



# Scraping

*Nom de filière ou programme*

*Master 1*





# Plan

- I. Web Scraping : Définition**
- II. Pourquoi Web Scraping ?**
- III. Le processus de Web Scraping**
- IV. Outils de Web Scraping**
- V. BeautifulSoup**
- VI. Protection contre le web scraping**
- VII. Exemple Web Scraping**
- VIII. L'aspect éthérique et juridiques du Web Scraping**

# Definition

- Le **web scraping**, également connu sous le nom de data scraping, est le processus d'extraction et de collecte de données sur Internet ou des sites Web. Les données récupérées peuvent être enregistrées sur un système local ou peuvent être utilisées pour l'analyse des données.



- Une fois les données extraites et stockées, elles peuvent être utilisées de plusieurs manières. Par exemple, pour trouver des informations de contact ou comparer des prix sur différents sites web.



# Definition

- Le web-scraping permet de collecter des données :
  - Automatisement : Des outils spécialisés naviguent et récupèrent les informations sans intervention humaine, réduisant les erreurs et augmentant l'efficacité.
  - Rapidement : Cette automatisation permet de traiter des pages web à une vitesse bien supérieure à celle d'une personne, ce qui est crucial pour les tâches où le temps est un facteur déterminant.
  - En grande quantité : Capable de gérer des volumes de données massifs, le web scraping est idéal pour les projets nécessitant de grandes quantités d'informations, telles que l'analyse de marché

# Pourquoi web Scraping

**1. Comparaison des prix des produits :** Les sites web de comparaison de prix peuvent utiliser le web scraping pour extraire des données de différents sites de vente en ligne et afficher les résultats aux utilisateurs

**2. Données d'IA et d'apprentissage automatique :** Le web scraping pour l'apprentissage automatique aide les data scientists à collecter les informations requises pour alimenter leurs ensembles de données.

**3. Développement d'applications :** Les développeurs peuvent utiliser le web scraping pour intégrer des données provenant de différentes sources dans leurs applications.

**4. Analyse de marché :** Les professionnels du marketing et les entreprises peuvent utiliser le web scraping pour recueillir des données sur les tendances du marché, les préférences des consommateurs et d'autres informations pertinentes.



# Pourquoi web Scraping

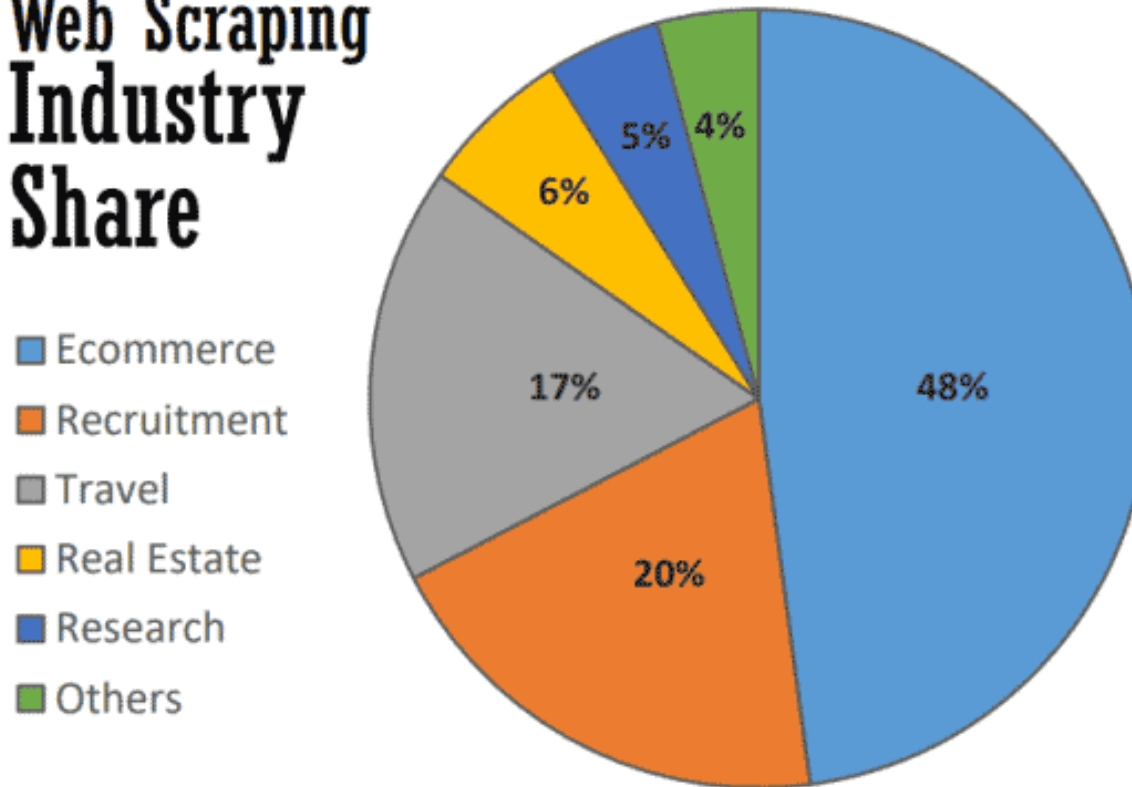
## 4. Surveillance du sentiment des consommateurs

**5. Mise à jour de contenu :** Certains sites web peuvent utiliser le web scraping pour mettre à jour automatiquement leur contenu en récupérant des informations à partir de sources externes.



# Industrie de Web Scraping

## Web Scraping Industry Share



- Dans le meilleur des cas, le web scraping sert à de nombreuses fins utiles dans de nombreux secteurs.
- En 2021, près de la moitié de l'ensemble de l'extraction web est utilisé pour soutenir les stratégies de commerce électronique.

## Most scraped Websites

- E-commerce

amazon

ebay

Google

- Offres de travail

indeed

LinkedIn

- Réseaux Sociaux



# Processus de base du web scraping

1. **Identification de la cible** : Identifiez les pages spécifiques ou les sections du site contenant les données que vous souhaitez extraire.
2. **Analyse de la structure du site** : Identifiez les balises HTML, les classes, les identifiants et les modèles qui entourent les données que vous souhaitez extraire.
3. **Sélection de l'outil de scraping** : Utilisez les bibliothèques de scraping pour naviguer à travers la structure HTML du site et extraire les informations nécessaires.
4. **Exécution du script de scraping** : Écrivez un script pour accéder au site web et extraire les données souhaitées, puis lancez le script pour collecter les données à partir du site web cible.
5. **Stockage des données** : stocker les données extraites dans un fichier local, une base de données, ou même un service de stockage cloud.

# Analyse de la structure du site

- Dans le code HTML d'une page web, les informations sont associées à différents types de balises :
  - **title** renseigne le titre de la page
  - **h1** à **h6** composent les titres et sous-titres du contenu
  - **img** est une balise associée aux images
  - **a** permet d'insérer un lien hypertexte
  - **table** est la balise associée aux tableaux
  - **span** et **div** sont des balises génériques qui peuvent être associées à une classe spécifique (ce qui permet de rendre la balise unique).



# Pouvez-vous extraire tous les sites Web ?

Le scraping entraîne une augmentation du trafic sur le site Web et peut provoquer une panne du serveur du site Web.

Ainsi, tous les sites Web ne permettent pas aux utilisateurs d'extraire leurs données.

Comment savoir quels sites Web sont autorisés ou non ?

---

# Robots.txt

- Vous pouvez consulter le fichier « **robots.txt** » du site Web.
- Il vous suffit de mettre robots.txt après l'URL que vous souhaitez récupérer et vous verrez des informations indiquant si l'hébergeur du site Web vous autorise à supprimer le site Web.
- Prenons **Google.com** comme exemple :

← → ↻ google.com/robots.txt

```
User-agent: *  
Disallow: /search  
Allow: /search/about  
Allow: /search/static  
Allow: /search/howsearchworks  
Disallow: /sdch  
Disallow: /groups  
Disallow: /index.html?  
Disallow: /?  
Allow: /?hl=  
Disallow: /?hl=*  
Allow: /?hl=*&gws_rd=ssl$  
Disallow: /?hl=*&gws_rd=ssl  
Allow: /?gws_rd=ssl$  
Allow: /?ptl=true$  
Disallow: /imgres  
Disallow: /u/  
Disallow: /preferences  
Disallow: /setprefs  
Disallow: /default  
Disallow: /m?  
Disallow: /m/  
Allow: /m/finance  
Disallow: /wml?  
Disallow: /wml/?  
Disallow: /wml/search?  
Disallow: /xhtml?  
Disallow: /xhtml/?  
Disallow: /xhtml/search?  
Disallow: /xml?  
Disallow: /imode?  
Disallow: /imode/?  
Disallow: /imode/search?  
Disallow: /jsky?  
Disallow: /jsky/?  
Disallow: /jsky/search?  
Disallow: /pda?  
Disallow: /pda/?  
Disallow: /pda/search?  
Disallow: /sprint_xhtml  
Disallow: /sprint_wml  
Disallow: /pqa
```

Vous pouvez voir que Google n'autorise pas le web scraping pour bon nombre de ses sous-sites. Cependant, il autorise certains chemins comme « `/m/finance` » et donc si vous souhaitez collecter des informations sur la finance, c'est un endroit tout à fait légal à gratter.

# L'aspect éthérique et juridiques du Web Scraping

- **Consentement** : Le scraping de sites web sans consentement peut violer les termes et conditions de ces sites, ce qui pose des questions d'éthique sur le respect de la propriété et des règles établies par les propriétaires de sites web.
- **Respect de la vie privée** : La collecte de données personnelles sans consentement des individus peut enfreindre les lois sur la protection de la vie privée, comme le Règlement général sur la protection des données (RGPD) en Europe.



# L'aspect éthérique et juridiques du Web Scraping

- **Charge sur les ressources** : Un scraping intensif peut surcharger les serveurs des sites web, affectant potentiellement leur fonctionnement et leur accessibilité pour les autres utilisateurs.
- **Utilisation des données** : L'utilisation des données collectées pour des fins malveillantes, comme le spamming ou la concurrence déloyale, est également un aspect critique.
- **Propriété intellectuelle** : Le contenu des sites web peut être protégé par des droits d'auteur, et sa réutilisation sans autorisation peut constituer une infraction.

# Avantages de web scraping

- **Gain de temps** : Lorsque vous utilisez le web scraping, vous n'avez pas besoin de collecter manuellement les données des sites Web et vous pouvez rapidement supprimer plusieurs sites Web en même temps.
- **Données à grande échelle** : le Web scraping vous fournit des données dans un volume bien supérieur à celui que vous pourriez jamais collecter manuellement.
- **Rentable** : un simple grattoir peut souvent faire l'affaire, vous n'avez donc pas besoin d'investir dans des systèmes complexes ou du personnel supplémentaire
- **Modifiable** : créez un grattoir pour une tâche et vous pouvez souvent le modifier pour une tâche différente en n'apportant que de petites modifications.
- **Précis et robuste** : configurez correctement votre scraper et il collectera avec précision les données directement à partir des sites Web, avec un très faible risque d'introduction d'erreurs.

## Outils de Web Scraping

BeautifulSoup



Scrapy



# Outils de Web Scraping

- **Beautiful Soup** : Une bibliothèque Python qui facilite l'extraction d'informations à partir de fichiers HTML et XML. Elle est souvent utilisée conjointement avec la bibliothèque requests pour effectuer des requêtes HTTP.
- **Requests** : Bien que principalement une bibliothèque Python pour effectuer des requêtes HTTP, elle est souvent utilisée en conjonction avec BeautifulSoup pour extraire des données de pages web.
- **Scrapy** : Un framework Python open source dédié au web scraping. Il offre une architecture robuste pour le scraping de sites web de manière structurée et extensible.
- **Selenium** : Une suite d'outils pour automatiser les navigateurs web. Il est souvent utilisé pour le scraping de sites web qui utilisent JavaScript pour générer leur contenu, car il permet de simuler l'interaction avec un navigateur.

BeautifulSoup



Requests



Scrapy



# Outils de Web Scraping

- **Octoparse** : Octoparse est un logiciel de web scraping, qui permet d'extraire un volume important de données sur le web. Cet **outil no code** offre la possibilité de transformer les pages web en feuilles de calcul structurées, sans avoir besoin de connaissance technique.
- **ParseHub** : ParseHub est un outil de web scraping permettant aux professionnels d'extraire des données, même complexes, à partir de pages web, sans avoir besoin de savoir coder. Les données sont collectées à partir de l'interface desktop et exportées dans des feuilles de calcul structurées.
- ParseHub et Octoparse sont deux outils similaires dans le sens où ils sont tous deux des plateformes de web scraping visuel, permettant aux utilisateurs d'extraire des données à partir de sites web **sans nécessiter de compétences avancées en programmation.**



# Les basiques BeautifulSoup

## 1. Récupérer la page web

```
url = "https://www.mon-site.com"  
response = requests.get(url)
```

## 2. Analyser le contenu HTML : Utilisez BeautifulSoup pour analyser le contenu HTML de la page.

```
soup = BeautifulSoup(response.text, 'html.parser')
```

## 3. Trouver les éléments cibles : Utilisez les méthodes de BeautifulSoup pour trouver les éléments HTML que vous souhaitez extraire.

```
links = soup.find_all('a', class_='article-link')
```

# Les basiques BeautifulSoup

```
soup.title
```

résultat



```
<title>Ecole Supérieure de Technologie - Fkih Ben Salah</title>
```

- Si vous n'avez pas besoin de l'élément complet, mais seulement du texte, vous pouvez également le faire avec `get_text()`:

```
soup.title.get_text()
```

résultat

```
'Ecole Supérieure de Technologie - Fkih Ben Salah'
```

- 
- Et si vous n'avez besoin que de l'attribut d'un élément ? Aucun problème :

```
soup.a["href"]
```

```
book.a.get("href")
```

# Common Anti-Scraping Techniques



Setting Up  
Robots.txt



Filtering Requests  
By User Agent



Blacklisting IP  
Addresses



Showing A Captcha



Honeypots

# Protection contre le web scraping

- **Blocage des adresses IP** : De nombreux hébergeurs web gardent la trace des adresses IP de leurs visiteurs. Si un hébergeur remarque qu'un visiteur particulier génère de nombreuses requêtes de serveur (comme dans le cas de certains extracteurs de sites web ou robots), il peut alors bloquer entièrement l'IP.

## Common Anti-Scraping Techniques



Setting Up  
Robots.txt



Filtering Requests  
By User Agent



Blacklisting IP  
Addresses



Showing A Captcha



Honeypots

# Protection contre le web scraping

- **Configuration de robots.txt** : Un fichier robots.txt permet à un hébergeur web d'indiquer aux extracteurs, aux analyseurs et aux autres robots ce à quoi ils peuvent ou non accéder. Par exemple, certains sites web utilisent un fichier robots.txt pour rester privés en indiquant aux moteurs de recherche de ne pas les indexer. Si la plupart des moteurs de recherche respectent ces fichiers, ce n'est pas le cas de nombreuses formes malveillantes d'extracteurs web.

## Common Anti-Scraping Techniques



Setting Up  
Robots.txt



Filtering Requests  
By User Agent



Blacklisting IP  
Addresses



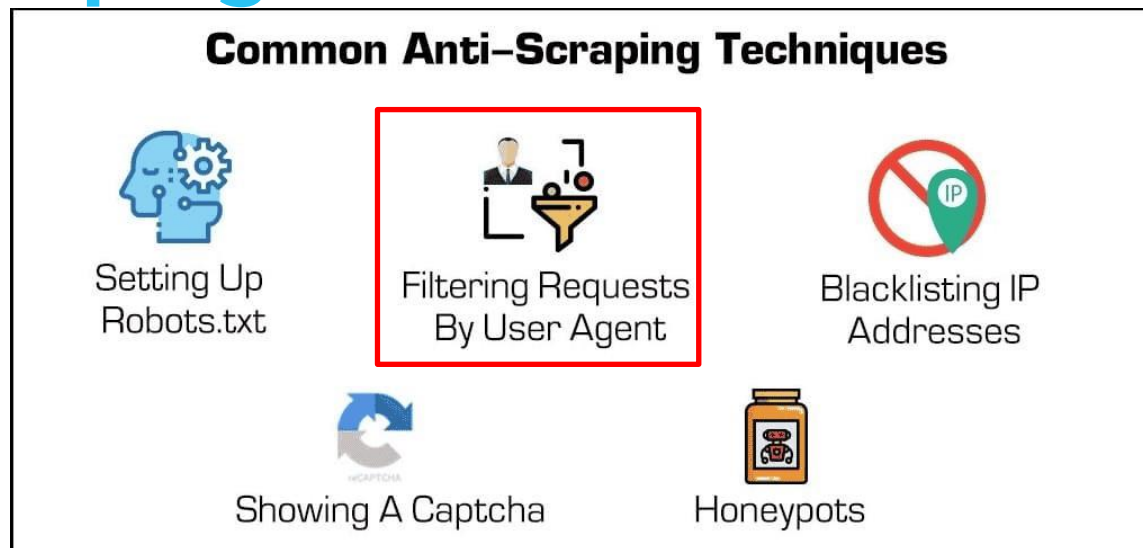
Showing A Captcha



Honeypots

# Protection contre le web scraping

- **Filtrage des requêtes** : Lorsqu'une personne visite un site web, elle « demande » une page HTML au serveur web. Ces requêtes sont souvent visibles pour les hébergeurs web, qui peuvent voir certains facteurs d'identification tels que les adresses IP et les agents utilisateurs comme les navigateurs web. Nous avons déjà abordé le blocage des IP, mais les hébergeurs web peuvent également filtrer par agent utilisateur.



# Protection contre le web scraping

- **Afficher un Captcha** : Avez-vous déjà dû saisir une étrange chaîne de texte ou cliquer sur au moins six voiliers avant d'accéder à une page ? Alors vous avez rencontré un Captcha. Bien qu'ils soient simples, ils sont incroyablement efficaces pour filtrer les extracteurs web et autres robots.

## Common Anti-Scraping Techniques



Setting Up  
Robots.txt



Filtering Requests  
By User Agent



Blacklisting IP  
Addresses



Showing A Captcha



Honeypots

# Protection contre le web scraping

- **Honeypots** : Un honeypot ou pot de miel est un type de piège utilisé pour attirer et identifier les visiteurs indésirables. Dans le cas des extracteurs web, un hébergeur web peut inclure des liens invisibles sur sa page web. Les utilisateurs humains ne s'en apercevront pas, mais les robots les visiteront automatiquement en les faisant défiler, ce qui permettra aux hébergeurs web de collecter (et de bloquer) leurs adresses IP ou leurs agents utilisateurs.

## Common Anti-Scraping Techniques



Setting Up  
Robots.txt



Filtering Requests  
By User Agent



Blacklisting IP  
Addresses



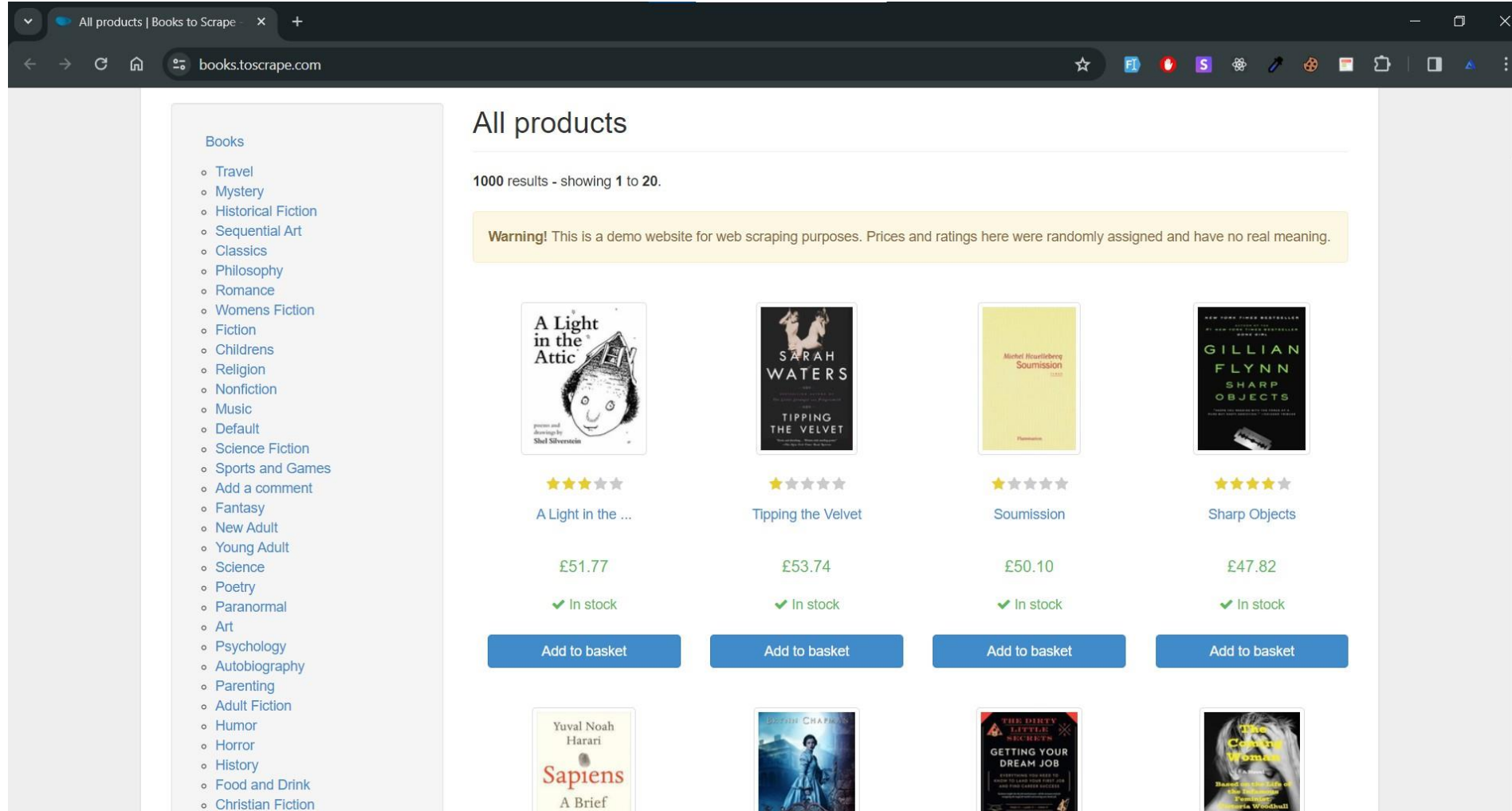
Showing A Captcha



Honeypots

# <https://books.toscrape.com/>

- C'est un endroit sûr pour les débutants qui apprennent le web scraping et pour les développeurs qui valident également leurs technologies de scraping.



# Grandes lignes du projet

- Nous allons récupérer les informations suivantes sur notre site <https://books.toscrape.com/>
  - Les titres de livres
  - Les prix
  - La disponibilité des stocks
  - Lien pour obtenir chaque livre
- Après avoir collecté les informations, nous allons les stocker dans un cadre de données Pandas et les convertir en fichier CSV pour un accès facile et une analyse plus approfondie.



# Partie Pratique

# isoler les éléments qui no

```
import requests
from bs4 import BeautifulSoup
import csv
import pandas as pd
import os
```

✓ 9.8s

```
URL = 'https://books.toscrape.com/'
response = requests.get(URL)
response
```

✓ 0.9s

<Response [200]>

oses. Prices and ratings here were randomly

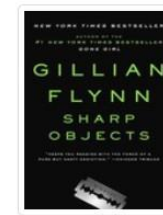


Soumission

£50.10

✓ In stock

Add to basket

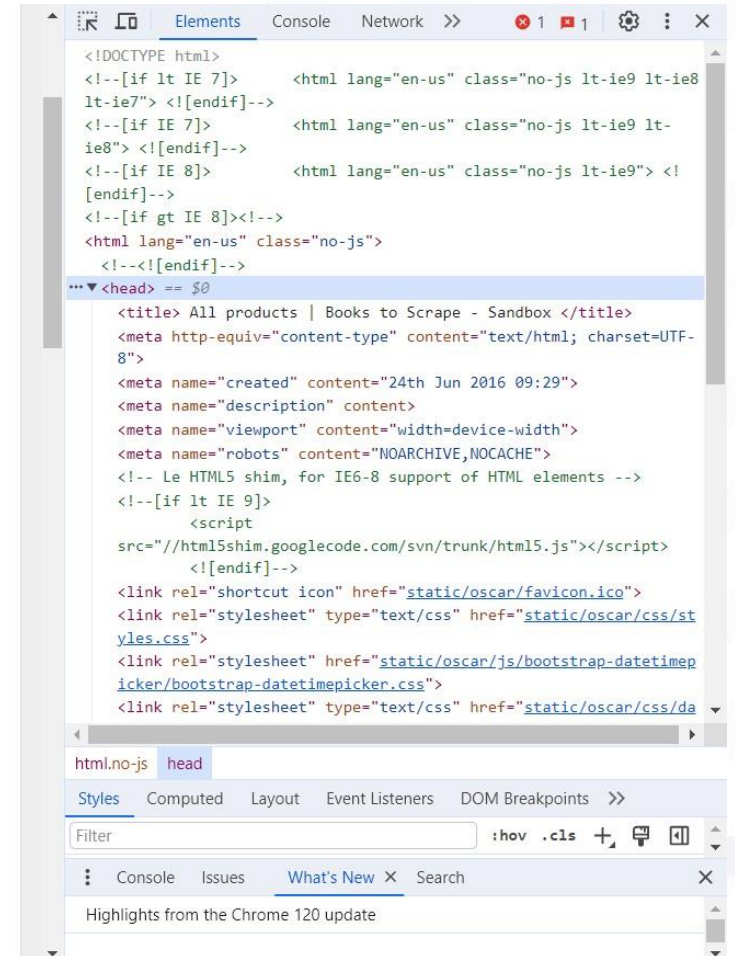


Sharp Objects

£47.82

✓ In stock

Add to basket



```
page_contents = response.text
page_contents
```

[7] ✓ 0.0s

```
'<!DOCTYPE html>\n<!--[if lt IE 7]> <html lang="en-us" class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->\n<!--[if IE 7]>
```

# Utilisez BeautifulSoup pour analyser et extraire les informations

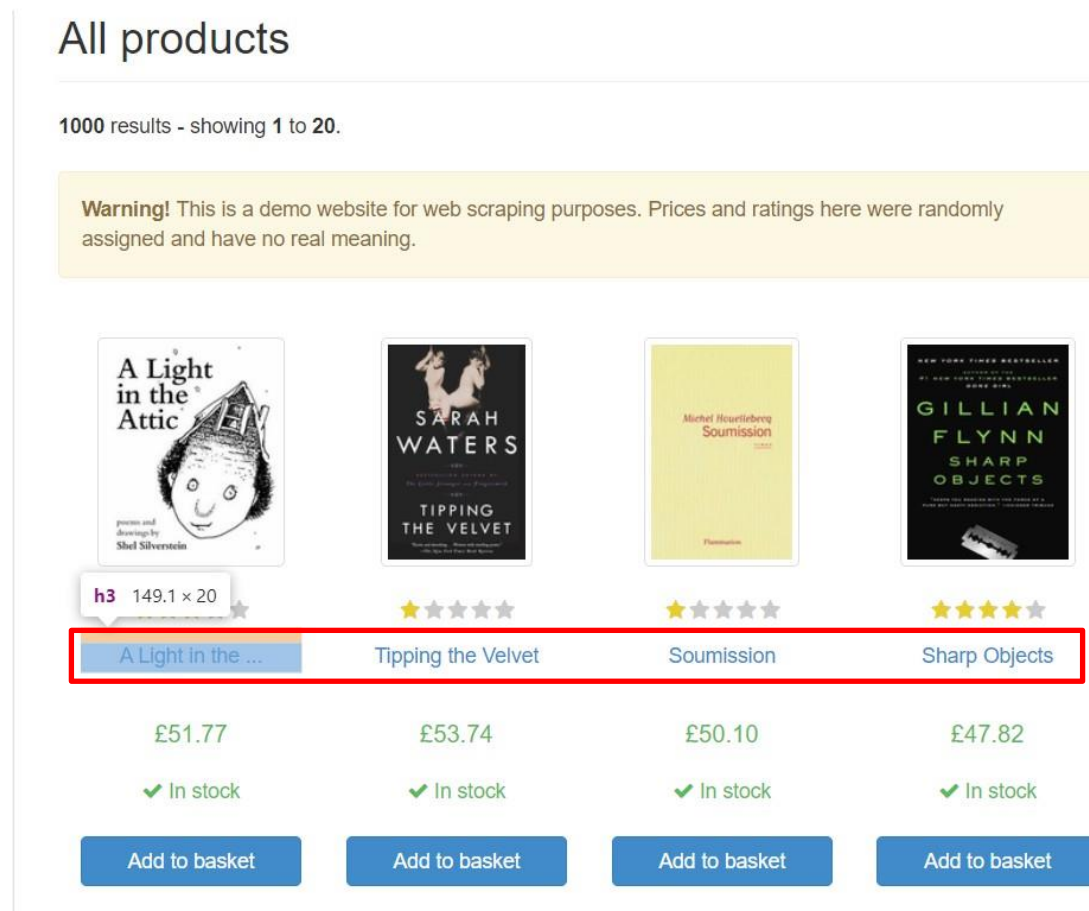
```
from bs4 import BeautifulSoup
doc = BeautifulSoup(page_contents, 'html.parser')
doc
✓ 0.3s

<!DOCTYPE html>

<!--[if lt IE 7]>      <html lang="en-us" class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
<!--[if IE 7]>        <html lang="en-us" class="no-js lt-ie9 lt-ie8"> <![endif]-->
<!--[if IE 8]>        <html lang="en-us" class="no-js lt-ie9"> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang="en-us"> <!--<![endif]-->
<head>
<title>
  All products | Books to Scrape - Sandbox
</title>
<meta content="text/html; charset=utf-8" http-equiv="content-type"/>
<meta content="24th Jun 2016 09:29" name="created"/>
<meta content="" name="description"/>
<meta content="width=device-width" name="viewport"/>
<meta content="NOARCHIVE,NOCACHE" name="robots"/>
<!-- Le HTML5 shim, for IE6-8 support of HTML elements -->
<!--[if lt IE 9]>
  <script src="//html5shim.googlecode.com/svn/trunk/html5.js"></script>
```

# Utilisation de l'élément Inspect pour obtenir l'emplacement des informations nécessaires.

## get\_book\_titles récupère le texte d'une balise dans la balise h3.

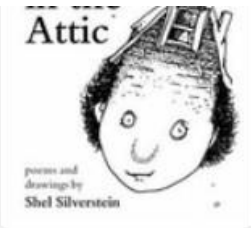
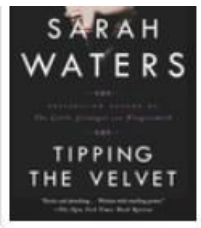

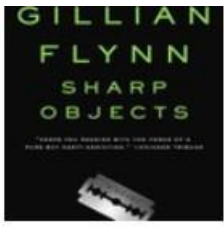


# Récupérer les titres de livres

```
def get_book_titles(doc):  
    Book_title_tags = doc.find_all('h3')  
    Book_titles = []  
    for tags in Book_title_tags:  
        Book_titles.append(tags.text)  
    return Book_titles  
get_book_titles(doc)
```

✓ 0.0s

```
['A Light in the ...',  
'Tipping the Velvet',  
'Soumission',  
'Sharp Objects',  
'Sapiens: A Brief History ...',  
'The Requiem Red',  
'The Dirty Little Secrets ...',  
'The Coming Woman: A ...',  
'The Boys in the ...',  
'The Black Maria',  
'Starving Hearts (Triangular Trade ...',  
'Shakespeare's Sonnets',  
'Set Me Free',  
'Scott Pilgrim's Precious Little ...',  
'Rip it Up and ...',  
'Our Band Could Be ...',
```

 <p>★★★★☆</p> <p>A Light in the ...</p> <p>p.price_color 149.1 x 30</p> <p>£51.77</p> <p>✓ In stock</p>	 <p>★★★★☆</p> <p>Tipping the Velvet</p> <p>£53.74</p> <p>✓ In stock</p>	 <p>★★★★☆</p> <p>Soumission</p> <p>£50.10</p> <p>✓ In stock</p>	 <p>★★★★☆</p> <p>Sharp Objects</p> <p>£47.82</p> <p>✓ In stock</p>
--	--	---	---

```

<div class="product_pod">
  <div class="image_container">...</div>
  <p class="star-rating Three">...</p>
  <h3>
    <a href="catalogue/a-light-in-the-attic_1000/index.html" title="A Light in the Attic">A
    Light in the ...</a>
  </h3>
  <div class="product price">
    <p class="price_color">£51.77</p> == $0
    <p class="instock availability">...</p>
    <form>...</form>
  </div>
</article>

```

```

def get_book_price(doc):
    Book_price_tags = doc.find_all('p', class_='price_color')
    Book_price = []
    for tags in Book_price_tags:
        Book_price.append(tags.text.replace('Â', ''))
    return Book_price

get_book_price(doc)

```

✓ 0.0s

```

['£51.77',
 '£53.74',
 '£50.10',
 '£47.82',
 '£54.23',
 '£22.65',
 '£33.34',
 '£17.93',
 '£22.60',
 '£52.15',
 '£13.99',
 '£20.66',
 '£17.46',
 '£52.29',
 '£35.02',
 '£57.25',

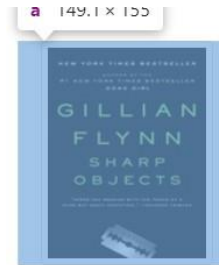
```



# Récupérer des liens pour chaque livre

```
def get_book_url(Book_title_tags):
    Book_url = []
    for article in Book_title_tags:
        for link in article.find_all('a', href = True):
            url = link['href']
            links = 'https://books.toscrape.com/' + url
            if links not in Book_url:
                Book_url.append(links)
    return Book_url
titles = doc.find_all('h3')
get_book_url(titles)
```

✓ 0.0s

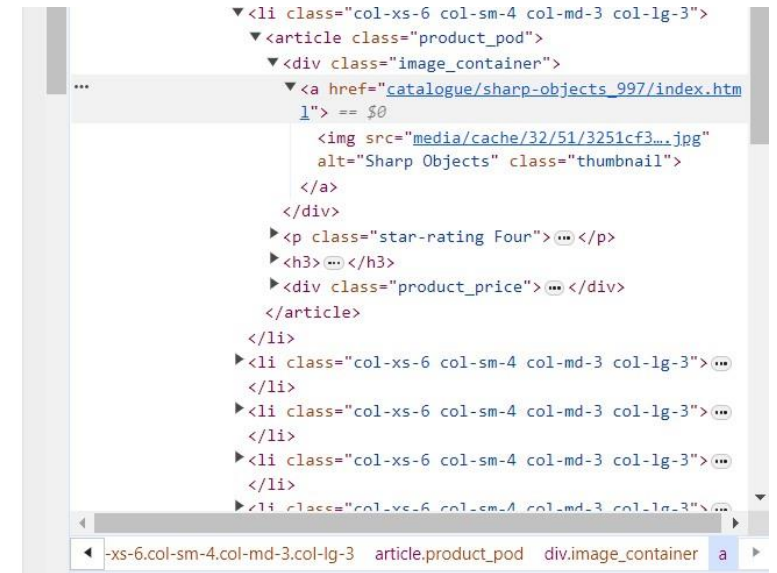


★★★★★  
Sharp Objects

£47.82

✓ In stock

Add to basket



```
['https://books.toscrape.com/catalogue/a-light-in-the-attic_1000/index.html',
 'https://books.toscrape.com/catalogue/tipping-the-velvet_999/index.html',
 'https://books.toscrape.com/catalogue/soumission_998/index.html',
 'https://books.toscrape.com/catalogue/sharp-objects_997/index.html',
 'https://books.toscrape.com/catalogue/sapiens-a-brief-history-of-humankind_996/index.html',
 'https://books.toscrape.com/catalogue/the-requiem-red_995/index.html',
 'https://books.toscrape.com/catalogue/the-dirty-little-secrets-of-getting-your-dream-job_994/index.html',
 'https://books.toscrape.com/catalogue/the-coming-woman-a-novel-based-on-the-life-of-the-infamous-feminist-victoria-woodhull_993/index.html',
 'https://books.toscrape.com/catalogue/the-boys-in-the-boat-nine-americans-and-their-epic-quest-for-gold-at-the-1936-berlin-olympics_992/index.html',
 'https://books.toscrape.com/catalogue/the-black-maria_991/index.html',
 'https://books.toscrape.com/catalogue/starving-hearts-triangular-trade-trilogy-1_990/index.html',
 'https://books.toscrape.com/catalogue/shakespeares-sonnets_989/index.html',
 'https://books.toscrape.com/catalogue/set-me-free_988/index.html',
 'https://books.toscrape.com/catalogue/scott-pilgrims-precious-little-life-scott-pilgrim-1_987/index.html',
```

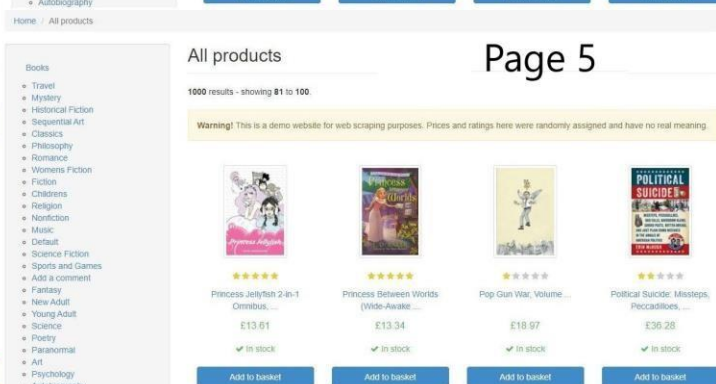
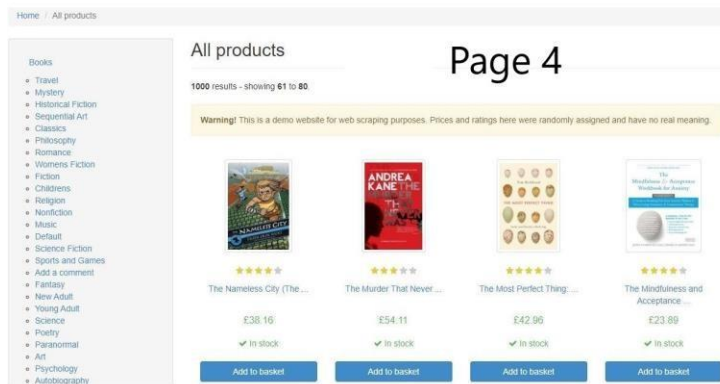
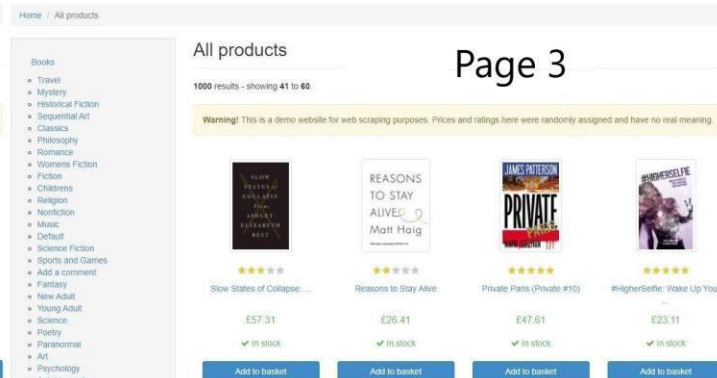
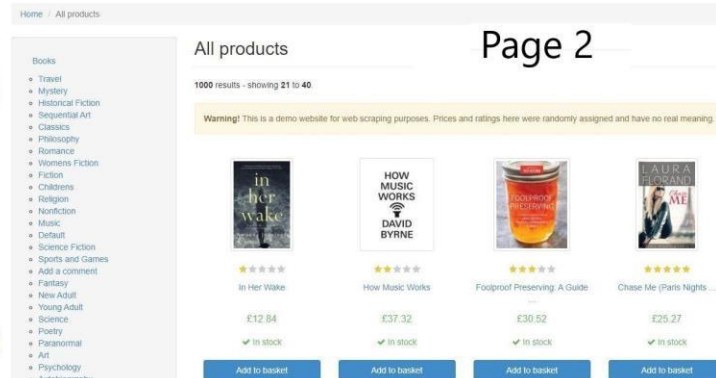
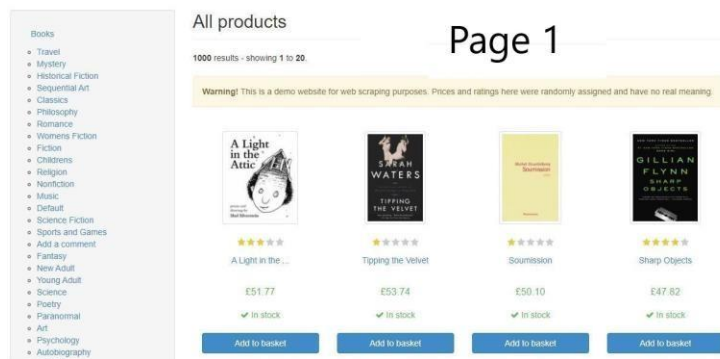
# Collecte de données à partir de plusieurs pages

- Dans cette section, nous allons créer une fonction pour collecter les informations de plusieurs pages.
- L'image montre les cinq premières pages, à partir desquelles nos informations doivent être récupérées.

<https://books.toscrape.com/catalogue/page-1.html>

<https://books.toscrape.com/catalogue/page-2.html>

<https://books.toscrape.com/catalogue/page-3.html>



<https://books.toscrape.com/catalogue/page-4.html>

<https://books.toscrape.com/catalogue/page-5.html>



# Merci de Votre Attention



# Des questions

[pedagogie@caplogy.com](mailto:pedagogie@caplogy.com)