

Projet : Utilisez les bases de Python pour l'analyse de marché

Client : analyste marketing chez Books Online

## 1. Description du projet :

Les analystes marketing chez Books Online, une importante librairie en ligne spécialisée dans les livres d'occasion, dans le cadre de leurs fonctions, suivent manuellement les prix des livres d'occasion sur les sites web de leurs concurrents, ce qui se révèle fastidieux.

Le projet consiste à automatiser ces analyses pour extraire les informations tarifaires d'un revendeur de livres en ligne [Books to Scrape](#), qui se déclinera sous la forme d'une application exécutable à la demande visant à récupérer diverses données, dont les prix, au moment de son exécution, et ceci via un programme (un scraper) développé en Python.

## 2. environnement à utiliser : Python

## 3. Pré-requis

3.1. Python : langage de programmation interprété et orienté objet

3.2. Jupyter notebook : interface de programmation interactive (sections en langage naturel et des sections en langage informatique)

## 4. Installation :

4.1. Python : langage de programmation interprété et orienté objet.

Installer Python sous Windows, sous Mac et Linux Python est déjà installé.

Sous Windows : sous internet aller sur le site de Python à l'adresse [python.org/downloads](https://python.org/downloads) pour installer Python

Installer la version 3.x.x. (Cochez la case Add Python 3.5 to PATH (Ajouter Python 3.5 au PATH pour exécuter Python directement depuis l'invite de commandes)).

4.2. Jupyter notebook : interface de programmation interactive (sections en langage naturel et des sections en langage informatique). Pour lancer Jupyter tapez jupyter notebook sous l'invite de commande du PC.

## 5. ETL : extract transform load

### 5.1. Extraction des données du site : import requests

Dans Python, la bibliothèque\* requests récupère la page web du site analysé :

```
url = 'http://books.toscrape.com'
```

(\*ensemble logiciel de modules (classes (types d'objets), fonctions, constantes, ...)

### 5.2. Transformation : from bs4 import BeautifulSoup

bs4 est une bibliothèque Python d'analyse syntaxique de documents HTML et XML.

Les données de la page web étudiée (un livre particulier par ex) sont récupérées ou parsées (parcourir le contenu d'une page web pour en extraire des éléments) et les informations obtenues peuvent être traitées et/ou sauvegardées.

### 5.3. Load :

#### 5.3.1.import csv

Création d'un fichier csv par catégorie de livres (50 catégories sur le site web) contenant n livres avec les données à analyser

- product\_page\_url
- universal\_product\_code (upc)
- title
- price\_including\_tax
- price\_excluding\_tax
- number\_available
- product\_description
- category
- review\_rating
- image\_url

#### 5.3.2. création d'un fichier image.png par de la page de couverture de chaque livre (1000 livres sur le site web)

## 6. Lancement du programme

6.1. Lancer le programme Python : Python Projet 2 - analyse marketing Books Online

6.2. Le programme Python accède au site web <https://books.toscrape.com>

6.3. Import des bibliothèques pour :

6.3.1. Accéder à une page web

6.3.2. Parser une page web : récupérer les données livres avec les balises html

6.3.3. Créer un fichier .png : photo de la page de couverture du livre (1000 livres toute catégorie comprise)

6.3.4. Créer un fichier .csv : 1 fichier .csv par catégorie (50 catégories), chaque ligne du fichier .csv correspond à un livre avec ses données (titre, prix, ...)

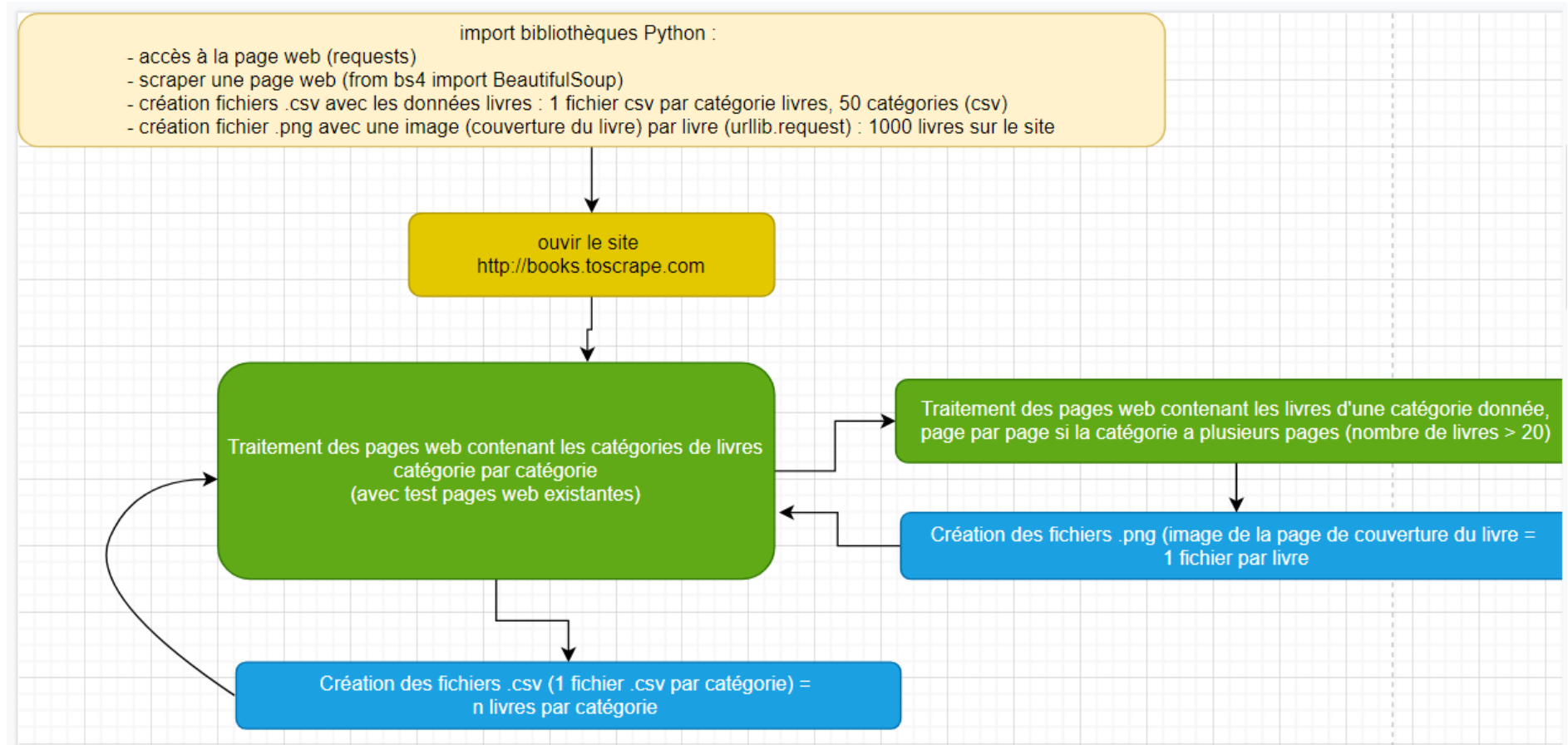
6.4. Le programme Python contient un programme principal qui appelle des fonctions de traitement des données

6.5. Il parcourt toutes les pages web catégorie par catégorie

6.6. Il crée un fichier nom\_catégorie.csv par catégorie et dans chaque fichier .csv il mémorise les données de chaque livre (une ligne d'un fichier csv contient les données d'un livre pour une catégorie donnée) : 50 fichiers .csv

6.7. Il crée un fichier .png par livre qui est une photo de la page de couverture du livre : 1000 livres

7. Diagramme avec <https://www.draw.io/>





















































## 8. Déroulé du programme Python avec Excel



















Programme Python 2 : Python Projet 2 - analyse marketing Books Online					
accès au site :	url = 'http://books.toscrape.com/'				
	import requests				
	l'objet response reçoit la réponse du serveur à la demande d'accès au site web : si trouvé code 200 sinon erreur (autre code) avec la fonction get de la bibliothèque requests	response = requests.get(url)			
	l'objet soup permet d'accéder à la page web du site	soup = BeautifulSoup(response.text, 'html.parser')			
	boucle pour accéder à toutes les catégories du site en parcourant les données html de la page web	for categories in soup.find('ul', class_='nav nav-list').find('li').find('ul').find_all('li'):			
		l'objet href pointe sur la 1ère catégorie (fin du lien url) : Travel	href = categories.a.get('href')		
		l'objet url reçoit le lien de la 1ère catégorie pour ouvrir la page web et accéder à tous les livres de cette catégorie : Travel	url = "https://books.toscrape.com/{j}".format(href)		
		l'objet response reçoit la réponse du serveur à la demande d'accès à la page web de la 1ère catégorie Travel : si trouvé code 200 sinon erreur (autre code)	response = requests.get(url) # 1ère catégorie		
		l'objet soup permet d'accéder à la page web de la 1ère catégorie			
		lancement de la fonction qui va traiter tous les livres de la 1ère catégorie	traitement_page()		
		boucle pour pointer sur chaque url livre (balise" html "h3") mémorisé dans l'objet links	link = soup.find_all('h3')		
		lancement de la fonction qui va traiter les données livre par livre de la 1ère catégorie	livres(links)		
			on ouvre la page web du 1er livre Si la page existe	url = links response = requests.get(url)	
			mémorisation des données livre à enregistrer dans un fichier csv en parcourant avec BeautifulSoup (soup.find et les balises" html associée à chaque donnée) Transformation des données en fonction de ce qui est souhaité avoir dans le fichier csv	product_page_url = url title = soup.find('div', {'class':'col-sm-6 product_main'}).find('h1').text title = title.replace(':', '') universal_product_code = product_information[0].text # code UPC price_excluding_tax = product_information[2].text.replace('Â€', '') price_including_tax = product_information[3].text.replace('Â€', '') number_available = product_information[5].text product_description = " soup = BeautifulSoup(response.content) product_description = soup.find_all('p')[3].text transTable = product_description.maketrans("àâäëèëïöüüüç"---lR\$%^&'"," "aaaaeeiiouuauc  ", "") product_description = product_description.translate(transTable) ( .....) category = soup.find('ul', {'class':'breadcrumb'}).find_all('a')[2].text review_rating = soup.find_all('tr')[6].find('td').text nbre_ettoile = soup.find('div', {'class':'col-sm-6 product_main'}).find_all('p')[2] ( .....) image_url ( .....)	
			lancement de la fonction qui va créer une image (fichier .png) par livre avec 2 paramètres : l'url de l'image et le titre du livre	export_image(image_url, title)	
				ouverture de l'url de l'image test de l'existence de cette page web Transformation de l'url en fonction de ce qui est souhaité et accepté en nom de fichier windows	response = urllib.request.urlopen(image_url) image = response.read()
				création du fichier .png	with open('nom_image.png', 'wb') as file: file.write(image)
			l'objet final_result enregistre les données du livre en cours traité	final_result	
			l'objet list_results concatène toutes les lignes livres d'une catégorie donnée	list_results.append(final_result)	
		Une catégorie peut contenir de 1 à n livres, chaque page web d'une catégorie donnée peut contenir de 1 à 20 livres et donc plusieurs pages web si le nombre de livres est > 20. En bas et à droite de chaque page un bouton "next" est présent si nombre de livres > 20 qui est enregistré dans l'objet page_suivante	page_suivante = soup.find('li', class_='next')		
		Traitement des livres de la page suivante de la catégorie en cours si elle existe	while page_suivante is not None:		
		l'objet complément permet de construire l'url de la page suivante	complément = "page-"+str(no_page)+"_html"		
		l'objet response reçoit la réponse du serveur	url = url.replace(index, complément) response = requests.get(url)		
		l'objet soup permet d'accéder à la page web du site	soup = BeautifulSoup(response.text, 'html.parser')		
		lancement de la fonction qui va traiter tous les livres de la page suivante	traitement_page()		
		lancement de la fonction qui va créer le fichier csv avec toutes les lignes d'une catégorie donnée	export_csv()		

## 9. Le programme Python génère au final :



















### 9.1. Les 50 fichiers .csv : 1 fichier .csv par catégorie

 academic_40.csv	 health_47.csv	
 add-a-comment_18.csv	 historical_42.csv	
 adult-fiction_29.csv	 historical-fiction_4.csv	
 art_25.csv	 history_32.csv	
 autobiography_27.csv	 horror_31.csv	
 biography_36.csv	 humor_30.csv	
 business_35.csv	 music_14.csv	
 childrens_11.csv	 mystery_3.csv	
 christian_43.csv	 new-adult_20.csv	 science_22.csv
 christian-fiction_34.csv	 nonfiction_13.csv	 science-fiction_16.csv
 classics_6.csv	 novels_46.csv	 self-help_41.csv
 contemporary_38.csv	 paranormal_24.csv	 sequential-art_5.csv
 crime_51.csv	 parenting_28.csv	 short-stories_45.csv
 cultural_49.csv	 philosophy_7.csv	 spirituality_39.csv
 default_15.csv	 poetry_23.csv	 sports-and-games_17.csv
 erotica_50.csv	 politics_48.csv	 suspense_44.csv
 fantasy_19.csv	 psychology_26.csv	 thriller_37.csv
 fiction_10.csv	 religion_12.csv	 travel_2.csv
 food-and-drink_33.csv	 romance_8.csv	 womens-fiction_9.csv
		 young-adult_21.csv

## 9.2. Extrait des 1000 fichiers .png : 1 fichier image par livre dans l'ordre alphabétique

-  (Un)Qualified How God Uses Broken People to Do Big Things.png
-  1st to Die .png
-  8 Keys to Mental Health Through Exercise.png
-  10% Happier How I Tamed the Voice in My Head Reduced Stress Without Losing My Edge and Found Self-Help That Actually Works.png
-  10-Day Green Smoothie Cleanse Lose Up to 15 Pounds in 10 Days!.png
-  11.png
-  13 Hours The Inside Account of What Really Happened In Benghazi.png
-  23 Degrees South A Tropical Tale of Changing Whether....png
-  32 Yolks.png
-  1000 Places to See Before You Die.png
-  1491 New Revelations of the Americas Before Columbus.png
-  A Brush of Wings .png
-  A Clash of Kings .png
-  A Court of Thorns and Roses .png
-  A Distant Mirror The Calamitous 14th Century.png
-  A Feast for Crows .png
-  A Fierce and Subtle Poison.png
-  A Flight of Arrows .png

■ ■ ■

-  Will Grayson Will Grayson .png
  -  Will You Won't You Want Me .png
  -  William Shakespeare's Star Wars Verily A New Hope .png
  -  Without Borders .png
  -  Without Shame.png
  -  Wonder Woman Earth One Volume One .png
  -  World War Z An Oral History of the Zombie War.png
  -  World Without End .png
  -  Worlds Elsewhere Journeys Around Shakespeare's Globe.png
  -  Wuthering Heights.png
  -  Y The Last Man Vol. 1 Unmanned .png
  -  You .png
  -  You Are a Badass How to Stop Doubting Your Greatness and Start Living an Awesome Life.png
  -  You Are What You Love The Spiritual Power of Habit.png
  -  You can't bury them all Poems.png
  -  Zealot The Life and Times of Jesus of Nazareth.png
  -  Zero History .png
  -  Zero to One Notes on Startups or How to Build the Future.png
-