



# \*Le web scraping\*

*par*

Dr. Samira LAGRINI



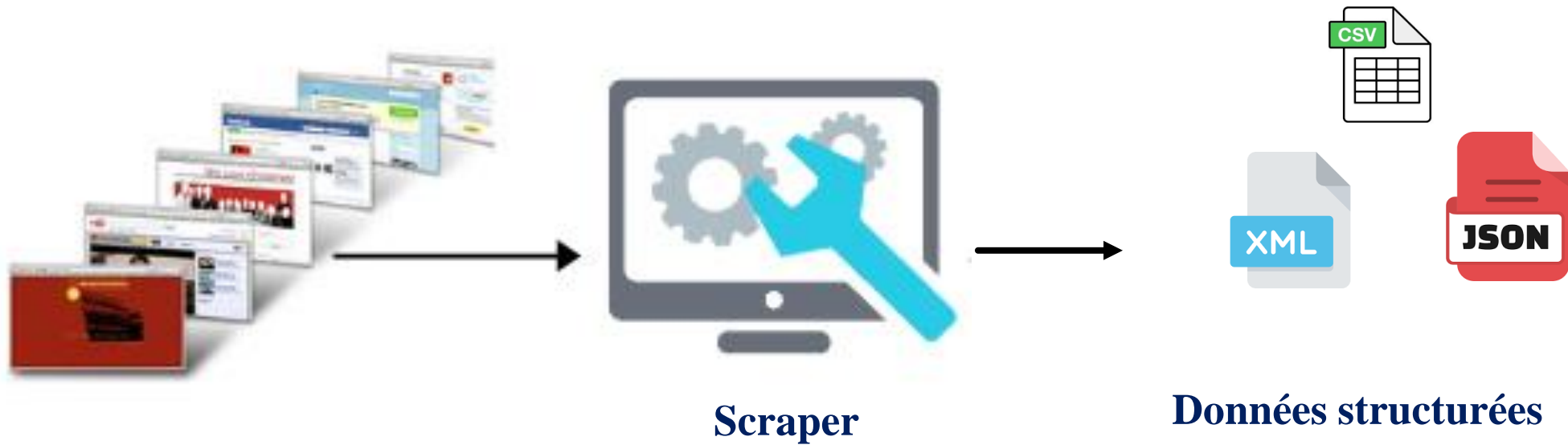
Année universitaire:2024/2025

# Qu'est ce que le web scraping????

- ▶ La collecte de données web, ou '**web scraping**', est la première étape dans le processus du web mining.
- ▶ **Le web scraping** souvent appelé '**extraction de données web**' consiste à extraire des données textuelles ou multimedias à partir de sites web de manière automatisée.
- ▶ Plutôt que de copier manuellement les données depuis les pages web, le web scraping utilise des petits programmes appelés '**scraper**' pour naviguer sur les sites web, extraire les données désirées, puis les stocker dans des formats structurés pour une analyse ultérieure.



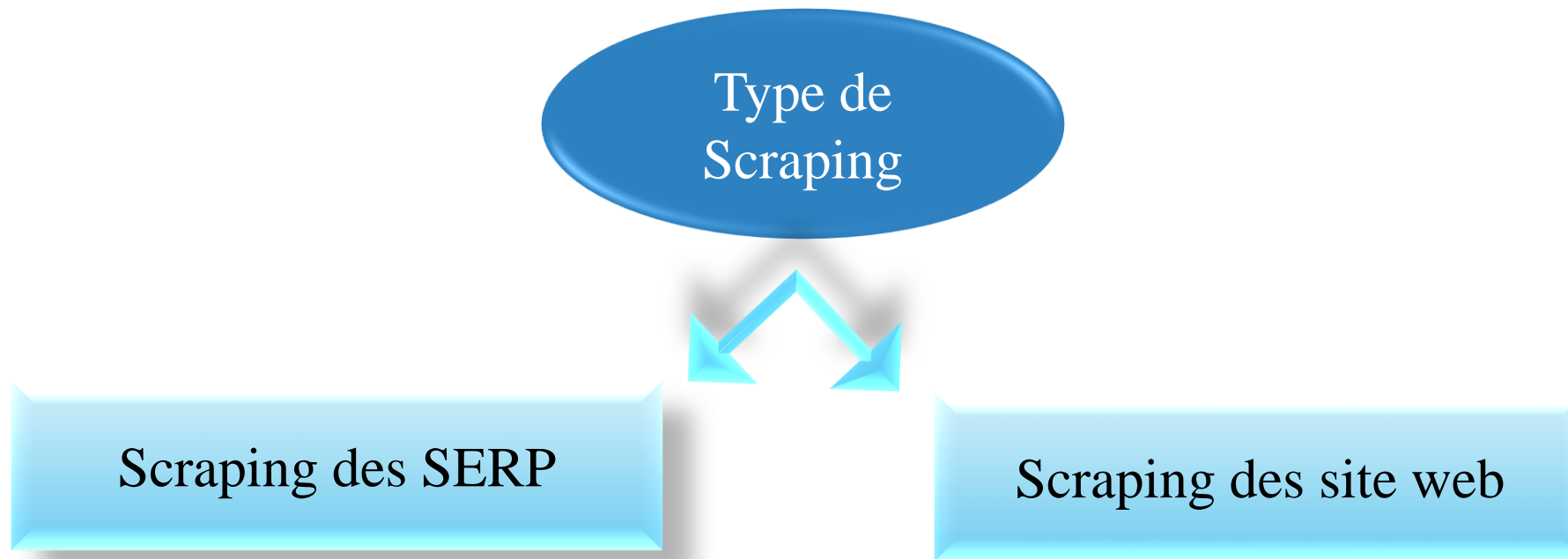
# Qu'est ce que le web scraping????



# Utilité de scraping

- ▶ **Collecte automatique de données:** le scraping permet de transformer des grandes masses de données non structurées sur le web en données exploitables, facilitant l'analyse.
- ▶ **Analyse Concurrentielle:** Dans les domaines tels que le marketing, ou la surveillance des prix, le scraping permet de suivre et d'analyser les stratégies et les offres des concurrents.
- ▶ **Recherche Académique et Analyse:** le scraping permet d'extraire de grands volumes de données utiles nécessaires pour des analyses avancées.
- ▶ **Suivi de l'Actualité:** le scraping peut aider à surveiller en temps réel les informations diffusées sur des sites web d'actualités, les réseaux sociaux ou d'autres plateformes, en fournissant une veille efficace sur des sujets spécifiques.
- ▶ **Faciliter l'Apprentissage Automatique :** Les modèles de machine learning nécessitent de grandes quantités de données pour l'entraînement. Le scraping peut aider à rassembler ces données en grande quantité, ce qui permet de créer des modèles plus robustes et précis.

# Types de scraping



# Scraping des SERP

- ▶ **SERP (Search Engine Results Page)** est la page qui s'affiche après avoir entré une requête sur un moteur de recherche (ex: Google, Bing...),
- ▶ La SERP contient différent types de données: des liens vers des pages web, des extraits enrichis, des annonces publicitaires, des titres, des images, des vidéos, ....etc
- ▶ Le scraping de SERP est utilisé pour collecter des données provenant de **multiple sources**

## Utilité:

- Suivre le positionnement des sites dans les SERP pour des mots-clés spécifiques.
- Analyse de la concurrence
- Identifier les tendances de recherche, les requêtes populaires, ....etc



# Scraping des Sites Web

- ▶ Il s'agit de l'extraction des données directement à partir des pages web elles-mêmes (et non des résultats de recherche).
- ▶ Ce type de scraping cible les données contenues sur un site web particulier (les informations de produits, les titres, les liens...etc).

## ➡ **Différence Essentielle**

Scraping de SERP est axé sur l'extraction des résultats de recherche pour analyser le positionnement dans les moteurs de recherche.

Scraping de sites web cible les données spécifiques contenues sur les pages des sites eux-mêmes.



# Processus de web scraping





# Processus de web scraping

## 1) Identifier l'URL de la source de données

Cette étape consiste à déterminer l'emplacement exact des données et la manière d'y accéder.

## 2) Envoyer une requête pour collecter le contenu brut

Envoyez une requête HTTP vers l'URL identifiée pour obtenir le **contenu brut** :

- **HTML brut** : Le code complet de la page web, incluant les balises, le texte et les attributs.
- **JSON ou XML** : Les données brutes fournies par une API, structurées mais nécessitant une extraction.
- **Texte brut** dans un fichier (ex. : PDF, CSV) : Le texte ou les données non transformées contenues dans le document.

# Processus de web scraping

## 3) Analyser (extraire) et interpréter le contenu brut

Parcourez le contenu brut pour identifier et extraire les données pertinentes, en ciblant les éléments spécifiques dans le code HTML, les champs JSON, ou les sections de texte d'un fichier.

Pour cela, Utilisez des **bibliothèques** (comme **BeautifulSoup**, **json**, **etc.**) et des **parseurs** (comme **html.parser**, **lxml**, **etc.**)

Organisez les données extraites dans un format structuré (tableaux, dictionnaires, etc.) pour les rendre exploitables.

## 4) Stocker les données extraites

Sauvegardez les données extraites et nettoyées dans un format adapté (CSV, JSON, base de données, etc.), afin de les utiliser pour des analyses ou des traitements.

# Processus de web scraping

## *Remarque*

- ❖ La gestion des erreurs doit être intégrée à chaque étape pour garantir la robustesse et la fiabilité du processus, en vérifiant l'accès aux URLs, en traitant les erreurs de requêtes, en validant l'intégrité du contenu brut, et en s'assurant que le stockage ou l'utilisation se fait correctement.



**Exemple:** scraper les noms et les prix des produits d'une page de vente en ligne

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

# Étape 1 : Identifier l'URL de la source de données
url = 'https://exemple.com/produits'

# Étape 2 : Envoyer une requête pour collecter le contenu brut
response = requests.get(url)
html_content = response.text # Contenu brut récupéré

# Étape 3 : Analyser (extraire) le contenu brut
soup = BeautifulSoup(html_content, 'html.parser')

# Extraction des noms et des prix des produits
noms = [item.text.strip() for item in soup.find_all('h2', class_='nom-produit')]
prix = [item.text.strip() for item in soup.find_all('span', class_='prix-produit')]

# Interprétation : Organiser les données dans un tableau
produits = [{ 'Nom': nom, 'Prix': prix } for nom, prix in zip(noms, prix)]

# Étape 4 : Stocker ou utiliser les données
df = pd.DataFrame(produits)
df.to_csv('produits.csv', index=False)

print("Les données ont été extraites et sauvegardées dans 'produits.csv'.")
```

# Est ce qu'on peut scraper tous les site web?

**Non**, il n'est pas autorisé de scraper **tous** les sites web.

Certains sites limitent ou interdisent le scraping pour diverses raisons, notamment la **protection des données**, la **confidentialité**, et la **charge du serveur**.



## Comment comprendre ce qui est possible et légal ?

- Il suffit de lire le fichier 'robots.txt' de site que vous voulez scraper ses données.
- Pour cela, il faut mettre robots.txt après L'URL de site à scraper
- Ce fichier définit les parties du site que les bots (y compris les scrapers) sont autorisés ou interdits à visiter.
- Si le fichier robots.txt contient des instructions '**Disallow**', cela signifie que le scraping de ces pages est interdit.

# robots.txt

Analysons le fichier **robots.txt** de **yahoo.com**

Cette ligne signifie que les règles ci-dessous s'appliquent à tous les robots quel que soit leur nom.

Chaque ligne "Disallow" indique aux robots de ne pas accéder aux URLs commençant par ces directives: p , r , bin, .....

Cette ligne indique que **Scrapy** n'a pas la permission d'explorer **aucune partie** du site.

```
User-agent: *
Disallow: /p/
Disallow: /r/
Disallow: /bin/
Disallow: /caas/
Disallow: /blank.html
Disallow: /includes/
Disallow: /_td_api
Disallow: /tdv2_fp
Disallow: /nel_ms
Disallow: /fp_ms
Disallow: /sports_fp_ms
Disallow: /search_ms
Disallow: /_tdpp_api
Disallow: /_remote
Disallow: /_multiremote
Disallow: /_tdhl_api
Disallow: /digest
Disallow: /fpjs
Disallow: /mvic

User-agent: Nutch
Disallow: /

User-agent: omgili
Disallow: /

User-agent: omgilibot
Disallow: /

User-agent: panscient.com
Disallow: /

User-agent: Perplexity-ai
Disallow: /

User-agent: PerplexityBot
Disallow: /

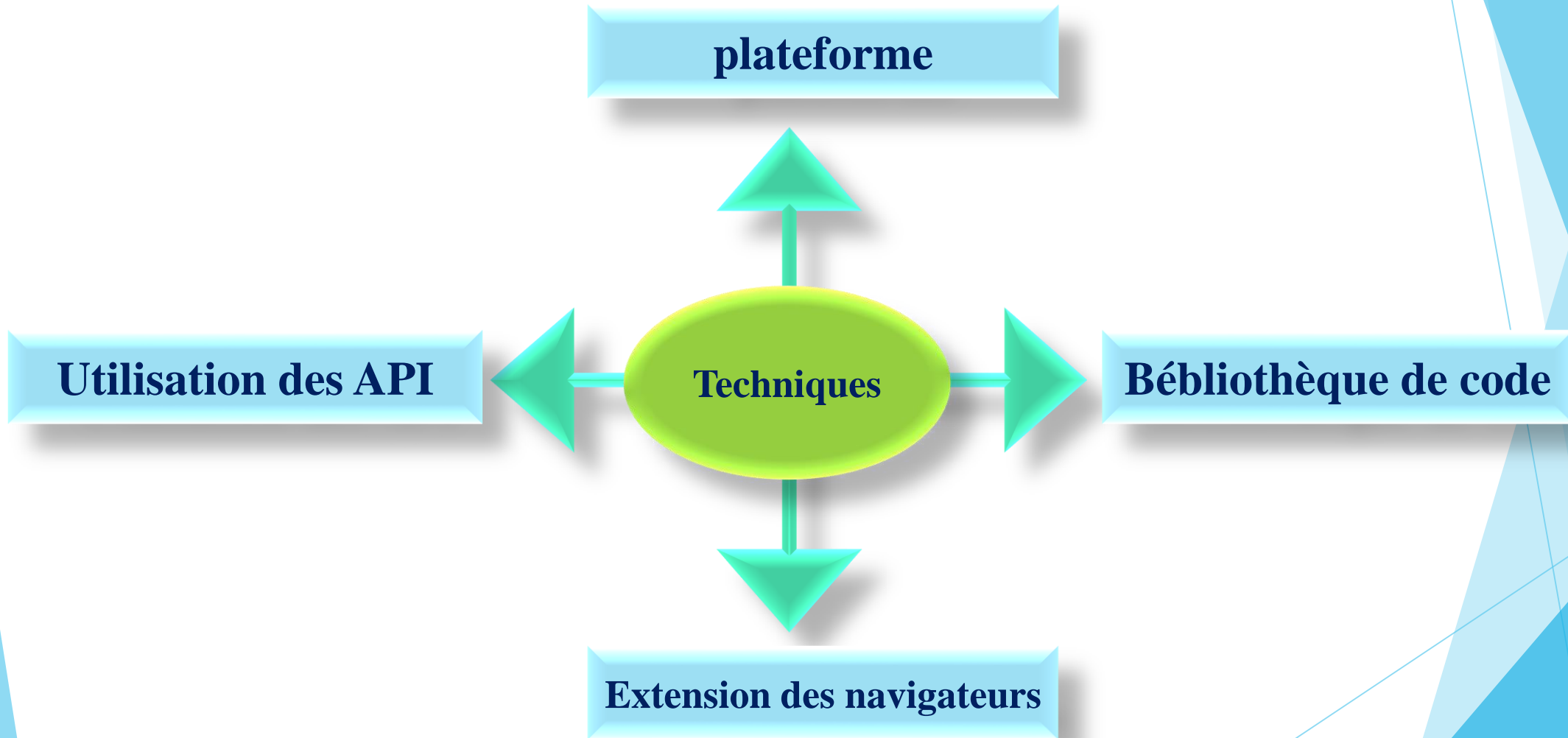
User-agent: PetalBot
Disallow: /

User-agent: PiplBot
Disallow: /

User-agent: scoop.it
Disallow: /

User-agent: Scrapy
Disallow: /
```

# Techniques de web scraping





# Scraping avec des Bibliothèques de code et des Frameworks

- Utilisation de **bibliothèques et de frameworks** de programmation pour écrire du code qui extrait des données de sites web.

## *Exemples d'outils :*

- **BeautifulSoup** : bibliothèque python utilisé pour extraire des données des pages statiques (HTML et XML)

### **Requests-HTML**

Bibliothèque légère permettant de rendre les pages dynamiques et d'extraire le contenu généré par JavaScript.

- **Scrapy** : Framework python open source offrant des outils robusts pour extraire des données de pages statiques.

- **Selenium, Playwright** : Pour interagir avec des pages dynamiques et gérer le JavaScript.

# Scraping via des plateformes

- ❑ Utilisation de plateformes spécialisés dans le scraping, offrant des solutions prêtes à l'emploi sans nécessiter de codage.
- ❑ Ces plateformes permettent de configurer visuellement des projets de scraping.

## *Exemples d'Outils*

- **Octoparse**: plateforme visuelle permettant de créer des scrapers sans écrire de code
- **ParseHub**: utilise des techniques d'apprentissage automatique pour extraire des données des sites complexes
- **Import.io** : offre des fonctionnalités avancées pour extraire les données

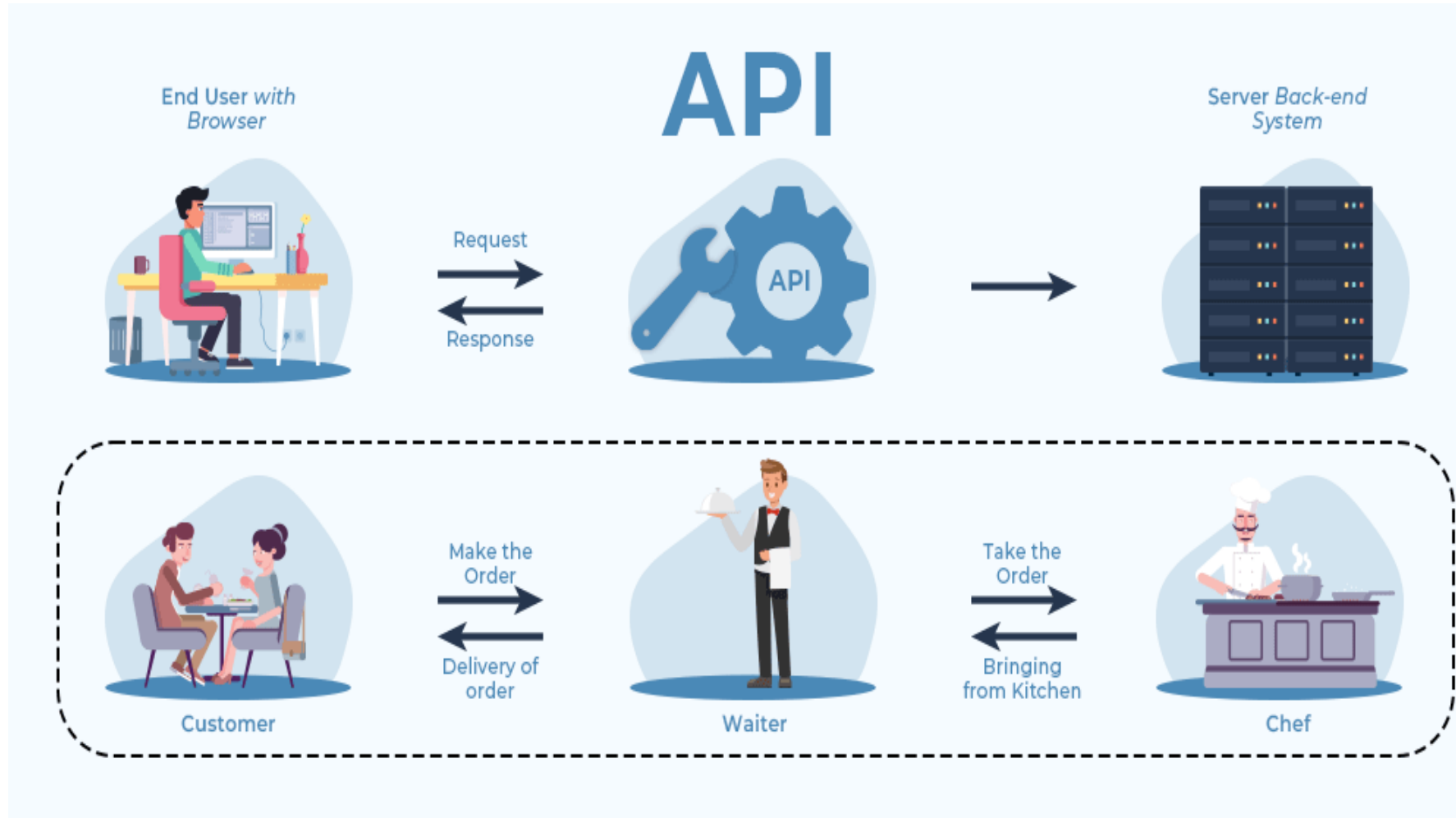
# Scraping via des APIs

- ❑ Une **API (Interfaces de Programmation d'Applications)** est un ensemble de règles et de protocoles qui permettent à une application d'interagir avec un autre logiciel ou service.
- ❑ Une API sert de pont entre différents systèmes, permettant l'échange de données et de fonctionnalités.
- ❑ De nombreux sites proposent des **APIs publiques** pour accéder aux données de manière structurée et fiable.

*Exemple d'API:*

- ▶ **APIs de réseaux sociaux** (Twitter, Facebook, LinkedIn)
- ▶ **APIs de géolocalisation** (Google Maps API)
- ▶ **APIs de données publiques** (API OpenWeather)

# Scraping via des APIs



# Scraping via des APIs

## Pourquoi utiliser des APIs ?

- les APIs fournissent des données déjà formatées (structurées)
- Les données obtenues via les APIs sont généralement offertes légalement par le fournisseur du service, ce qui limite les risques légaux liés à la collecte de données.
- Efficacité et rapidité : Les APIs sont souvent optimisées pour renvoyer uniquement les données nécessaires, ce qui améliore les performances et réduit le temps de réponse.

# Scraping via des Extensions de Navigateurs

- ▶ Utilisation d'extensions ou de plugins installés sur le navigateur pour extraire des données directement depuis les pages web.
- ▶ Simplicité d'utilisation avec une interface conviviale.
- ▶ Extraction des données visibles sur les pages chargées dans le navigateur.

## *Exemples*

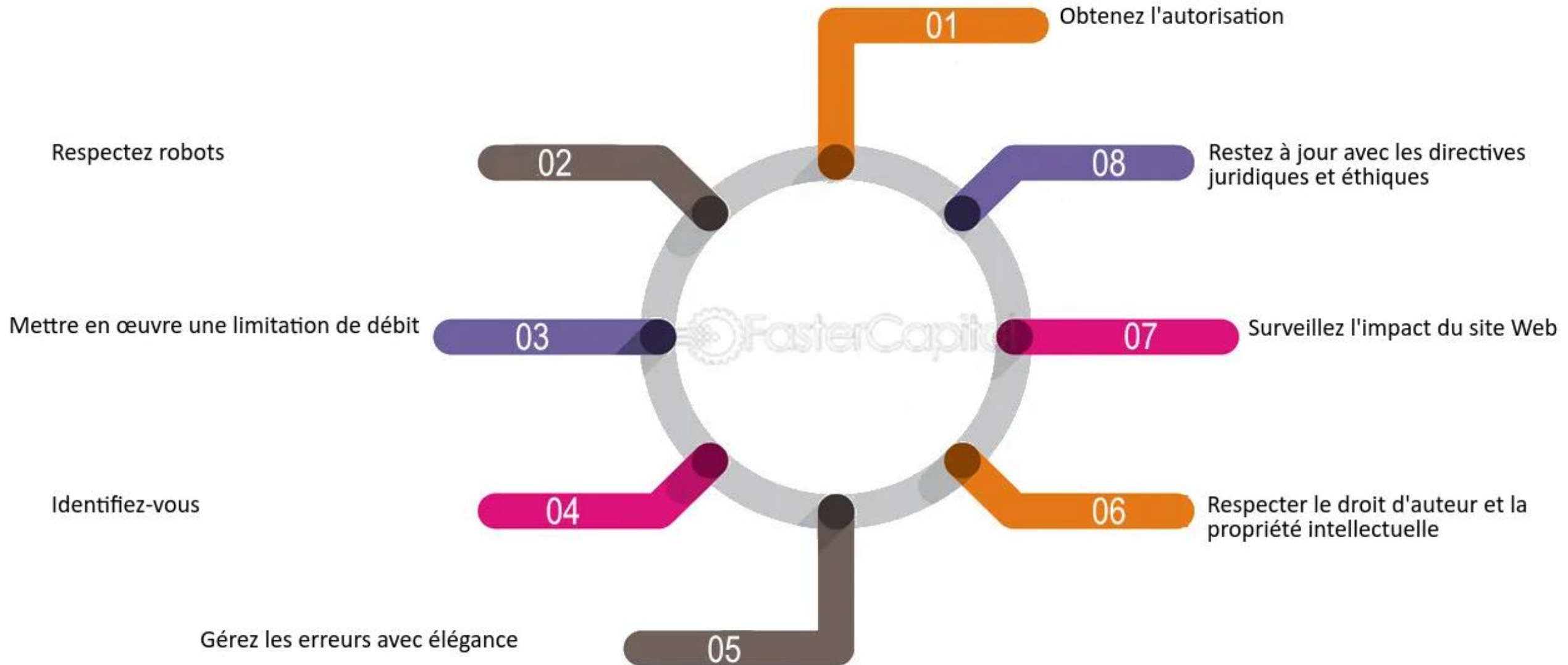
- Data Miner, Web Scraper (extensions pour Google Chrome).

# Comparaison entre les techniques de scraping

Technique de scraping	Avantages	Inconvénients
Utilisation des bibliothèques et de Framework	Flexible - Contrôle total sur le processus de scraping.	Nécessite des compétences en programmation.
les Plateformes	Facile à utiliser et à configurer.	- Moins flexible que le codage direct - Peut être limité par des fonctionnalités payantes ou des restrictions de volume.
Les APIs	Rapide, précis, et conforme aux politiques des sites web. Réduit le risque de blocage, car les données sont fournies de manière formelle.	- Accès limité par des restrictions d'utilisation. - Ne permet d'obtenir que les données partagées par l'API.
Les Extensions de Navigateurs	Facile à installer et à utiliser	Moins adapté aux projets complexes. Limitations en termes de fonctionnalité et de performance.



# Éthiques du Web Scraping



# Éthiques du Web Scraping

Il est essentiel de :

- Demander une **autorisation préalable** ou utiliser les **API fournies** pour un accès légitime aux données.
- Lire et respecter les **conditions d'utilisation** des sites web.
- Vérifier le **fichier robots.txt** pour s'assurer que le scraping est autorisé.
- S'assurer de la **légalité des données** collectées et respecter les lois spécifiques de chaque pays en matière de collecte de données.
- Se conformer aux **réglementations de protection des données**, notamment en évitant d'extraire des données sensibles ou personnelles pour préserver la confidentialité des utilisateurs.
- Adopter une approche **responsable et éthique** pour éviter de nuire aux performances des sites web.



# Protection contre le web scraping



Setting Up  
Robots.txt



Filtering Requests  
By User Agent



Blacklisting IP  
Addresses



Showing A Captcha



Honeypots

# Protection contre le web scraping

## ❑ Fichier robots.txt

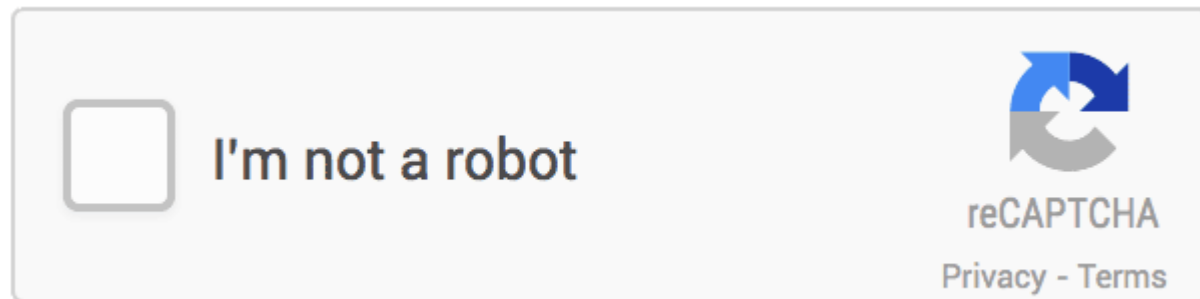
Les sites peuvent utiliser le fichier robots.txt pour indiquer aux scrapers les parties du site qu'ils ne peuvent pas scraper. Bien que cela ne soit pas contraignant techniquement, il s'agit d'une première ligne de défense pour les bots respectueux des règles.

# Protection contre le web scraping

## ❑ Affichage d'un CAPTCHA

Les CAPTCHA exigent que l'utilisateur saisi un texte ou identifie des éléments d'une image, ce qui est difficile pour les bots.

Google reCAPTCHA est une version avancée qui analyse le comportement des utilisateurs pour distinguer les humains des bots sans nécessiter d'interaction supplémentaire.



# Protection contre le web scraping

## ❑ Blocage par User-Agent


Les sites peuvent bloquer ou limiter l'accès à certains user-agents identifiés comme des crawlers ou des scrapers, ce qui oblige ces derniers à déguiser leur identité en imitant le user-agent d'un navigateur légitime.

- ❑ Un *user-agent* d'un navigateur est une chaîne de texte envoyée dans les en-têtes HTTP lors des requêtes d'un client (comme un navigateur) vers un serveur web. Cette chaîne fournit des informations sur le navigateur utilisé, le système d'exploitation, la version du navigateur, et parfois le type d'appareil.

*Exemple:*

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/97.0.4692.71  
Safari/537.36

# Protection contre le web scraping



bien que les *user-agents* des navigateurs et des scrapers puissent être très similaires, les scrapers peuvent parfois être détectés par des détails mineurs, comme des versions incohérentes, un manque de précision dans les informations système, ou un changement fréquent de *user-agent*.



# Protection contre le web scraping

## ► **Changement Dynamique de Contenu**

Cette technique consiste à modifier dynamiquement la structure ou le contenu d'une page de manière régulière.

Les scrapers peinent alors à interpréter le contenu, car les éléments à extraire changent constamment.

# Protection contre le web scraping

## Les honeypots

- ❑ Les honeypots sont des éléments cachés (*un lien ou un champ de formulaire caché*) que les utilisateurs humains ne peuvent pas voir ni interagir avec, mais qui sont accessibles aux bots automatisés.
- ❑ Les **honeypots** sont utilisée pour attirer et piéger les scrapers et autres bots malveillants.
- ❑ Lorsqu'un honeypot est déclenché, le site peut automatiquement bloquer ou restreindre l'accès de l'adresse IP ou de l'agent utilisateur associé au bot suspect.

# Protection contre le web scraping

## ► Limitations de Taux (Rate Limiting)

Cette technique contrôle le nombre de requêtes envoyées à un site par une adresse IP sur une période définie.

Si le nombre dépasse un seuil prédéfini, l'accès est temporairement restreint ou bloqué.

# Protection contre le web scraping

## ► Protection par session et cookies

Certains sites exigent l'établissement de sessions valides et l'utilisation de cookies pour accéder aux pages.

Sans ces cookies ou sans une session active, le scraper peut se voir refuser l'accès.

➡ Ces techniques peuvent être utilisées individuellement ou en combinaison pour rendre le scraping difficile, coûteux ou même impossible sans autorisation appropriée.

# Quelle est la différence entre le scraping et le crawling



- ❖ **Le scraping** est l'extraction des données spécifiques (les prix des produits, les avis des utilisateurs..) à partir d'une source en ligne bien défini (page web, sites web)
- ❖ **Le crawling** est le processus d'exploration automatique de pages web à l'aide de programmes appelés "**crawlers** » afin de découvrir et d'indexer de nouvelles pages web.

## Exemple d'utilisation

- ▶ Les moteurs de recherche utilisent des **crawlers** pour explorer et indexer les pages web, ce qui leur permet de rendre ces pages accessibles dans les résultats de recherche.
- ▶ Un site de comparaison des prix utilise des scrapers pour extraire et collecter les prix des produit des sites de e-commerce

# Quelle est la différence entre le scraping et le crawling

Aspect	Crawling	Scraping
Objectif	Découvrir et indexer des pages web	Extraire des données spécifiques des pages web
Cible	Web entier	Cible bien défini
Analyse	Globale	Approfondie
Fin	Indexation et archivage	Stratégique
Processus	Exploration systématique de liens et de pages	Extraction ciblée de contenus ou d'informations

## ❑ Utilisation Conjointe

Dans de nombreux projets, le crawling et le scraping sont utilisés ensemble :

**Le crawling** → identifie les pages pertinentes en suivant les liens sur un site web.

**Le scraping** → intervient ensuite pour extraire des données précises des pages identifiées par le crawler.

# Travaux pratiques





# TP1 : Extraction de Données Produits avec l'API eBay

## Objectif

Ce TP vise à vous familiariser avec l'utilisation de l'API d'eBay pour extraire des informations sur des produits spécifiques. Vous allez apprendre à configurer l'accès à l'API, à effectuer des requêtes pour récupérer des données produits et à analyser ces informations.

# Étapes à suivre

- 1) **Configuration de l'accès à l'API eBay** (Créer un compte développeur, Générer les clés d'API)
- 2) **Exploration de la documentation de l'API** :Se familiariser avec l'API "Browse" d'eBay, qui permet de rechercher et récupérer des informations sur les produits.
- 3) **Récupération des données produits** (nom du produit, le prix, la description et la disponibilité.)
- 4) **Analyse des données** :Nettoyer et structurer les données récupérées.  
Effectuer des analyses statistiques simples, comme la moyenne des prix par catégorie.
- 5) **Visualisation des résultats**  
Créer des graphiques pour illustrer les tendances observées, par exemple la distribution des prix par catégorie.

