it is important to document yourself on this subject before any illegal maneuver.

Goals

In this module, we will take an example a page of the Amazon site. This page is accessible via any browser thanks to the site URL (https://www.amazon.fr).
The objective will be to recover information concerning the best sales of literature books.
If the site is not accessible (for maintenance or any other reason), it is quite possible to practice on another web page.

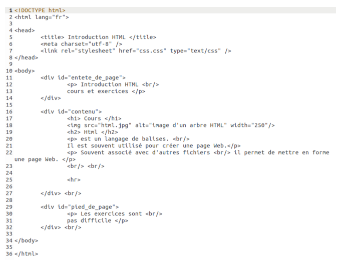
More precisely, this module will allow you to:

- understand and know how to read the sources of a web page
- knowing how to navigate in an HTML document and identify page data
- recover site data

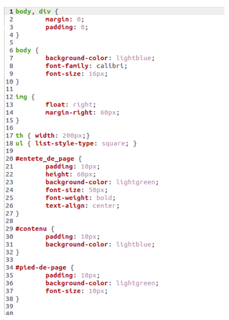
Inspect a web page

The content of the web pages can be coded in different programming languages. This content is herberged in one or more documents whose type varies depending on the language used. In most cases, a web page is coded, among other things, with HTML and CSS languages and is therefore made up of an HTML document and a CSS file.

Example of an HTML document (left) and a CSS file (right)



```html
<!DOCTYPE html>
<html lang="fr">

<head>
        <title> Introduction HTML </title>
        <meta charset="utf-8" />
        <link rel="stylesheet" href="css.css" type="text/css" />
</head>

<body>
        <div id="entete_de_page">
                <p> Introduction HTML <br/>
                cours et exercices </p>
        </div>

        <div id="contenu">
                <h1> Cours </h1>
                <img src="html.jpg" alt="image d'un arbre HTML" width="250"/>
                <h2> Html </h2>
                <p> est un langage de balises. <br/>
                Il est souvent utilisé pour créer une page Web.</p>
                <p> Souvent associé avec d'autres fichiers <br/> il permet de mettre en forme
une page Web. </p>
                <br/> <br/>

                <hr>

        </div> <br/>

        <div id="pied_de_page">
                <p> Les exercices sont <br/>
                pas difficile </p>
        </div> <br/>

</body>

</html>
```



```css
body, div {
        margin: 0;
        padding: 0;
}

body {
        background-color: lightblue;
        font-family: calibri;
        font-size: 16px;
}

img {
        float: right;
        margin-right: 60px;
}

th { width: 200px;}
ul { list-style-type: square; }

#entete_de_page {
        padding: 10px;
        height: 60px;
        background-color: lightgreen;
        font-size: 50px;
        font-weight: bold;
        text-align: center;
}

#contenu {
        padding: 10px;
        background-color: lightblue;
}

#pied-de-page {
        padding: 10px;
        background-color: lightgreen;
        font-size: 10px;
}
```

Using a right click with your mouse, it is possible to access the source web page code on which you are. The inspectory 'button or display the source code of the page` then allows you to display the different languages with which the page is coded.

In [1]:

```python
### Insert your solution
import pandas as pd
```

Hide solution

In [ ]:

```python
print("At the top, we have access to the Amazon HTML document. At the bottom, we observe the CSS code of the page.")
```

But before using the necessary tools to recover the source code and extract the data, it is important to know how to identify and understand the different web languages that you risk meeting during your navigations.

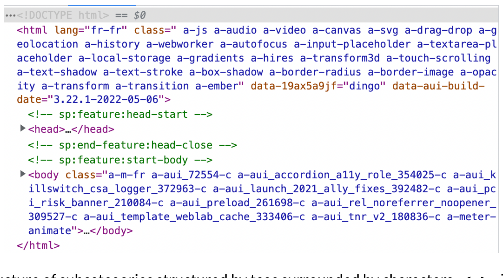
Introduction to web languages

Web pages, intended to be deployed on the internet, can be published in several languages (HTML, GML, XML etc). The main web languages used to build web applications are the following three:

1. HTML

HTML for HyperText Markup Language allows the management and organization of the content of the page. This language is hosted in an HTML document.

Image of the HTML code of the Amazon page with closed tags



```
•••<!DOCTYPE html> == $0
  <html lang="fr-fr" class=" a-js a-audio a-video a-canvas a-svg a-drag-drop a-g
  eolocation a-history a-webworker a-autofocus a-input-placeholder a-textarea-pl
  aceholder a-local-storage a-gradients a-hires a-transform3d a-touch-scrolling
  a-text-shadow a-text-stroke a-box-shadow a-border-radius a-border-image a-opac
  ity a-transform a-transition a-ember" data-19ax5a9jf="dingo" data-aui-build-
  date="3.22.1-2022-05-06">
    <!-- sp:feature:head-start -->
   ▶<head>…</head>
    <!-- sp:end-feature:head-close -->
    <!-- sp:feature:start-body -->
   ▶<body class="a-m-fr a-aui_72554-c a-aui_accordion_a11y_role_354025-c a-aui_k
    illswitch_csa_logger_372963-c a-aui_launch_2021_ally_fixes_392482-c a-aui_pc
    i_risk_banner_210084-c a-aui_preload_261698-c a-aui_rel_noreferrer_noopener_
    309527-c a-aui_template_weblab_cache_333406-c a-aui_tnr_v2_180836-c a-meter-
    animate">…</body>
  </html>
```

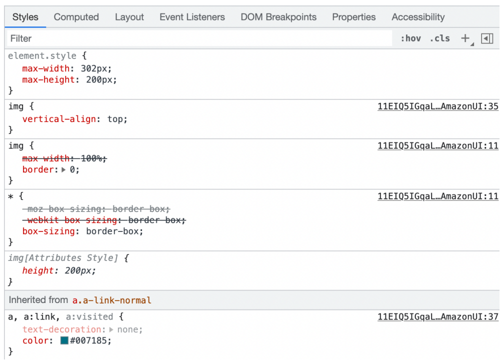
An HTML page is organized as a tree structure of subcategories structured by tags surrounded by characters `< >`. The minimum code to create a valid HTML page is made up of the following tags:

- `<!DOCTYPE html>` : an HTML document always starts by specifying the type of the document.
- `<html>` : the opening tag <html> and the closing tag </html> frame the entire page code. The document is then split in half, the head and the body.
- `<head>` : the tag <head> is a header element. It contains general characteristics of the document such as nature and content or the title of the page.
- `<title>` : allows you to indicate the title of the visible page on the top of the tabs of your browser.
- `<meta>` : this tag makes it possible to communicate to navigators the metadata on the page by giving the nature and content of the page.
- `<body>` : contains the elements defining the content of the page to the user as the different texts, images, etc.

2. CSS

CSS for Cascading Style Sheets, allows managing the appearance of the web page (colors, size, font, layout, etc.).
It completes the HTML and can be directly included in an HTML document by being stored in the tags `<style>` generally located in the header element `<head>`. Otherwise, it is hosted in a dedicated and stored CSS file in the element `<link/>` also located in `<head>` in most of the cases.

Image of the CSS code of the Amazon page



```
element.style {
   max-width: 302px;
   max-height: 200px;
}
img {                                          11EIQ5IGqaL…AmazonUI:35
   vertical-align: top;
}
img {                                          11EIQ5IGqaL…AmazonUI:11
   ~~max-width: 100%;~~
   border: ▸ 0;
}
* {                                            11EIQ5IGqaL…AmazonUI:11
   ~~-moz-box-sizing: border-box;~~
   ~~-webkit-box-sizing: border-box;~~
   box-sizing: border-box;
}
img[Attributes Style] {
   height: 200px;
}
Inherited from a.a-link-normal
a, a:link, a:visited {                         11EIQ5IGqaL…AmazonUI:37
   text-decoration: ▸ none;
   color: ■#007185;
}
```

3. Javascript

In [2]:
```
### Insert your code
```

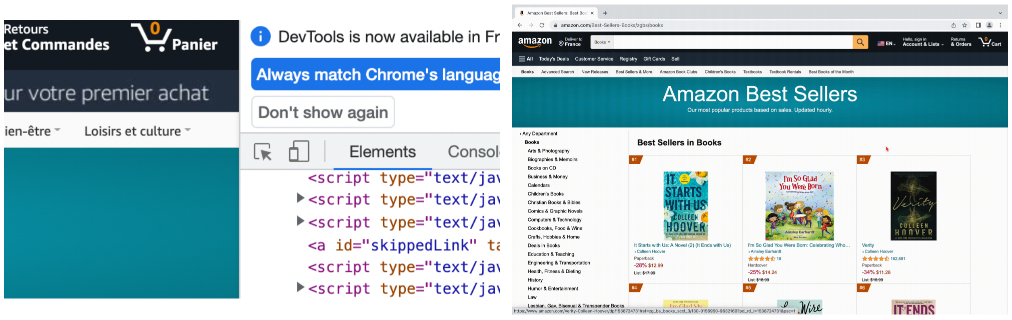
Hide solution

In [5]:
```
print("There are two ways to identify that it is indeed an interactive web page:\n"
"1. The page contains many interaction buttons (image, title, ...)\n"
"2. Many tags <script> are present in the HTML document.")
```

```
There are two ways to identify that it is indeed an interactive web page:
1. The page contains many interaction buttons (image, title, ...)
2. Many tags <script> are present in the HTML document.
```

After accessing the source code of the web page, you must be able to find the information visible on the web page. For this, you can:

- navigate with your mouse in the available code after inspected the page (each tag is associated with part of the page).
- click on the arrow visible on the image below and then identify the information by moving your mouse on the data. Three elements are displayed: the tag, the name of the class and the size as well as the location of the element code in the HTML document visible on the left.



```
<script type="text/ja
▸<script type="text/ja
▸<script type="text/ja
 <a id="skippedLink" t
 <script type="text/ja
▸<script type="text/ja
```

- **e)** Display the information tags (image, title, prize, ...) of the first book.
- **f)** Locate the source code relating to the various information in the book. For each information, what word follows the name of the tag?

In [3]:
```
### Insert your code
```

Hide solution

In [4]:

```python
### The solution can be obsolete due to the constant modifications of the Amazon source code

# By clicking on the arrow mentioned above, you can easily identify the tags (purple characters)

print("Image: img \n",
      "Titre: div \n",
      "Ecrivain: div \n",
      "Format: span \n",
      "Prix: span \n")

print('For each information, the tags are followed by the word class')
```

```
Image: img
 Titre: div
 Ecrivain: div
 Format: span
 Prix: span

For each information, the tags are followed by the word class
```

> The web scraping is carried out in several steps: recover the source code, find the information that interests us and extract it. \ Consequently, identifying the tag and, more generally, locating the code of a data is essential to recover the information of the site.

Main Python modules

> The objective being to collect information in an automated manner, we will then use a scraping web module which will allow this task to be completed.
> Python has several modules allowing the consultation of web resources, here are the three main ones:

1. BeautifulSoup

> Simple and quick to handle, Beatifulsoup is the ideal module to start in the web scraping. It is for this reason that it will be introduced to you in the rest of the module.
> On the other hand, this tool is not very robust and is therefore not the most suitable for carrying out complex projects.

2. Scrapy

> Faster and more complete, Scrapy is a more efficient module than BeautifulSoup. More difficult to master, it is a better choice for delicate projects.

3. Selenium

> Selenium is above all a tool that was mainly created for automated web tests. Due to its compatibility with JavaScript, it is also used for web scraping. And, unlike the two previous modules, since Selenium deciphers JavaScript, he is able to do dynamic scraping. It is a very efficient and fast tool that is more or less as efficient as Scrapy.

Unvalidate