Mahamat Hassan Issa
Data scientist et Formateur

Cours Web Scraping

Partie 1 : Structure d'une page web

Voici une liste des mots-clés et concepts essentiels à connaître pour bien comprendre et pratiquer le web scraping :

Mots-clés techniques

1. **HTML:** Langage de balisage structurant les pages web.
    - Balises courantes : , <p>, <a>, <table>, <img>.
2. **CSS:** Utilisé pour styliser les pages web.

Une **balise** en HTML est une composante fondamentale utilisée pour structurer et organiser le contenu d'une page web. Elle est représentée par des mots-clés encadrés par des chevrons (< >) et sert à définir le type et le rôle du contenu qu'elle encapsule.

Exemple simple d'une page HTML avec des balises :

```html
<!DOCTYPE html>
<html>
<head>
    <title>Exemple de balises</title>
</head>
<body>
    <h1>Bienvenue sur ma page</h1>
    <p>Ceci est un paragraphe.</p>
    <a href="https://example.com">Visitez ce lien</a>
</body>
</html>
```

Voici une liste des balises HTML les plus courantes et leurs définitions pour bien comprendre la structure d'une page web lors du web scraping :

Balises de structure

1. **<html>:** La balise racine d'un document HTML. Tout le contenu HTML est encapsulé à l'intérieur.
2. **<head>:** Contient des méta-informations sur le document (titre, encodage, styles, scripts, etc.).
3. **<body>:** Contient tout le contenu visible de la page (texte, images, vidéos, etc.).

Balises de texte

4. **<h1>à <h6>:** Balises pour les titres. <h1>est le plus important et <h6>le moins.
5. **<p>:** Définit un paragraphe de texte.

6. **:** Utilisé pour styliser ou sélectionner une partie spécifique d'un texte.
7. **:** Rend le texte en gras (important sémantiquement).
8. **:** Rendre le texte en italique (importance sémantique).

Balises de privilèges et de navigation

9. **<a>:** Définit un lien hypertexte. Attribut clé : hrefpour l'URL cible.
10. **<nav>:** Définit une section de navigation (menu, liens internes).

Balises de liste

11. **<ul>:** Définit une liste non ordonnée (puces).
12. **<ol>:** Définit une liste ordonnée (numéros).
13. **<li>:** Définit un élément d'une liste.

Balises de table

14. **<table>:** Définit une table.
15. **<tr>:** Définit une ligne de table.
16. **<td>:** Définit une cellule dans une table.
17. **<th>:** Définit une cellule d'en-tête dans une table.

Balises multimédias

18. **<img>:** Insère une image. Attribut clé : srcpour l'URL de l'image.
19. **<video>:** Insérez une vidéo.
20. **<audio>:** Insérer un fichier audio.

21. **<source>:** Définit une source pour les balises <video>ou <audio>.

Formulaires de contact

22. **<form>:** Définit un formulaire pour la saisie utilisateur.
23. **<input>:** Définit un champ de saisie. Attributs clés :
    - type: Déterminez le type de saisie (texte, email, mot de passe, etc.).
24. **<textarea>:** Définit une zone de texte multiligne.
25. **<button>:** Définit un bouton cliquable.

Partie 2 : Web Scraping avec BeautifulSoup

BeautifulSoup : Points Clés

1. **Définition** :
    - BeautifulSoup est une bibliothèque Python utilisée pour extraire et analyser les données provenant de fichiers HTML et XML.
    - Elle permet de naviguer dans des documents structurés (balises, attributs, etc.) de manière simple et intuitive.

   Exemple pratique :

```python
from bs4 import BeautifulSoup
import requests

# Récupérer le contenu d'une page web
url = "https://example.com"
response = requests.get(url)
soup = BeautifulSoup(response.content, 'html.parser')

# Extraire des titres
titles = soup.find_all('h2')
for title in titles:
    print(title.text)
```

2. **Fonctionnalités principales** :
    - **Parsing HTML/XML** : Convertit le contenu brut HTML ou XML en un arbre d'objets Python manipulable.
    - **Recherche de contenu** :
        - Rechercher des balises spécifiques (find, find_all).
        - Naviguer entre les balises parents, enfants et frères.
    - **Nettoyage de données** : Éliminer les balises inutiles, extraire uniquement les données pertinentes.
    - **Support de divers parseurs** : Compatible avec des parseurs comme html.parser, lxml, et html5lib.
3. **Utilisation typique** :
    - Web scraping (extraction de données de pages web).
    - Extraction de texte pour des analyses ultérieures.
    - Prétraitement de données non structurées issues du web.

Rôle dans la science des données

1. **Collecte de données** :
    - Utilisée pour extraire des données brutes d'Internet (par exemple : articles, prix de produits, commentaires, etc.).
    - Facilite la construction de jeux de données personnalisés à partir de sources web.
2. **Préparation des données** :
    - Nettoyage des données extraites pour les structurer et les rendre prêtes à être analysées.
    - Suppression des balises HTML, gestion des espaces blancs, et extraction d'informations utiles (titres, tableaux, images, etc.).
3. **Analyse de texte** :
    - Extraction de contenu textuel pour des tâches comme le traitement du langage naturel (NLP).
    - Analyse de sentiments, classification de textes, ou résumé automatique.
4. **Exploration des tendances** :
    - Identifiant des modèles ou des tendances dans les données extraites, comme les avis des consommateurs, les fluctuations de prix, etc.