TP2 : Web Scraping Avancé et Analyse de Données avec Pandas

Scraping + Pandas + Visualisation

Applications Concrètes au Génie Civil

Niveau : École d'Ingénieurs - Première année

Durée : 2h30

24 octobre 2025

1 Introduction et rappels

1.1 Objectifs du TP2

Ce deuxième TP se concentre sur des techniques avancées de web scraping et l'analyse des données collectées avec Pandas. Vous allez :

— Scraper un catalogue en ligne de matériaux BTP (MarketBTP)

— Gérer la pagination pour collecter de grandes quantités de données

— **Utiliser Pandas pour analyser les données collectées**

— **Réaliser des statistiques descriptives sur des données de génie civil**

— **Visualiser les données avec Matplotlib**

— Comparer les prix et identifier les meilleures opportunités

1.2 Présentation du site MarketBTP

Pour ce TP, vous allez scraper un site web en ligne : **MarketBTP**, un catalogue professionnel de matériaux de construction.

URL du site : http://www.malomatique.free.fr/MarketBTP/index.html

Caractéristiques du site :

— 60 produits BTP répartis sur 3 pages

— 5 catégories : Béton, Acier, Maçonnerie, Isolation, Charpente

— Pour chaque produit : nom, fournisseur, prix, caractéristiques techniques, délai, région, note, disponibilité

— Données réalistes pour analyses pertinentes

— Pagination fonctionnelle : page-2.html, page-3.html

1.3 Vérification de l'environnement

Assurez-vous d'avoir installé les bibliothèques nécessaires :

```
pip install requests beautifulsoup4 lxml pandas matplotlib
```

Test rapide :

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt

print("Environnement pret !")
print(f"Version Pandas : {pd.__version__}")
```

2 Introduction à Pandas (20 min)

2.1 Qu'est-ce que Pandas ?

Pandas est une bibliothèque Python puissante pour la manipulation et l'analyse de données. Elle est particulièrement utile pour :

— Organiser des données tabulaires (comme des tableaux Excel)

— Calculer des statistiques (moyennes, totaux, etc.)
— Filtrer et trier des données
— Exporter vers CSV, Excel, etc.
— Visualiser des données

2.2 Structure principale : le DataFrame

Un DataFrame est un tableau à deux dimensions avec des lignes et des colonnes.

```python
import pandas as pd

# A partir d'un dictionnaire
data = {
    'Materiau': ['Beton', 'Acier', 'Bois'],
    'Prix': [95.50, 850.00, 120.00],
    'Unite': ['m3', 'tonne', 'm3']
}

df = pd.DataFrame(data)
print(df)

# Resultat :
#   Materiau    Prix  Unite
# 0    Beton   95.50     m3
# 1    Acier  850.00  tonne
# 2     Bois  120.00     m3
```

Listing 1 – Créer un DataFrame

2.3 Opérations de base sur les DataFrames

2.3.1 Affichage et information

```python
# Afficher les premieres lignes
print(df.head())

# Afficher les dernieres lignes
print(df.tail())

# Informations sur le DataFrame
print(df.info())

# Statistiques descriptives
print(df.describe())

# Dimensions (lignes, colonnes)
print(df.shape)
```

Listing 2 – Explorer un DataFrame

2.3.2 Sélection de données

```python
# Selectionner une colonne
prix = df['Prix']

# Selectionner plusieurs colonnes
selection = df[['Materiau', 'Prix']]

# Selectionner des lignes par index
```

```python
premiere_ligne = df.iloc[0]

# Filtrer des lignes selon une condition
materiaux_chers = df[df['Prix'] > 100]
```

Listing 3 – Selectionner des donnees

2.3.3 Calculs statistiques

```python
# Moyenne
prix_moyen = df['Prix'].mean()

# Mediane
prix_median = df['Prix'].median()

# Minimum et maximum
prix_min = df['Prix'].min()
prix_max = df['Prix'].max()

# Somme
prix_total = df['Prix'].sum()

# Ecart-type
ecart_type = df['Prix'].std()

print(f"Prix moyen : {prix_moyen:.2f}")
print(f"Prix total : {prix_total:.2f}")
```

Listing 4 – Statistiques sur les donnees

2.3.4 Tri et classement

```python
# Trier par prix croissant
df_trie = df.sort_values('Prix')

# Trier par prix decroissant
df_trie_desc = df.sort_values('Prix', ascending=False)

# Trier par plusieurs colonnes
df_multi = df.sort_values(['Unite', 'Prix'])
```

Listing 5 – Trier les donnees

2.3.5 Export de données

```python
# Exporter en CSV
df.to_csv('materiaux.csv', index=False, encoding='utf-8')

# Exporter en Excel (necessite openpyxl)
df.to_excel('materiaux.xlsx', index=False)

# Lire un CSV
df_lecture = pd.read_csv('materiaux.csv')
```

Listing 6 – Exporter un DataFrame

2.4 Visualisation avec Matplotlib

```python
import matplotlib.pyplot as plt

# Configuration pour affichage correct des accents
plt.rcParams['font.sans-serif'] = ['DejaVu Sans']

# Graphique en barres
df.plot(x='Materiau', y='Prix', kind='bar',
        title='Prix des materiaux',
        xlabel='Materiau',
        ylabel='Prix (euros)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('prix_materiaux.png')
plt.show()

# Diagramme circulaire
df.set_index('Materiau')['Prix'].plot(kind='pie',
                                      autopct='%1.1f%%')
plt.title('Repartition des couts')
plt.show()
```

Listing 7 – Graphiques simples

3 Exercices pratiques (1h40)

3.1 Exercice 1 : Scraper et analyser le catalogue MarketBTP (60 min)

Site à scraper : http://www.malomatique.free.fr/MarketBTP/index.html

3.1.1 Objectifs

Créez un script `exercice1.py` qui :

1. Scrape les 3 pages du catalogue MarketBTP
2. Extrait pour chaque produit :
   - Le type de produit (Béton, Acier, Maçonnerie, etc.)
   - Le nom du produit
   - Le fournisseur
   - Le prix (valeur numérique)
   - L'unité (m^3, tonne, m^2, ml, unité)
   - Les caractéristiques techniques
   - Le délai de livraison
   - La région
   - La note (nombre d'étoiles)
   - La disponibilité
3. Stocke les données dans un DataFrame Pandas
4. Effectue une analyse complète :
   - Affiche les statistiques descriptives
   - Identifie les produits les plus chers et les moins chers
   - Calcule le prix moyen par catégorie
   - Calcule le prix moyen par fournisseur

— Analyse les délais de livraison
— Filtre les produits disponibles uniquement

5. Crée des visualisations :
   — Graphique en barres des 10 produits les plus chers
   — Histogramme de distribution des prix
   — Graphique de répartition par catégorie
   — Comparaison des prix moyens par fournisseur
6. Exporte les données en CSV

3.1.2 Code de départ

```python
import requests
from bs4 import BeautifulSoup
import pandas as pd
import matplotlib.pyplot as plt
import re
import time

# Configuration matplotlib pour accents
plt.rcParams['font.sans-serif'] = ['DejaVu Sans']

def extraire_prix(texte_prix):
    """
    Extrait le prix numerique d'une chaine
    Ex: '95.50 euros/m3' -> 95.50
    """
    match = re.search(r'(\d+\.?\d*)', texte_prix)
    if match:
        return float(match.group(1))
    return 0.0

def compter_etoiles(texte_note):
    """
    Compte le nombre d'etoiles pleines
    Ex: 5 etoiles pleines, 0 vides -> 5
    """
    # Compter les etoiles pleines (caractere Unicode U+2605)
    return texte_note.count('\u2605')

def scraper_page(url):
    """Scrape une page et retourne la liste des produits"""
    response = requests.get(url)

    if response.status_code != 200:
        print(f"Erreur : {response.status_code}")
        return []

    soup = BeautifulSoup(response.text, 'html.parser')
    produits = []

    # Trouver tous les produits
    cards = soup.find_all('div', class_='product-card')

    for card in cards:
        produit = {
            'Type': '',
            'Nom': '',
            'Fournisseur': '',
            'Prix': 0.0,
```

```
            'Unite': '',
            'Note': 0,
            'Disponibilite': '',
            'Delai': '',
            'Region': ''
        }

        # A completer : extraire toutes les donnees
        # Extraire le type
        type_elem = card.find('span', class_='product-type')
        if type_elem:
            produit['Type'] = type_elem.text.strip()

        # A completer : extraire les autres champs...

        produits.append(produit)

    return produits

def analyser_donnees(df):
    """Analyse statistique des donnees"""
    print("\n" + "="*60)
    print("ANALYSE DES DONNEES - CATALOGUE MARKETBTP")
    print("="*60)

    # Informations generales
    print(f"\nNombre total de produits : {len(df)}")
    print(f"Nombre de categories : {df['Type'].nunique()}")
    print(f"Nombre de fournisseurs : {df['Fournisseur'].nunique()}")

    # Statistiques sur les prix
    print("\n--- STATISTIQUES DES PRIX ---")
    print(df['Prix'].describe())

    # A completer : autres analyses...

def visualiser_donnees(df):
    """Cree des graphiques"""

    # 1. Top 10 des produits les plus chers
    plt.figure(figsize=(12, 6))
    top10 = df.nlargest(10, 'Prix')
    plt.barh(range(len(top10)), top10['Prix'], color='steelblue')
    plt.yticks(range(len(top10)), top10['Nom'], fontsize=9)
    plt.xlabel('Prix (euros)', fontsize=12)
    plt.title('Top 10 des produits les plus chers',
              fontsize=14, fontweight='bold')
    plt.tight_layout()
    plt.savefig('top10_produits.png', dpi=300)
    plt.show()

    # A completer : autres graphiques...

# Programme principal
def main():
    print("="*60)
    print("COLLECTEUR ET ANALYSEUR MARKETBTP")
    print("="*60)

    base_url = 'http://www.malomatique.free.fr/MarketBTP/'
    pages = ['index.html', 'page-2.html', 'page-3.html']
    tous_les_produits = []

```

```python
    # Scraping des 3 pages
    for i, page in enumerate(pages, 1):
        url = base_url + page
        print(f"\nScraping page {i}...")
        produits = scraper_page(url)
        tous_les_produits.extend(produits)
        time.sleep(1)  # Pause pour ne pas surcharger le serveur

    # Conversion en DataFrame
    df = pd.DataFrame(tous_les_produits)

    print(f"\nTotal de produits collectes : {len(df)}")

    # Nettoyage
    df = df[df['Prix'] > 0]

    # Analyse
    analyser_donnees(df)

    # Visualisation
    visualiser_donnees(df)

    # Export CSV
    df.to_csv('marketbtp_analyse.csv', index=False, encoding='utf-8')
    print("\nDonnees exportees dans 'marketbtp_analyse.csv'")

if __name__ == "__main__":
    main()
```

Listing 8 – exercice1.py - Structure de base