Chapitre V Web Scraping

Cours Data Science M1 IA-GL

Ilyas Bambrik

2024

Table des matières

Introduction 3

I - **Collecte de données** 4

II - **BeautifulSoup** 8

Références 13

Introduction

Dans cette section, nous allons aborder à la collection de données avec deux modules particuliers. Premièrement, le module *requests* permettant de transmettre des requêtes HTTP et de recevoir les données renvoyées par le serveur. Par la suite, le module *BeautifulSoup* est utilisé pour extraire les informations pertinentes.

Collecte de données



En science des données, nous pouvons réaliser de nombreux travaux avec le bon DataSet. Une fois que nous disposons de données intéressantes, nous pouvons utiliser Pandas ou Matplotlib pour analyser ou visualiser les tendances. Mais comment obtenir ces données en premier lieu ?

S'ils nous sont fournies dans un fichier csv ou json structuré / semi-structuré, le reste du travail est simple! Par contre, la plupart du temps, nous devons collecter les données nous-mêmes.

Souvent, vous trouverez le site web idéal contenant toutes les données dont vous avez besoin. C'est là que *BeautifulSoup* est utile pour gratter (scrap) le code HTML. Si nous trouvons les données que nous souhaitons analyser en ligne, nous pouvons utiliser *BeautifulSoup*[19*] pour les transformer en une structure que nous pouvons comprendre. Cette bibliothèque Python nous permet de récupérer facilement et rapidement des informations d'un site web et de les mettre dans un DataFrame.

Lorsque vous utilisez le Web scraping sur des sites web, il est nécessaire de suivre certain directives afin de respecter les sites et les conditions d'utilisation du site Web. Lisez la déclaration sur l'utilisation légale des données. Habituellement, les données que vous récupérez ne doivent pas être utilisées à des fins commerciales.

Ne spammez pas le site web avec des milliers de requêtes par seconde. Un grand nombre de requêtes peut détruire un site web qui n'est pas préparé à cette quantité de trafic. En règle générale, une bonne pratique est d'effectuer une requête sur une page Web chaque 30 secondes.

Par ailleurs, cette opération dépend de la structure du site web cible. Si la mise en page du site change, vous devrez modifier votre code de scraping pour être conforme à la nouvelle structure du site.

Python dispose d'une bibliothèque *requests* facilite grandement l'obtention de contenu. Tout ce que nous avons à faire est d'importer la bibliothèque, puis d'introduire l'URL du contenu que nous voulons obtenir. Le code suivant envoie la requête *GET* pour la ressource située dans la page *https://www.licbplus.com.dz/* :

```
1 import requests
2
3 webpage = requests.get('https://www.licbplus.com.dz/')
4 print(webpage.text)
```

Le résultat de l'exécution de ce programme est le code HTML retourné par le serveur web. Dans le contexte du Data Science, la collection des données periodique à partir d'un site distant serais intéressante pour créer un DataSet.

Idéalement le résultat serais structuré sous forme Json ou XML. La méthode *json(),* de l'objet retourné comme réponse, permet de transformer le résultat reçu en objet dictionnaire qui a une syntaxe et structure identique à l'objet JSON. En outre, l'attribut *status_code* peut être utilisé pour accéder au code réponse HTTP:

```
1 import requests
2
3 webpage = requests.get('https://googlechromelabs.github.io/chrome-for-testing
  /last-known-good-versions-with-downloads.json')
```

```
print(webpage.status_code)
print(webpage.json())
```

```
200
{'timestamp': '2024-07-24T11:08:54.116Z', 'channels': {'Stable': {'channel':
'Stable', 'version': '127.0.6533.72', 'revision': '1313161', 'downloads': {'chrome':
[{'platform': 'linux64', 'url': 'https://storage.googleapis.com/chrome-for-testing-
public/127.0.6533.72/linux64/chrome-linux64.zip'}, {'platform': 'mac-arm64', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/127.0.6533.72/mac-arm64
/chrome-mac-arm64.zip'}, {'platform': 'mac-x64', 'url': 'https://storage.googleapis.
com/chrome-for-testing-public/127.0.6533.72/mac-x64/chrome-mac-x64.zip'},
{'platform': 'win32', 'url': 'https://storage.googleapis.com/chrome-for-testing-public
/127.0.6533.72/win32/chrome-win32.zip'}, {'platform': 'win64', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/127.0.6533.72/win64/chrome-
win64.zip'}], 'chromedriver': [{'platform': 'linux64', 'url': 'https://storage.
googleapis.com/chrome-for-testing-public/127.0.6533.72/linux64/chromedriver-linux64.
zip'}, {'platform': 'mac-arm64', 'url': 'https://storage.googleapis.com/chrome-for-
testing-public/127.0.6533.72/mac-arm64/chromedriver-mac-arm64.zip'}, {'platform':
'mac-x64', 'url': 'https://storage.googleapis.com/chrome-for-testing-public/127.
0.6533.72/mac-x64/chromedriver-mac-x64.zip'}, {'platform': 'win32', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/127.0.6533.72/win32
/chromedriver-win32.zip'}, {'platform': 'win64', 'url': 'https://storage.googleapis.
com/chrome-for-testing-public/127.0.6533.72/win64/chromedriver-win64.zip'}], 'chrome-
headless-shell': [{'platform': 'linux64', 'url': 'https://storage.googleapis.com
/chrome-for-testing-public/127.0.6533.72/linux64/chrome-headless-shell-linux64.zip'},
{'platform': 'mac-arm64', 'url': 'https://storage.googleapis.com/chrome-for-testing-
public/127.0.6533.72/mac-arm64/chrome-headless-shell-mac-arm64.zip'}, {'platform':
'mac-x64', 'url': 'https://storage.googleapis.com/chrome-for-testing-public/127.
0.6533.72/mac-x64/chrome-headless-shell-mac-x64.zip'}, {'platform': 'win32', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/127.0.6533.72/win32/chrome-
headless-shell-win32.zip'}, {'platform': 'win64', 'url': 'https://storage.googleapis.
com/chrome-for-testing-public/127.0.6533.72/win64/chrome-headless-shell-win64.
zip'}]}}, 'Beta': {'channel': 'Beta', 'version': '127.0.6533.57', 'revision':
'1313161', 'downloads': {'chrome': [{'platform': 'linux64', 'url': 'https://storage.
googleapis.com/chrome-for-testing-public/127.0.6533.57/linux64/chrome-linux64.zip'},
{'platform': 'mac-arm64', 'url': 'https://storage.googleapis.com/chrome-for-testing-
public/127.0.6533.57/mac-arm64/chrome-mac-arm64.zip'}, {'platform': 'mac-x64', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/127.0.6533.57/mac-x64
/chrome-mac-x64.zip'}, {'platform': 'win32', 'url': 'https://storage.googleapis.com
/chrome-for-testing-public/127.0.6533.57/win32/chrome-win32.zip'}, {'platform':
'win64', 'url': 'https://storage.googleapis.com/chrome-for-testing-public/127.
0.6533.57/win64/chrome-win64.zip'}], 'chromedriver': [{'platform': 'linux64', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/127.0.6533.57/linux64
/chromedriver-linux64.zip'}, {'platform': 'mac-arm64', 'url': 'https://storage.
googleapis.com/chrome-for-testing-public/127.0.6533.57/mac-arm64/chromedriver-mac-
arm64.zip'}, {'platform': 'mac-x64', 'url': 'https://storage.googleapis.com/chrome-
for-testing-public/127.0.6533.57/mac-x64/chromedriver-mac-x64.zip'}, {'platform':
'win32', 'url': 'https://storage.googleapis.com/chrome-for-testing-public/127.
0.6533.57/win32/chromedriver-win32.zip'}, {'platform': 'win64', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/127.0.6533.57/win64
/chromedriver-win64.zip'}], 'chrome-headless-shell': [{'platform': 'linux64', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/127.0.6533.57/linux64
/chrome-headless-shell-linux64.zip'}, {'platform': 'mac-arm64', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/127.0.6533.57/mac-arm64
/chrome-headless-shell-mac-arm64.zip'}, {'platform': 'mac-x64', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/127.0.6533.57/mac-x64
/chrome-headless-shell-mac-x64.zip'}, {'platform': 'win32', 'url': 'https://storage.
googleapis.com/chrome-for-testing-public/127.0.6533.57/win32/chrome-headless-shell-
win32.zip'}, {'platform': 'win64', 'url': 'https://storage.googleapis.com/chrome-for-
testing-public/127.0.6533.57/win64/chrome-headless-shell-win64.zip'}]}}, 'Dev':
{'channel': 'Dev', 'version': '128.0.6601.2', 'revision': '1328598', 'downloads':
{'chrome': [{'platform': 'linux64', 'url': 'https://storage.googleapis.com/chrome-for-
testing-public/128.0.6601.2/linux64/chrome-linux64.zip'}, {'platform': 'mac-arm64',
'url': 'https://storage.googleapis.com/chrome-for-testing-public/128.0.6601.2/mac-
arm64/chrome-mac-arm64.zip'}, {'platform': 'mac-x64', 'url': 'https://storage.
googleapis.com/chrome-for-testing-public/128.0.6601.2/mac-x64/chrome-mac-x64.zip'},
{'platform': 'win32', 'url': 'https://storage.googleapis.com/chrome-for-testing-public
/128.0.6601.2/win32/chrome-win32.zip'}, {'platform': 'win64', 'url': 'https://storage.
googleapis.com/chrome-for-testing-public/128.0.6601.2/win64/chrome-win64.zip'}],
'chromedriver': [{'platform': 'linux64', 'url': 'https://storage.googleapis.com
/chrome-for-testing-public/128.0.6601.2/linux64/chromedriver-linux64.zip'},
{'platform': 'mac-arm64', 'url': 'https://storage.googleapis.com/chrome-for-testing-
public/128.0.6601.2/mac-arm64/chromedriver-mac-arm64.zip'}, {'platform': 'mac-x64',
'url': 'https://storage.googleapis.com/chrome-for-testing-public/128.0.6601.2/mac-x64
/chromedriver-mac-x64.zip'}, {'platform': 'win32', 'url': 'https://storage.googleapis.
com/chrome-for-testing-public/128.0.6601.2/win32/chromedriver-win32.zip'},
{'platform': 'win64', 'url': 'https://storage.googleapis.com/chrome-for-testing-public
/128.0.6601.2/win64/chromedriver-win64.zip'}], 'chrome-headless-shell': [{'platform':
'linux64', 'url': 'https://storage.googleapis.com/chrome-for-testing-public/128.
0.6601.2/linux64/chrome-headless-shell-linux64.zip'}, {'platform': 'mac-arm64',
'url': 'https://storage.googleapis.com/chrome-for-testing-public/128.0.6601.2/mac-
arm64/chrome-headless-shell-mac-arm64.zip'}, {'platform': 'mac-x64', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/128.0.6601.2/mac-x64/chrome-
headless-shell-mac-x64.zip'}, {'platform': 'win32', 'url': 'https://storage.
googleapis.com/chrome-for-testing-public/128.0.6601.2/win32/chrome-headless-shell-
```

```
win32.zip'}, {'platform': 'win64', 'url': 'https://storage.googleapis.com/chrome-for-
testing-public/128.0.6601.2/win64/chrome-headless-shell-win64.zip'}]}}, 'Canary':
{'channel': 'Canary', 'version': '129.0.6614.0', 'revision': '1332009', 'downloads':
{'chrome': [{'platform': 'linux64', 'url': 'https://storage.googleapis.com/chrome-for-
testing-public/129.0.6614.0/linux64/chrome-linux64.zip'}, {'platform': 'mac-arm64',
'url': 'https://storage.googleapis.com/chrome-for-testing-public/129.0.6614.0/mac-
arm64/chrome-mac-arm64.zip'}, {'platform': 'mac-x64', 'url': 'https://storage.
googleapis.com/chrome-for-testing-public/129.0.6614.0/mac-x64/chrome-mac-x64.zip'},
{'platform': 'win32', 'url': 'https://storage.googleapis.com/chrome-for-testing-public
/129.0.6614.0/win32/chrome-win32.zip'}, {'platform': 'win64', 'url': 'https://storage.
googleapis.com/chrome-for-testing-public/129.0.6614.0/win64/chrome-win64.zip'}],
'chromedriver': [{'platform': 'linux64', 'url': 'https://storage.googleapis.com
/chrome-for-testing-public/129.0.6614.0/linux64/chromedriver-linux64.zip'},
{'platform': 'mac-arm64', 'url': 'https://storage.googleapis.com/chrome-for-testing-
public/129.0.6614.0/mac-arm64/chromedriver-mac-arm64.zip'}, {'platform': 'mac-x64',
'url': 'https://storage.googleapis.com/chrome-for-testing-public/129.0.6614.0/mac-x64
/chromedriver-mac-x64.zip'}, {'platform': 'win32', 'url': 'https://storage.googleapis.
com/chrome-for-testing-public/129.0.6614.0/win32/chromedriver-win32.zip'},
{'platform': 'win64', 'url': 'https://storage.googleapis.com/chrome-for-testing-public
/129.0.6614.0/win64/chromedriver-win64.zip'}], 'chrome-headless-shell': [{'platform':
'linux64', 'url': 'https://storage.googleapis.com/chrome-for-testing-public/129.
0.6614.0/linux64/chrome-headless-shell-linux64.zip'}, {'platform': 'mac-arm64',
'url': 'https://storage.googleapis.com/chrome-for-testing-public/129.0.6614.0/mac-
arm64/chrome-headless-shell-mac-arm64.zip'}, {'platform': 'mac-x64', 'url':
'https://storage.googleapis.com/chrome-for-testing-public/129.0.6614.0/mac-x64/chrome-
headless-shell-mac-x64.zip'}, {'platform': 'win32', 'url': 'https://storage.
googleapis.com/chrome-for-testing-public/129.0.6614.0/win32/chrome-headless-shell-
win32.zip'}, {'platform': 'win64', 'url': 'https://storage.googleapis.com/chrome-for-
testing-public/129.0.6614.0/win64/chrome-headless-shell-win64.zip'}]}}}}
```

Comme déclaré initialement, il est préférable de transmettre des requêtes sur des intervalles régulières. Pour ceci, la fonction *sleep()* du module *time* est utilisée. Par exemple, le code suivant illustre comment transmettre une requête à l'url précédant une fois par 24 heurs. La fonction *time.sleep()* prend le temps d'attente en secondes :

```
1 import requests
2 import time
3 while True:
4     webpage = requests.get('https://googlechromelabs.github.io/chrome-for-testing
  /last-known-good-versions-with-downloads.json')
5     print(webpage.status_code)
6     print(webpage.json())
7     time.sleep(24*60*60)
```

Dans les deux premiers exemples, nous avons utiliser *.text* et *.json()* de l'objet réponse afin d'accéder /transformer le contenu reçu. En cas où le résultat reçu n'est pas un contenu textuel (image, fichier exe, etc), l'attribut *.content* est utilisé afin d'accéder à la suite d'octets brute. La valeur détenue par cette attribut est de type *bytes*.

De même, l'attribut *.headers* du résultat contient les entêtes renvoyés par le serveur (Content-type, Last-Modified, etc) sous forme de dictionnaire :

```
1 import requests
2 webpage = requests.get('https://www.licbplus.com.dz/')
3 print(webpage.headers)
4 print(webpage.headers['Content-Type'])
```

```
1 {'Connection': 'Keep-Alive', 'Keep-Alive': 'timeout=5, max=100', 'content-type':
  'text/html; charset=UTF-8', 'link': '<https://www.licbplus.com.dz/wp-json/>; rel="
  https://api.w.org/", <https://www.licbplus.com.dz/wp-json/wp/v2/pages/36>; rel="
  alternate"; type="application/json", <https://www.licbplus.com.dz/>; rel=shortlink',
  'etag': '"600708-1721822742;br"', 'x-litespeed-cache': 'hit', 'content-encoding':
  'br', 'vary': 'Accept-Encoding', 'content-length': '36888', 'date': 'Wed, 24 Jul 2024
  16:51:41 GMT', 'server': 'DZ-HTTP'}
2 text/html; charset=UTF-8
3 {'Connection': 'Keep-Alive', 'Keep-Alive': 'timeout=5, max=100', 'content-type':
  'text/html; charset=UTF-8', 'link': '<https://www.licbplus.com.dz/wp-json/>; rel="
  https://api.w.org/", <https://www.licbplus.com.dz/wp-json/wp/v2/pages/36>; rel="
  alternate"; type="application/json", <https://www.licbplus.com.dz/>; rel=shortlink',
  'etag': '"600708-1721822742;br"', 'x-litespeed-cache': 'hit', 'content-encoding':
  'br', 'vary': 'Accept-Encoding', 'content-length': '36888', 'date': 'Wed, 24 Jul 2024
  16:51:41 GMT', 'server': 'DZ-HTTP'}
```

```
text/html; charset=UTF-8
```

On peux aussi spécifier les entêtes de la requête transmis au serveur par le paramètre optionnel *headers :*

```
import requests
import time

webpage = requests.get('https://googlechromelabs.github.io/chrome-for-testing
/last-known-good-versions-with-downloads.json',
                       headers={'Accept': 'application/json'})
print(webpage.status_code)
print(webpage.json())
```

Le module *requests* propose une fonction pour chaque méthode HTTP (GET, HEAD, POST,PUT, DELETE, OPTIONS). Généralement, s'il est nécessaire de transmettre des données au serveur (comme la clé d'authentification ou le contenu d'un formulaire) pour recevoir le résultat, ces derniers sont exprimés dans le corps (*body*) de la requête au lieu de l'url et la méthode POST est utilisée. Le paramètre optionnel *data* est utilisé pour exprimer les données à transmettre sous forme de dictionnaire, liste de tuples, suite d'octets ou un objet fichier :

```
import requests
objDict= {'somekey': 'somevalue'}
resultat = requests.post('https://www.exemple.API.com/', data= objDict, headers= {
"Cookie":"=XKJHQSF21924JSAQF"})
print(resultat .text)
```

Remarque : API

Une page web qui retourne un résultat sous forme JSON/XML (contenu semi-structurée) est appelée *API (Application Programming Interface)*. Par exemple l'API suivante vous renvoie des informations de localisation géographique sur une adresse IP donnée (161.185.160.93 dans l'exemple)

https://ipinfo.io/161.185.160.93/geo

https://ipinfo.io/geo (sans le segment contenant l'adresse IP), l'API renvoie des informations sur la machine qui a générer la requête.

BeautifulSoup

II

Le traitement des données semi-structurées JSON ou XML est facile généralement. Par contre, il est plus difficile de récupérer les données à partir d'un contenu HTML. Par ailleurs, si la structure de la page change, il faudra adapter le code afin d'accéder aux informations ciblées. *BeautifulSoup* est une bibliothèque qui facilite le traitement d'un document HTML pour récupérer le contenu cible. Supposons que nous sommes intéressés par les prix des PC bureau proposés par licbplus.com. Pour commencer, il faudra inspecter le code HTML/ structure de la page. La grille des produits est présentée dans un élément *div* définit avec la classe *product-grid* et id *product-list*. Le prix de chaque article est présenté dans une balise div avec une classe *product-price.*



Avant de commencer à utiliser *BeautifulSoup*, vous devez installer le module :

```
pip install beautifulsoup4
```

BeautifulSoup utilise les concepts de navigation DOM (Document Object Model) HTML qui permettent la sélection des éléments selon l'identifiant, classe css et attributs. Pour commencer, il est nécessaire d'importer le module *BeautifulSoup* et puis de parser le contenu HTML. Le type de paramètre à parser doit être spécifier dans le deuxième argument qui permet aussi de parser d'autres types de contenu. Par la suite, le résultat retourné est similaire à l'objet *document* en Javascript qui permet la sélection d'éléments dans le document. Le code suivant utilise la méthode *.select()* pour retrouver tout élément *span* contenu dans un élément *div* possédant la classe *product-card-bottom*.

Tout de même, les éléments retournés sont toujours des éléments HTML. Pour accéder au texte contenu dans chaque élément, il suffit d'utiliser l'attribut *.text* (équivalente à *.innerText* in JavaScript):

```
import requests
from bs4 import BeautifulSoup
```

```
webpage = requests.get('https://www.licbplus.com.dz/global-category/desktop-computers')
bs = BeautifulSoup(webpage.content, "html.parser")
result=bs.select("div.product-card-bottom span")
print(result)
print([div.text for div in result])
```

Il est possible d'accéder aux attributs de chaque élément retourné par la méthode *.select()* avec la méthode *.get()*. Le code suivant sélectionne les images contenues dans la page, puis pour chaque image *.get('src')* renvoie le chemin vers l'image cible.

```
import requests
from bs4 import BeautifulSoup
webpage = requests.get('https://www.licbplus.com.dz/global-category/desktop-computers')
bs = BeautifulSoup(webpage.content, "html.parser")
images_produit=bs.select("img")
print(*map(lambda img:img.get("src"),images_produit))

```

La méthode *.select()* admet comme paramètre n'importe quel sélecteur *css* valide comme *document.querySelectorAll()* en Javascript. Le code suivant sectionne les éléments ayant un attribut *class="fi-rs-eye"* .

```
import requests
from bs4 import BeautifulSoup
reponse=requests.get('https://www.licbplus.com.dz/global-category/desktop-computers')
document = BeautifulSoup(reponse.content, "html.parser")
resultat=document.select('[class="fi-rs-eye"]')
print(resultat)
```

```
['Simons, retour au point de départ ?',
 "Nouveau rebondissement concernant le cas Xavi Simons. Joueur sous contrat avec le Paris Saint-Germain, l'international néerlandais est intéressé par un nouveau prêt, après celui de la saison passée à Leipzig. Selon les derniers échos relayés par Fabrizio Romano, le Batave serait finalement proche... d'un retour à Leipzig ! Le club allemand ferait le forcing pour obtenir un nouveau prêt. Une possibilité qui pourrait convaincre Paris.",
 '',
 'Yvon Mvogo vers Bournemouth\xa0?',
 'Sous contrat jusqu'en juin 2025 avec Lorient, le portier Yvon Mvogo pourrait rebondir du côté de Bournemouth en Premier League selon l'Equipe. Les deux clubs partagent le même propriétaire ce qui pourrait faciliter la transaction. Mvogo bénéficie d'un bon de sortie accordé par son président Loïc Féry.',
 '',
 'Deuxième offre du Bayern pour Désiré Doué',
 "Le Bayern Munich vient de proposer environ 55 millions d'euros bonus compris à Rennes pour recruter Doué selon l'Equipe. Une offre qui pourrait satisfaire les Rennais qui en réclame un peu près 60 millions. Le PSG devrait revenir à la charge et proposer une nouvelle offre à la hauteur du club allemand.",
 '',
 'Morgan Guilavogui en prêt à Sankt Pauli',
 'L'attaquant Morgan Guilavogui va être prêté avec option d'achat par Lens à Sankt Pauli, tout juste promu en Bundesliga selon l'Equipe.',
 '',
 'Brentford va vendre Ivan Toney',
 'Selon The Athletic, Brentford va vendre Ivan Toney lors de ce mercato, à un an de la fin de son contrat. Les Bees préfèrent recevoir une indemnité de transfert cet été plutôt que de le voir partir gratuitement l'année prochaine.',
 '',
 'Assignon pisté par Aston Villa',
 'Le latéral droit du Stade Rennais, Lorenz Assignon, prêté la saison dernière à Burnely, intéresse le club anglais d'Aston Villa selon Foot Mercato. Rennes attend au moins 10 millions d'euros pour lâcher son joueur de 24 ans, aussi ciblé par l'AS Rome.',
```

```
 '',
 'Sarr d'accord avec Crystal Palace',
 'Selon The Athletic, Ismaila Sarr s'est mis d'accord avec Crystal Palace pour rejoindre le club. Un contrat de quatre ou cinq ans l'attend à Londres. Palace doit maintenant négocier avec Marseille.',
 '',
 'Le PSG explore la piste Sancho',
 'Selon Sky Sports, le PSG n'a pas encore proposé d'offre à Manchester United pour s'attacher les services de Jadon Sancho. Le club de la capitale étudie pour le moment le dossier. La Juventus, Dortmund ainsi qu'un autre club de Premier League sont également sur le coup.',
 '',
 'Séville veut Ansu Fati',
 'Selon les informations de Sport, relayées par Foot Mercato, le FC Séville fait le forcing pour recruter Ansu Fati. Le Barça veut le vendre et Séville a pour le moment proposé une offre en dessous des attentes du club catalan. Les négociations se poursuivent.',
 '',
 'Le PSG pousse pour Doué et cible Williams',
 'Désireux de recruter le milieu offensif de Rennes Désiré Doué, Nasser Al-Khelaifi s'investit personnellement dans le dossier selon The Athletic. Pour le moment, il n'y a aucun accord entre le club parisien et le Stade Rennais. Parallèlement, le PSG a noué les premiers contacts avec Bilbao pour le recrutement de Nico Williams. Sous contrat jusqu'en 2027 avec le club espagnol, l'ailier dispose d'une clause libératoire d'environ 60 millions d'euros.',
 '',
 'Abdoulaye Ndiaye va signer à Brest',
 'À la recherche d'un renfort en défense pour remplacer Lilian Brassier qui est parti à l'OM, Brest va recruter le défenseur central de Troyes Abdoulaye Ndiaye. Selon Foot Mercato, la transaction est un prêt avec option d'achat de 9 millions d'euros.',
 '',
 'Marseille bientôt d'accord avec Nketiah\xa0?',
 'À la recherche d'un attaquant pour pallier le départ de Pierre-Emerick Aubameyang, l'OM est proche d'un accord contractuel avec Eddie Nketiah selon l'Equipe. Il faut encore convaincre son club d'Arsenal avec qui il est sous contrat jusqu'en 2027.',
 '',
 'Adams signe au Torino',
 'Libre depuis l'expiration de son contrat avec Southampton, Che Adams vient de s'engager librement avec le Torino. C'est la première expérience à l'étranger pour l'Écossais.',
 '',
 'L'Inter Milan lorgne Zeze',
 'Le champion d'Italie en titre l'Inter Milan a approché Nantes pour s'attacher les services de Nathan Zeze, jeune défenseur central de 19 ans selon l'Equipe. Sous contrat jusqu'en 2028 avec les Canaris, Zeze est considéré comme intransférable par la famille Kita.',
 '',
 'Chelsea sur Filip Jorgensen',
 'Chelsea a approché le portier de Villarreal Filip Jorgensen pour un éventuel transfert selon The Athletic. Le gardien de but danois est également ciblé par l'Olympique de Marseille.',
 '',
 'Savic va signer à Trabzonspor',
 'Stefan Savic va quitter librement l'Atlético de Madrid pour s'engager avec Trabzonspor selon Relevo. Il va signer un contrat jusqu'en juin 2027 en Turquie.',
 '',
 'De Bruyne va rester à Manchester City',
 'Pep Guardiola a confirmé que le milieu belge Kevin de Bruyne va rester à Manchester City cette saison malgré les rumeurs d'un potentiel transfert en Arabie saoudite.',
 '',
 'Wan-Bissaka sur le départ',
 'Selon les informations de Sky Sport Germany, Manchester United pousse pour le départ d'Aaron Wan-Bissaka. United demande environ 10 millions d'euros pour se séparer de l'Anglais. Les Red Devils ciblent Noussair Mazraoui pour le remplacer.',
 '',
```

```
55 'Bonjour à tous et bienvenue dans ce nouveau live pour suivre les dernières
   rumeurs du mercato estival',
56 'Le mercato en direct - Sancho, Varane, Todibo, PSG, OM, Real Madrid : Suivez la
   journée des transferts en direct sur Eurosport']
```

Alternativement à .select(), la recherche peut être faite avec la méthode *.find()* et *.find_all()*. Il est aussi possible de limiter le nombre de résultats retournés par *select*(), *find* et *find_all*, avec le paramètre optionnel *limit*.

Le paramètre optionnel *string* de *find_all* peut être utilisé avec une fonction qui filtre les résultats retournés. Le code suivant recherche les éléments HTML *a* et puis, avec la fonction passée dans l'argument string, seulement les éléments contenant la chaîne *"XT"* sont retournés :

```
1 import requests
2 from bs4 import BeautifulSoup
3 reponse=requests.get('https://www.licbplus.com.dz/global-category/desktop-
  computers')
4 document = BeautifulSoup(reponse.content, "html.parser")
5 resultat=document.find_all('a',string=lambda text:False if text==None else "XT" in
   text )
6 print(resultat)
7
```

```
1 [<li class="menu-item menu-item-type-custom menu-item-object-custom menu-item-
  4268" id="menu-item-4268"><a class="ecome-menu-item-title" href="https://www.licbplus.
  com/product-category/composants-pc/hdd-interne/" title="HDD interne">HDD interne</a><
  /li>,
2 <li class="menu-item menu-item-type-custom menu-item-object-custom menu-item-
  12739" id="menu-item-12739"><a class="ecome-menu-item-title" href="https://www.
  licbplus.com/product-category/composants-pc/rack-hdd-interne/" title="Rack HDD
  interne">Rack HDD interne</a></li>,
3 <li class="menu-item menu-item-type-custom menu-item-object-custom menu-item-
  4269" id="menu-item-4269"><a class="ecome-menu-item-title" href="https://www.licbplus.
  com/product-category/peripherique-pc/hdd-externe/" title="HDD externe">HDD externe<
  /a></li>]
```

Il est aussi possible de créer un DataFrame à partir d'une table HTML. Pour commencer, il faudra lire le contenu du fichier HTML contenant la table. Par la suite il suffit d'invoquer *pd.read_html()* avec le contenu textuel du document HTML comme paramètre. Le résultat est une liste des DataFrames lus à partir des tableaux inclus dans le fichier HTML.

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <h2>HTML Table</h2>
5 <table>
6   <tr>
7   <th>Company</th><th>Contact</th><th>Country</th>
8   </tr>
9   <tr>
10  <td>Alfreds Futterkiste</td><td>Maria Anders</td><td>Germany</td>
11  </tr>
12  <tr>
13    <td>Centro comercial Moctezuma</td><td>Francisco Chang</td><td>Mexico</td>
14  </tr>
15  <tr>
16    <td>Ernst Handel</td><td>Roland Mendel</td><td>Austria</td>
17  </tr>
18 </table>
19 </body>
20 </html>
```

```
import pandas as pd
table_html=open("exmple_table.html")
df=pd.read_html(table_html.read())
table_html.close()
df[0].head()
```

	Company	Contact	Country
0	Alfreds Futterkiste	Maria Anders	Germany
1	Centro comercial Moctezuma	Francisco Chang	Mexico
2	Ernst Handel	Roland Mendel	Austria

Références

[19] BeautifulSoup 2024, https://www.crummy.com/software/BeautifulSoup/bs4/doc/