Lacime Melvin

Compte rendu de stage première année BTS SIO en l'entreprise

Entreprise d'accueil : Kévin Hutin

L'entreprise Kévin Hutin travaille dans le secteur informatique dans la conception de sites web comme le commerce en ligne et la création de logiciel dans la crypto-monnaie par exemple.

Le Web Scraping

Le Web Scraping :

Le Web Scraping consiste à récupérer des données déjà existantes de façon automatique à l'aide d'un programme (ici codé en python) sur un site web pour pouvoir les manipuler, les modifier ou les introduire dans un autre site ou une base de données par exemple. Cette technique est souvent utilisée dans le cadre d'une veille concurrentielle et dans les sites web de commerce en ligne.

Les différents logiciels :

Les logiciels utilisés seront Visual Studio Code pour éditer nos programmes avec des extensions afin de pouvoir coder en Python, Git Bash pour réaliser et travailler dans un environnement virtuel avec les librairies BeautifulSoup4 et Request.

Contexte

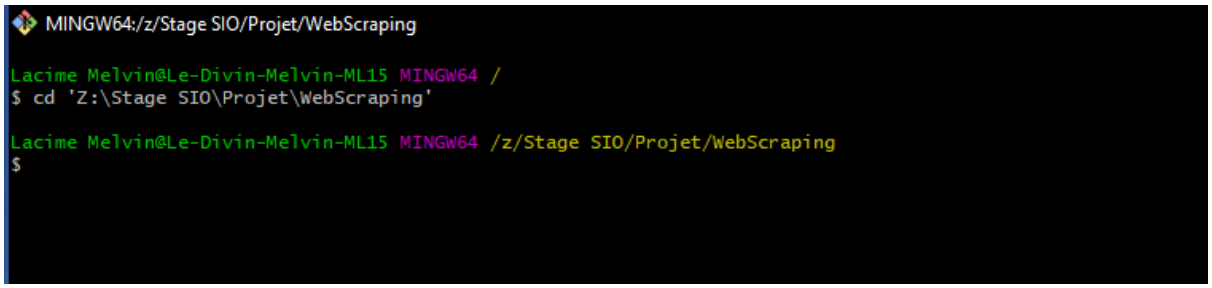
Au début de mon stage, j'ai eu le choix entre différents projets qui étaient de concevoir un site en python avec une base de données ou bien de faire du web scraping sur différents sites webs. J'ai pris l'option de faire du web scraping. Le but du stage était donc de réaliser un programme codé en python qui récupère des données sur un site web.

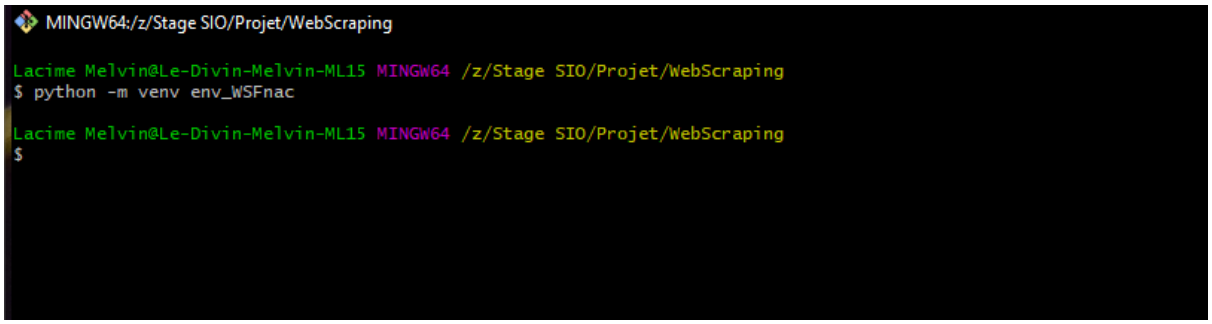
Comme je n'avais jamais pratiqué de web scraping auparavant, j'ai été aidé par mon tuteur de stage et comme il aimait souvent le préciser : "Google est ton ami ". J'ai donc fréquemment fait des recherches pour comprendre le code et les différents logiciels et corriger mes erreurs.

Installation et création environnement virtuel

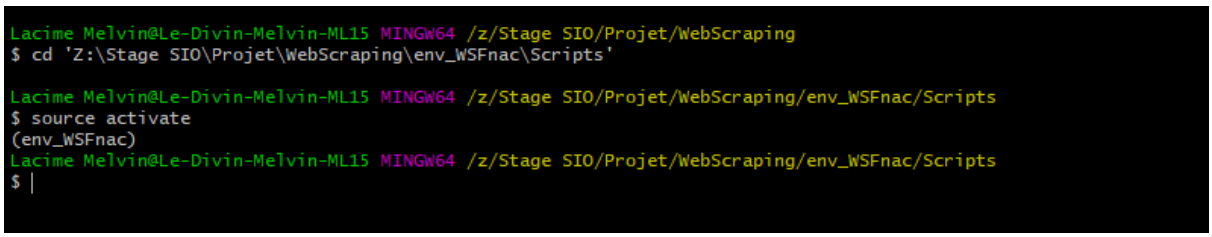
La première étape est de créer un environnement virtuel pour avoir un espace de travail avec le code et les librairies nécessaires au développement.

Il faut choisir un répertoire de destination, puis créer l'environnement.





Ensuite, activer cet environnement.



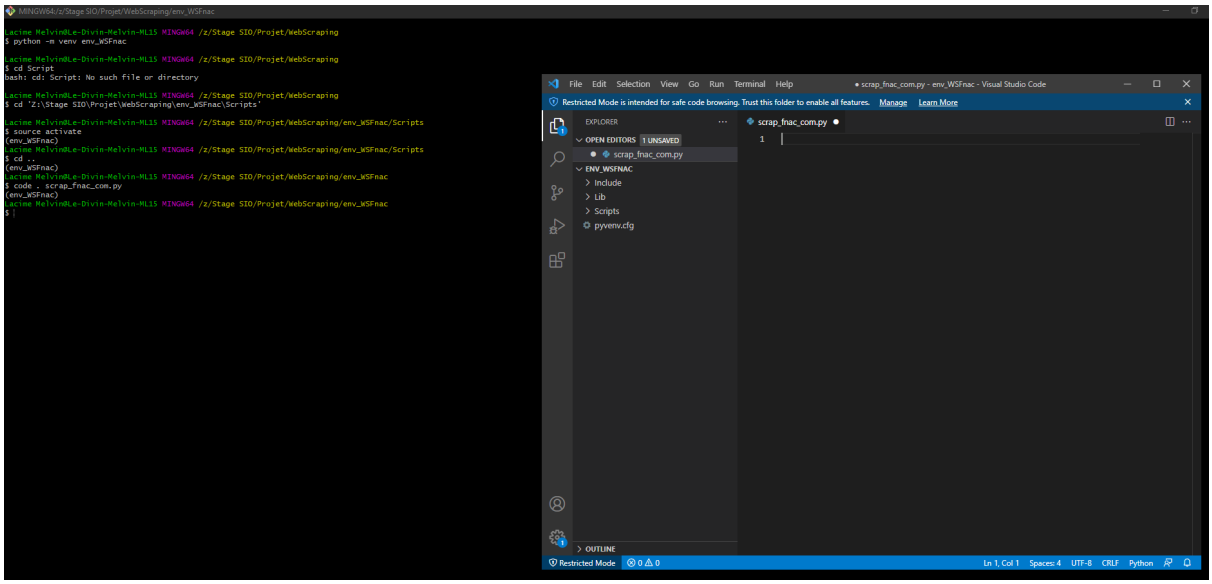
Pour pouvoir nous repérer et savoir si notre environnement est actif, il y a une petite indication entre parenthèses (ici **(env_WSFnac)**).

```
Lacime Melvin@Le-Divin-Melvin-ML15 MINGW64 /z/Stage SIO/Projet/WebScraping/env_WSFnac/Scripts
$ source activate
(env_WSFnac)
Lacime Melvin@Le-Divin-Melvin-ML15 MINGW64 /z/Stage SIO/Projet/WebScraping/env_WSFnac/Scripts
$ cd ..
(env_WSFnac)
Lacime Melvin@Le-Divin-Melvin-ML15 MINGW64 /z/Stage SIO/Projet/WebScraping/env_WSFnac
```

Par la suite, on crée le fichier qui contiendra notre code python.

```
(env_WSFnac)
Lacime Melvin@Le-Divin-Melvin-ML15 MINGW64 /z/Stage SIO/Projet/WebScraping/env_WSFnac
$ code . scrap_fnac_com.py
(env_WSFnac)
Lacime Melvin@Le-Divin-Melvin-ML15 MINGW64 /z/Stage SIO/Projet/WebScraping/env_WSFnac
$
```

Cette commande ouvre directement Visual Studio Code avec le nouveau fichier créé.



Et enfin, on installe les librairies.

```
(env_WSFnac)
Lacime Melvin@Le-Divin-Melvin-ML15 MINGW64 /z/Stage SIO/Projet/WebScraping/env_WSFnac
$ pip install beautifulsoup4
Collecting beautifulsoup4
  Downloading beautifulsoup4-4.10.0-py3-none-any.whl (97 kB)
Collecting soupsieve>1.2
  Using cached soupsieve-2.2.1-py3-none-any.whl (33 kB)
Installing collected packages: soupsieve, beautifulsoup4
Successfully installed beautifulsoup4-4.10.0 soupsieve-2.2.1
WARNING: You are using pip version 21.1.1; however, version 21.2.4 is available.
You should consider upgrading via the 'z:\stage sio\projet\webscraping\env_wsfnac\scripts\python.exe -m pip install --upgrade pip' command.
(env_WSFnac)
Lacime Melvin@Le-Divin-Melvin-ML15 MINGW64 /z/Stage SIO/Projet/WebScraping/env_WSFnac
$ pip install requests
Collecting requests
  Downloading requests-2.26.0-py2.py3-none-any.whl (62 kB)
Collecting charset-normalizer~=2.0.0
  Downloading charset_normalizer-2.0.4-py3-none-any.whl (36 kB)
Collecting idna<4,>=2.5
  Downloading idna-3.2-py3-none-any.whl (59 kB)
Collecting urllib3<1.27,>=1.21.1
  Using cached urllib3-1.26.6-py2.py3-none-any.whl (138 kB)
Collecting certifi>=2017.4.17
  Using cached certifi-2021.5.30-py2.py3-none-any.whl (145 kB)
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
Successfully installed certifi-2021.5.30 charset-normalizer-2.0.4 idna-3.2 requests-2.26.0 urllib3-1.26.6
WARNING: You are using pip version 21.1.1; however, version 21.2.4 is available.
You should consider upgrading via the 'z:\stage sio\projet\webscraping\env_wsfnac\scripts\python.exe -m pip install --upgrade pip' command.
(env_WSFnac)
Lacime Melvin@Le-Divin-Melvin-ML15 MINGW64 /z/Stage SIO/Projet/WebScraping/env_WSFnac
$
```

Le programme :

Pour faire le lien avec Git Bash, on importe les librairies.

Z: > Stage SIO > Projet > WebScraping > WSFnac > scrap_fnac_com.py

```python
# coding: utf-8

# Importation des librairies

from bs4 import BeautifulSoup
import requests
import time
import random
```

Puis on définit, l'URL du site où l'on veut collecter les données.

```python

# Définition de l'URL

urlpage = 'https://www.fnac.com/n396208/Minecraft/Livres-Minecraft'
```

Le header permet d'identifier les différentes caractéristiques du navigateur utilisé.

```python
# Requet du site et retourner le code HTML dans la variable 'page'

header_default = {

    'authority': 'livre.fnac.com',
    'cache-control': 'max-age=0',
    'sec-ch-ua': '^\\^',
    'sec-ch-ua-mobile': '?0',
    'upgrade-insecure-requests': '1',
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36',
    'accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;
    'sec-fetch-site': 'same-origin',
    'sec-fetch-mode': 'navigate',
    'sec-fetch-user': '?1',
    'sec-fetch-dest': 'document',
    'referer': 'https://livre.fnac.com/a14036396/Minecraft-Tome-3-Chroniques-de-l-Epee-de-bois-3-Grand-plongeon-roman-Minecraft-Nick-Eliopulos',
    'accept-language': 'fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7',
    'cookie': 'ORGN=FnacAff; OrderInSession=0; fd_vls_address=None; LSI=99=AAB^#VCuTS^#A1i2s^|B; UID=0c71b17d6-2c73-4328-a6df-d3113238198c; SID=0

    }

page = requests.get(urlpage, headers = header_default)

print(page.status_code)
```

Beautifulsoup permet de traiter les données que le programme va récupérer et les stock

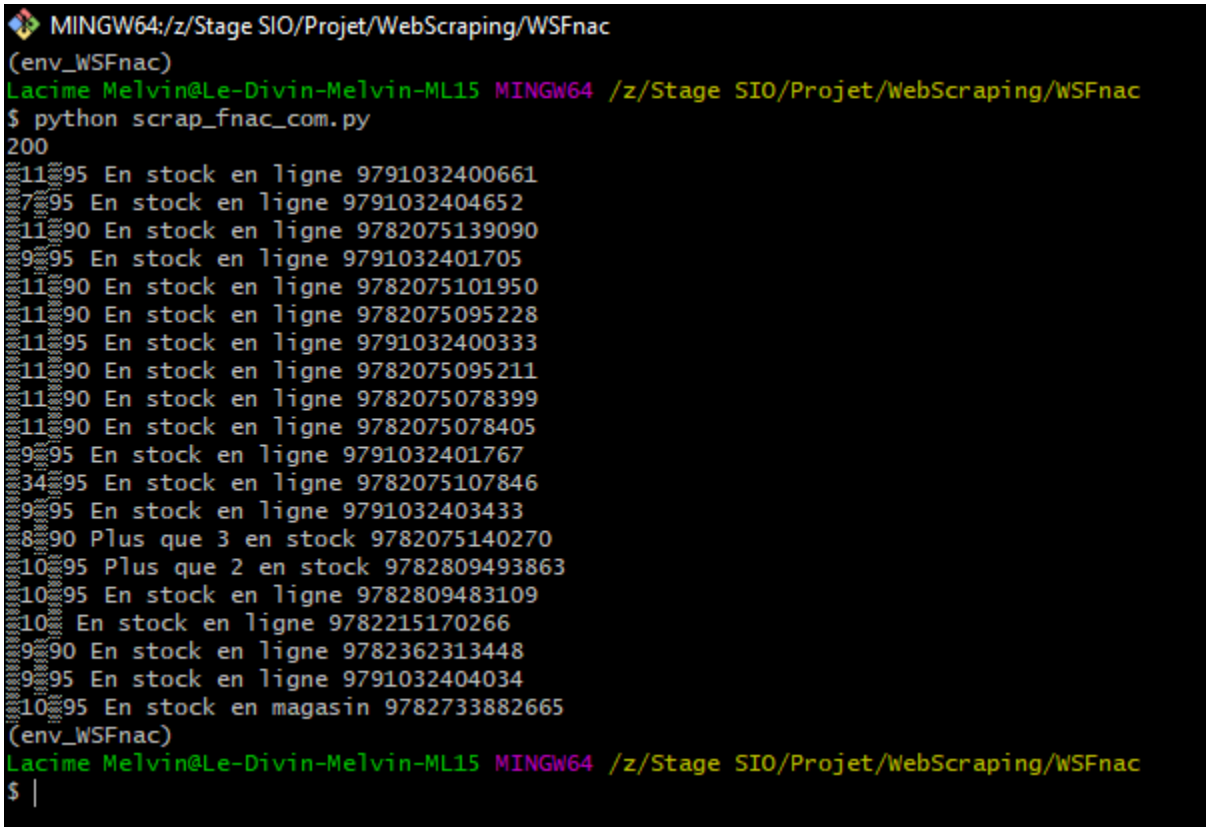
```
# Analyser le code avec bs4 et le stocker dans 'soup'

soup = BeautifulSoup(page.text, 'html.parser')

title_tags = soup.select('a.Article-title')

for title_tag in title_tags:

    time.sleep(random.uniform(0.5, 1))

    link = title_tag.get('href')
    page = requests.get(link, headers = header_default)
    soup = BeautifulSoup(page.text, 'html.parser')

    price_tag = soup.select_one('span.f-productOffers-tabLabel--price')
    availability_tag = soup.select_one('p .f-buyBox-availabilityStatus-available')
    ean_tag = soup.select_one('[data-ean13]')

    try :
        print(price_tag.getText(), availability_tag.findParent().getText(), ean_tag.get('data-ean13'))
    except :
        print(availability_tag)
```

Et enfin, on peut afficher les valeurs collectées.

```
    try :
        print(price_tag.getText(), availability_tag.findParent().getText(), ean_tag.get('data-ean13'))
    except :
        print(availability_tag)
```

Une fois le code terminé, pour lancer le programme on exécute sur GitBash la commande : python nom_du_fichier.py .

Le numéro 200 signifie que le lien avec le site sur lequel on récupère les données fonctionne.



Conclusion :

Grâce à ce stage, j'ai pu apprendre de nouvelles choses tout en améliorant mes compétences en matière de programmation. Comprendre comment fonctionne la résolution des multiples erreurs. Je ne sais pas encore si je souhaite continuer dans ce secteur d'activité mais cela reste très intéressant à développer.