



Stockage d'un grand nombre d'articles, dans un fichier excel de format csv, pour les exploiter facilement.

(From unstructured data into structured data and then storing it for further analysis).

Description de ce projet:

L'objectif de ce projet est d'écrire un programme dont l'utilité est de stocker un nombre grand nombre d'articles (qu'on fixera au programme) dans un fichier csv en un clic, juste en donnant les liens des site web comme input.

Les sites web et leurs liens seront stockés dans un fichier json (un format qui stock des informations structurés). On verra ça ci-dessous.

- Les articles doivent être stockés de façon structuré, pour pouvoir les exploiter facilement.

Le langage de programmation choisi pour accomplir cette tâche est python.

Chacun des siteweb contient des articles qu'on va extraire à l'aide de la bibliothèque newspaper de python, pour plus d'informations sur cette bibliothèque consultez le lien ci-dessous :

<https://newspaper.readthedocs.io/en/latest/>

Pour vous aider à comprendre le code :

Liste_des_articles : c'est une liste vide initialisé au début du programme dans laquelle on va stocker les différents articles scrappés.

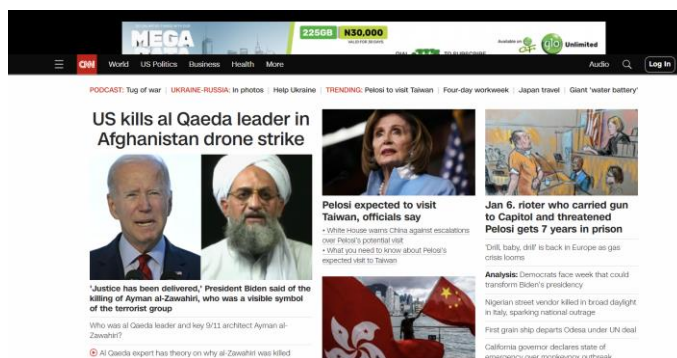
Chaque article est un dictionnaire dont la forme est la suivante :

```
{'title': '---', 'authors': '---', 'text': "", 'top_image': '---', 'movies': '---', 'link': "", 'published': '--'}
-}
```

Deux sites web pour le test, on rajoutera d'autres par la suite :

First web site :

<https://www.the-scientist.com/>



Cable News Network est une chaîne de télévision d'information en continu américaine fondée en 1980 par Ted Turner.

C'est parti :

- 1. Importation des bibliothèques requises.

```
In [25]: import feedparser as fp
import numpy as np
import json
import newspaper
from newspaper import Article
from time import mktime
from datetime import datetime
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import csv
```

- 2. Fixer le nombre d'articles à extraire, et initialiser des structures de données dans lesquelles on va stocker les articles, ainsi que l'ouverture du fichier json contenant les sites web.

```
In [43]: nb_articles = 100
list_des_articles = []

data = {}
data['journal'] = {}

# Importer le fichier json contenant les sites web
with open('sites_web.json') as data_file:
    sites_web = json.load(data_file)
```

```
In [44]: sites_web
```

```
Out[44]: {'the-scientist': {'link': 'https://www.the-scientist.com/'},
          'CNN': {'link': 'https://edition.cnn.com/?refresh=1'}}
```

Ici j'ai fixé le nombre d'articles maximum à stocker de chaque site web, on peut l'augmenter.

On a donné 2 sources de données au programme (deux sites web), donc on aura 200 articles au total.

Rq : en augmentant ce nombre, le programme prend plus de temps pour s'exécuter, mais en fin de compte les données seront stockées avec succès :).

Ci-dessous le fichier json :

```
jupyter urls.json a few seconds ago
File Edit View Language

1 {
2   "the-scientist": {
3     "link": "https://www.the-scientist.com/"
4   },
5   "CNN": {
6     "link": "https://edition.cnn.com/?refresh=1"
7   }
8 }
```

Pour ajouter d'autres sources vous n'avez qu'à suivre la logique, en rajoutant un autre élément dans ce dictionnaire, ainsi que le lien associé.

```

count = 1
for site_web, value in sites_web.items():
    print("Le site web : ", site_web)
    paper = newspaper.build(value['link'], memoize_articles=False)
    newsPaper = {
        "link": value['link'],
        "articles": []
    }
    noneTypeCount = 0
    for content in paper.articles:
        if count > nb_articles:
            break
        try:
            content.download()
            content.parse()
        except Exception as e:
            print(e)
            print("continuing...")
            continue
        # Again, for consistency, if there is no found publish date the arti
        # After 10 downloaded articles from the same newspaper without publi

        article = {}
        article['title'] = content.title
        article['authors'] = content.authors
        article['text'] = content.text
        article['top_image'] = content.top_image
        article['movies'] = content.movies
        article['link'] = content.url
        article['published'] = content.publish_date
        newsPaper['articles'].append(article)
        list_des_articles.append(article)
        print(count, "article extrait du ", site_web, " à l'aide de la bibli
        count = count + 1
        #noneTypeCount = 0
    count = 1
    data['journal'][site_web] = newsPaper

```

Dans ce code, on a utilisé la bibliothèque newspaper de python, pour extraire les articles et les analyser, (scapping articles, parsing articles), the data parser a la capacité d'identifier la date de publication, l'auteur, le text, l'image, ...

Ce qui va nous aider à les stocker de façon structuré.

Extraction de 100 articles du premier site, et l'affichage de leurs liens:

```
#none|typeCount = 0
count = 1
data['journal'][site_web] = newsPaper
```

Le premier site web : the-scientist

1 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/researchers-in-brazil-struggle-to-get-solid-covid-19-death-counts-67609>

2 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/spending-bill-boosts-us-science-budget-s-unlocks-gun-research-66855>

3 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/metabolic-biomarker-score-may-predict-death-in-next-510-years-66304>

4 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/trial-in-africa-probes-antibiotics-effects-on-child-mortality-66270>

5 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/government-scientists-launch-investigation-into-whale-deaths-65962>

6 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/genetic-mutation-that-prevents-hiv-infection-tied-to-earlier-death-65960>

7 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/image-of-the-day/image-of-the-day-beached-birds-65949>

8 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/pro-football-players-die-at-a-higher-rate-than-pro-baseball-players-65941>

9 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/cdc-reports-about-700-pregnancy-related-deaths-each-year-in-the-us-65848>

10 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/new-study-questions-whether-death-rat>

<https://www.the-scientist.com/the-literature/meet-the-algae-that-went-from-male-female-to-hermaphroditic-70266>

95 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/foundations/handmade-hemoglobin-1912-2012-70222>

96 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/multimedia/august-2022-interactive-crossword-puzzle-70285>

97 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/dolphins-may-remember-personal-experiences-70288>

98 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/gaia-theorist-james-lovelock-dies-at-103-70289>

99 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/small-wearable-patch-performs-continuous-ultrasound-70287>

100 article extrait du the-scientist à l'aide de la biblio newspaper, lien: <https://www.the-scientist.com/news-opinion/how-wandering-nuclei-shape-developing-embryos-70281>

Extraction des 100 articles du deuxième site web CNN :

Le premier site web : CNN

1 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://edition.cnn.com/business/media>

2 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://edition.cnn.com/travel/news>

3 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://edition.cnn.com/2022/07/31/africa/sudan-protests-military-rule-intl-hnk/index.html>

4 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://edition.cnn.com/2022/07/31/us/lawrence-rudolph-wife-killed-zambia-cec/index.html>

5 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://edition.cnn.com/2022/07/29/africa/sudan-russia-gold-investigation-cmd-intl/index.html>

6 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://edition.cnn.com/2022/07/28/africa/nigeria-opposition-senators-buhari-intl/index.html>

93 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://www.cnn.com/2022/05/11/sport/zhou-jihong-diving-fina-spt-intl/index.html>

94 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://www.cnn.com/2022/02/21/sport/winter-olympics-elite-wealthy-intl-spt/index.html>

95 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://www.cnn.com/2022/01/29/sport/university-of-michigan-robert-anderson-victims-intl-spt/index.html>

96 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://edition.cnn.com/interactive/2021/07/sport/marathon-tokyo-2020-spt-intl/>

97 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://www.cnn.com/2021/06/28/sport/navid-afkari-cmd-spt-intl/index.html>

98 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://www.cnn.com/2022/02/04/sport/gallery/opening-ceremony-beijing-winter-olympics/index.html>

99 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://www.cnn.com/2022/01/28/sport/gallery/winter-olympics-history/index.html>

100 article extrait du CNN à l'aide de la biblio newspaper, lien: <https://www.cnn.com/2022/01/26/sport/gallery/winter-olympic-athletes-to-watch-beijing/index.html>

2 ème section : Le stockage de ces articles dans un fichier excel de format csv :

```
In [47]: try:
        f = csv.writer(open('Scraped_data_news_output.csv', 'w', encoding='utf-8'))
        f.writerow(['Title', 'Authors', 'Text', 'Image', 'Videos', 'Link', 'Published_Dat
        #print(article)
        for article in list_des_articles:
            title = article['title']
            authors=article['authors']
            text=article['text']
            image=article['top_image']
            video=article['movies']
            link=article['link']
            publish_date=article['published']
            # Add each artist's name and associated link to a row
            f.writerow([title, authors, text, image, video, link, publish_date])
        except Exception as e: print(e)
```

En exécutant :

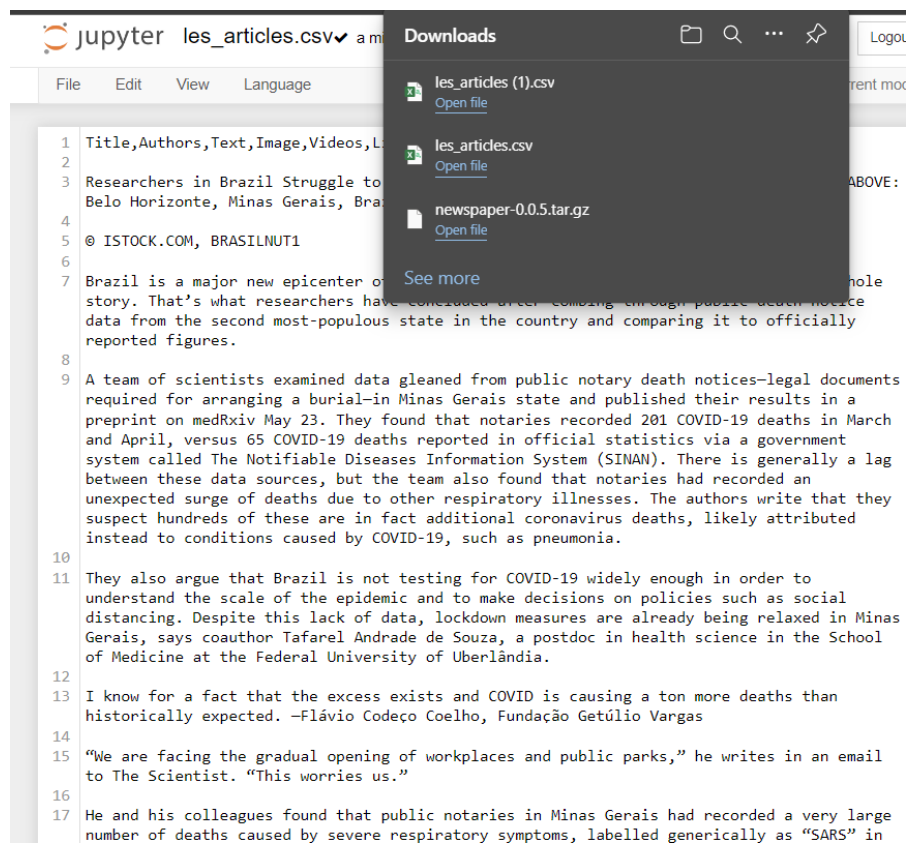
Un nouveau fichier qui apparait, et comme vous pouvez l'apercevoir il est vide, car il est en train de se remplir par les données.

 [les_articles.csv](#)

seconds ago

0 B

Après quelques minutes :





The screenshot shows a Jupyter Notebook interface. A file explorer overlay is open, displaying a list of files: 'les_articles (1).csv', 'les_articles.csv', and 'newspaper-0.0.5.tar.gz'. The 'les_articles.csv' file is highlighted. In the background, the Jupyter Notebook is open, showing a code cell with the following text:

```
1 Title,Authors,Text,Image,Videos,L
2
3 Researchers in Brazil Struggle to
4 Belo Horizonte, Minas Gerais, Bra
5 © ISTOCK.COM, BRASILNUT1
6
7 Brazil is a major new epicenter of
8 story. That's what researchers have
9 concluded after combing through public
10 data from the second most-populous
11 state in the country and comparing it
12 to officially reported figures.
13
14 A team of scientists examined data
15 gleaned from public notary death
16 notices—legal documents required
17 for arranging a burial—in Minas
18 Gerais state and published their
19 results in a preprint on medRxiv
20 May 23. They found that notaries
21 recorded 201 COVID-19 deaths in
22 March and April, versus 65 COVID-19
23 deaths reported in official
24 statistics via a government system
25 called The Notifiable Diseases
26 Information System (SINAN). There
27 is generally a lag between these
28 data sources, but the team also
29 found that notaries had recorded an
30 unexpected surge of deaths due to
31 other respiratory illnesses. The
32 authors write that they suspect
33 hundreds of these are in fact
34 additional coronavirus deaths,
35 likely attributed instead to
36 conditions caused by COVID-19,
37 such as pneumonia.
```


Rq: Vous trouverez le code dans mes dépôts sur dans le lien ci-dessous, ainsi que le fichier json, dans le lien ci-dessous:

<https://github.com/elmoutaquiHicham/articles-scrap/tree/main>

 programme.py	Create programme.py	18 minutes ago
 urls.json	Create urls.json	4 hours ago

Programme.py contient le code :

```
74 lines (64 sloc) | 2.26 KB | Raw | Blame
1
2  ## Importation des bibliothèques.
3  import feedparser as fp
4  import numpy as np
5  import json
6  import newspaper
7  from newspaper import Article
8  import pandas as pd
9  import numpy as np
10 import csv
11
12 nb_articles = 100
13 list_des_articles = []
14
15 data = {}
16 data['journal'] = {}
17
18 # Importer le fichier json contenant les sites web
19 with open('sites_web.json') as data_file:
20     sites_web = json.load(data_file)
21
22 count = 1
23 for site_web, value in sites_web.items():
24     print("Le site web : ", site_web)
25     paper = newspaper.build(value['link'], memoize_articles=False)
26     newsPaper = {
```

Sites_web.json :

```
11 lines (11 sloc) | 195 Bytes
1  {
2      "medscape": {
3          "link": "https://francais.medscape.com/"
4      },
5      "webdm": {
6          "link": "https://www.webmd.com/"
7      },
8      "the-scientist": {
9          "link": "https://www.the-scientist.com/"
10     }
11 }
```