# Projet
# Python Web Scraping

Benjamin LOPES

# Sommaire :

# Introduction:

Lors de ce projet, nous devons nous familiariser avec le web scraping. Nous utiliserons python et des bibliothèques adapter pour le web scraping : requests et BeautifulSoup pour faire nos requêtes et nous utiliserons la bibliothèque csv pour mettre le résultat de notre scraping dans un fichier csv. Dans un premier temps nous devrons donc définir le web scraping et ses étapes. Ensuite nous cartographierons le site afin de se familiariser avec celui-ci. Après cela nous nous occuperons de créer notre script python. Et enfin nous finirons par des étude de cas et l'analyse des données récupérer.

# Partie 1 : Définition et les étapes

Le web scraping est une technique utilisée pour collecter automatiquement des informations à partir de sites web. Elle permet des données structurées, comme du texte, des images ou des tableaux, à partir de pages web afin de les analyser ou de les réutiliser.

Le web scraping se fait en plusieurs étapes :

- 1. Identifier la cible :
    - On determine le site ou la page web qui contient les données à extraire
    - On analyse la structure des informations à récupérer (texte, tableaux, liens,ect.)

- 2. Analyser la structure HTML :
    - On utilise des outils comme les navigateurs web pour comprendre le code HTML de la page
    - On Localise les balises spécifiques contenant les données (exemple : , <table>, )

- 3. Envoyer une requête http
    - On utilise une bibliothèque (ici requests) pour envoyer une requête au serveur web et obtenir le contenu HTML de la page
- 4. Extraire les données
    - On analyse le contenu HTML avec des outils (ici BeautifulSoup)
    - On récupère les informations nécessaires à l'aide de sélecteurs (exemple : balsies, classes, IDs)
- 5. Nettoyer les données
    - On formate et structure les données extraites pour les rendre exploitables.
- 6. Stocker les données :
    - On enregistre les données dans un format approprié comme une fichier (exemple : csv, json, Excel, etc.) ou directement dans une base des bases de données relationnelles ou non.
- 7. Automatiser et respecter l'éthique
    - On crée des scripts automatisés pour exécuter régulièrement les tâches de scraping si nécessaire.
    - On respecte les règles du site web, notamment via le fichier robots.txt,

# Partie 2 : Cartographie du site

a. Identification des données à récupérer

Les données à récupérer sont les suivantes :

- Le bloc de la citation
- La citation
- L'Auteur de la citation
- Les tags associés à la citation

b. Identification des balises html

Les balises qui nous intéressent sont :

-  qui contient l'ensemble des données d'une citation
-  qui contient la citation
- <small class='author'> qui contient le nom auteur
-  / <a class='tag'> qui contient les tags associés à la citation

c. Création d'une table de correspondance avec les éléments précédents

| Données | Html |
|---|---|
| Bloc de la citation |    |
| La citation |     |
| L'auteur de la citation |     <small class='author'> |
| Les tags associés |     <a class='tag'> |

# Partie 3 : Description de ma solution

J'ai décider d'utiliser un fichier csv comme méthode de stockage pour mon web scraper. J'ai donc recherché des solutions à des problèmes comme l'affichage des données et j'ai convertit les donner récupérer en utf-16 permettant de conserver les guillemets présentent sur le site mais causant le fichier d'augmenter en taille. Et une fois ouvert en fichier Excel il faudra tout de même ajuster soi-même les cases pour faciliter la lecture.

Ensuite pour la récupération des données j'ai utilisé la bibliothèques requests de python qui comme son nom l'indique de faire des requêtes auprès des pages web. Et j'ai également utilisé BeautifulSoup qui permet de faciliter l'exploitation des données scraper d'un site.

Une fois la page récupérer nous allons traitez les données, pour commencer nous allons séparer les données en bloc de citation qui comporte l'ensemble des données d'une citation. Ensuite, une fois le découpage fait, pour chacun des blocs nous allons récupérer les données qui nous intéressent : la citation, l'auteur et les tags. Et une fois récupérer nous les plaçons dans une liste que l'on utilisera pour créer le fichier csv.

# Partie 4 : Cas d'études

Les données récupérer pourrai servir à différentes choses, par exemple extraire uniquement les citations d'un auteur ou encore toutes les citations ayant un tag un particulier. Nous aurions pu aussi récupérer les liens vers les pages dédies aux biographies des auteurs. Avec les données scaper nous aurions pu étudier les probabilité qu'un auteur soit lu par un visiteur basé sur la popularité des tags et/ou de l'auteur. Les données auraient aussi pu servir de base pour certaine information sur une page web que l'on créera ou on pourra encore servir de base pour la création d'une base de données.

# Conclusion :

Tout au long de ce projet et même avec les exercices précédents, nous avons appris à utiliser python pour manipuler des données. Nous finissons donc avec le web scraping qui consiste à récupérer les données d'une page web et d'extraire celle qui nous intéresse afin d'en faire autre chose comme de l'analyse statistique, création de base de données, ect. Nous avons donc appris a maitriser des bibliothèques qui nous étaient pour certaine inconnu et donc à nous documenter sur elles.