Projet de web scrapping Master 1 Ingenierie de Donnees et Applications

Universite Iba Der Thiam de Thies

@author: Khadim Mbacke NDIAYE

@author: Khadidiatou YADE

C'est quoi le web scrapping?

Le Web scraping (de l'anglais scraping = « gratter/racler ») consiste à extraire des données de sites Internet et à les enregistrer afin de les analyser ou de les utiliser de toute autre façon. Le scraping permet de collecter des informations de nature bien différente. Il peut par ex. s'agir de coordonnées comme des adresses e-mail ou des numéros de téléphone, mais aussi de mots-clés individuels ou d'URL. Ces informations sont alors rassemblées dans des bases de données locales ou des tableaux.

SUJET 7: Ensemble de données de citations (QuotesToScrape) : créez un ensemble de données de citations populaires pour différentes balises en scrapant le site Citations à gratter : http://quotes.toscrape.com

Type de cellule non compatible. Double-cliquez pour examiner/modifier le contenu.

```
#Installer la librarie
!pip install requests ––upgrade ––quiet
```

```
# Importer la librarie
import requests
```

recuperation de l'URL a scrapper

```
topic_url='http://quotes.toscrape.com'
```

Telechargement de la page web en utilisant requests.get

```
response = requests.get(topic_url)
response
```

```
<Response [200]>
```

Acceder au contenu de la page avec la commande .text avec la variable response

```
page_contents=response.text
print(page_contents)
```

```
                <h2>Top Ten tags</h2>

                <span class="tag-item">
                <a class="tag" style="font-size: 28px" href="/tag/love/">love</
                </span>

                <span class="tag-item">
                <a class="tag" style="font-size: 26px" href="/tag/inspirational
                </span>

                <span class="tag-item">
                <a class="tag" style="font-size: 26px" href="/tag/life/">life</
                </span>

                <span class="tag-item">
                <a class="tag" style="font-size: 24px" href="/tag/humor/">humor
                </span>

                <span class="tag-item">
                <a class="tag" style="font-size: 22px" href="/tag/books/">books
                </span>

                <span class="tag-item">
                <a class="tag" style="font-size: 14px" href="/tag/reading/">rea
                </span>
```

```
                <span class="tag-item">
                <a class="tag" style="font-size: 10px" href="/tag/friendship/">
                </span>

                <span class="tag-item">
                <a class="tag" style="font-size: 8px" href="/tag/friends/">frie
                </span>

                <span class="tag-item">
                <a class="tag" style="font-size: 8px" href="/tag/truth/">truth<
                </span>

                <span class="tag-item">
                <a class="tag" style="font-size: 6px" href="/tag/simile/">simil
                </span>


        </div>
    </div>

        </div>
        <footer class="footer">
            <div class="container">
                <p class="text-muted">
                    Quotes by: <a href="https://www.goodreads.com/quotes">GoodR
                </p>
                <p class="copyright">
                    Made with <span class='sh-red'>❤</span> by <a href="https:/
                </p>
            </div>
        </footer>
    </body>
    </html>


print(len(page_contents))

    11010
```

Affichage des 1000 premiere caracteres de la page web

```
page_contents[:1000]

    '<!DOCTYPE html>\n<html lang="en">\n<head>\n\t<meta charset="UTF-8">\n\t<ti
```

Enregistrement du contenu de la page web dans un fichier avec l'extension .html.

```
with open('quotes.toscrape.html', 'w', encoding="utf-8") as file:
    file.write(page_contents)
file
```

```
<_io.TextIOWrapper name='quotes.toscrape.html' mode='w' encoding='utf-8'>
```

Extraire des informations du HTML à l'aide de Beautiful Soup

Pour extraire des informations du code source HTML d'une page Web par programmation, nous pouvons utiliser la bibliothèque Beautiful Soup. Installons la bibliothèque et importons la classe BeautifulSoup depuis le module bs4.

```
# Installation de la librairie
!pip install beautifulsoup4 --upgrade --quiet
```

```
# importation du module BeautifulSoup
from bs4 import BeautifulSoup
```

Lecture du contenu du fichier quotes.toscrape.html et création d'un objet BeautifulSoup pour analyser le contenu.

```
with open('quotes.toscrape.html', 'r') as f:
    html_source = f.read()
```

```
html_source[:1500]
```

```
'<!DOCTYPE html>\n<html lang="en">\n<head>\n\t<meta charset="UTF-8">\n\t<ti
```

```
doc = BeautifulSoup(html_source, 'html.parser')
```

Le titre de la page web

```
title=doc.title
title
```

```
    <title>Quotes to Scrape</title>
```

```
#Affichage des premiéres occurrences de la balises div dans le but de chercher l
top_div=doc.div
top_div
```

```
    <div class="tags">
                Tags:
                <meta class="keywords" content="adulthood,success,value" itempr
    <a class="tag" href="/tag/adulthood/page/1/">adulthood</a>
    <a class="tag" href="/tag/success/page/1/">success</a>
    <a class="tag" href="/tag/value/page/1/">value</a>
    </div>
    </div>
    <div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
    <span class="text" itemprop="text">“It is better to be hated for what you a
    <span>by <small class="author" itemprop="author">André Gide</small>
    <a href="/author/Andre-Gide">(about)</a>
    </span>
    <div class="tags">
                Tags:
                <meta class="keywords" content="life,love" itemprop="keywords"/
    <a class="tag" href="/tag/life/page/1/">life</a>
    <a class="tag" href="/tag/love/page/1/">love</a>
    </div>
    </div>
    <div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
    <span class="text" itemprop="text">“I have not failed. I've just found 10,0
    <span>by <small class="author" itemprop="author">Thomas A. Edison</small>
    <a href="/author/Thomas-A-Edison">(about)</a>
    </span>
    <div class="tags">
                Tags:
                <meta class="keywords" content="edison,failure,inspirational,pa
    <a class="tag" href="/tag/edison/page/1/">edison</a>
    <a class="tag" href="/tag/failure/page/1/">failure</a>
    <a class="tag" href="/tag/inspirational/page/1/">inspirational</a>
    <a class="tag" href="/tag/paraphrased/page/1/">paraphrased</a>
    </div>
    </div>
    <div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
    <span class="text" itemprop="text">“A woman is like a tea bag; you never kn
    <span>by <small class="author" itemprop="author">Eleanor Roosevelt</small>
    <a href="/author/Eleanor-Roosevelt">(about)</a>
    </span>
    <div class="tags">
                Tags:
                <meta class="keywords" content="misattributed-eleanor-roosevelt
    <a class="tag" href="/tag/misattributed-eleanor-roosevelt/page/1/">misattri
    </div>
```

```
</div>
</div>
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">“A day without sunshine is like, you kno
<span>by <small class="author" itemprop="author">Steve Martin</small>
<a href="/author/Steve-Martin">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="humor,obvious,simile" itemprop=
<a class="tag" href="/tag/humor/page/1/">humor</a>
<a class="tag" href="/tag/obvious/page/1/">obvious</a>
<a class="tag" href="/tag/simile/page/1/">simile</a>
</div>
</div>
<nav>
<ul class="pager">
```

```
#Affichage de toute les balises contenant des citations auteurs et tags
articles=doc.find_all(class_='quote')
```

```
articles
```

```
[<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
 <span class="text" itemprop="text">“The world as we have created it is a p
 <span>by <small class="author" itemprop="author">Albert Einstein</small>
 <a href="/author/Albert-Einstein">(about)</a>
 </span>
 <div class="tags">
            Tags:
            <meta class="keywords" content="change,deep-thoughts,thinking,
 <a class="tag" href="/tag/change/page/1/">change</a>
 <a class="tag" href="/tag/deep-thoughts/page/1/">deep-thoughts</a>
 <a class="tag" href="/tag/thinking/page/1/">thinking</a>
 <a class="tag" href="/tag/world/page/1/">world</a>
 </div>
 </div>,
 <div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
 <span class="text" itemprop="text">“It is our choices, Harry, that show wh
 <span>by <small class="author" itemprop="author">J.K. Rowling</small>
 <a href="/author/J-K-Rowling">(about)</a>
 </span>
 <div class="tags">
            Tags:
            <meta class="keywords" content="abilities,choices" itemprop="k
 <a class="tag" href="/tag/abilities/page/1/">abilities</a>
 <a class="tag" href="/tag/choices/page/1/">choices</a>
 </div>
 </div>,
 <div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
 <span class="text" itemprop="text">“There are only two ways to live your l
 <span>by <small class="author" itemprop="author">Albert Einstein</small>
 <a href="/author/Albert-Einstein">(about)</a>
```

```
      </span>
      <div class="tags">
                  Tags:
                  <meta class="keywords" content="inspirational,life,live,miracl
      <a class="tag" href="/tag/inspirational/page/1/">inspirational</a>
      <a class="tag" href="/tag/life/page/1/">life</a>
      <a class="tag" href="/tag/live/page/1/">live</a>
      <a class="tag" href="/tag/miracle/page/1/">miracle</a>
      <a class="tag" href="/tag/miracles/page/1/">miracles</a>
      </div>
      </div>,
      <div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
      <span class="text" itemprop="text">“The person, be it gentleman or lady, w
      <span>by <small class="author" itemprop="author">Jane Austen</small>
      <a href="/author/Jane-Austen">(about)</a>
      </span>
      <div class="tags">
                  Tags:
                  <meta class="keywords" content="aliteracy,books,classic,humor"
      <a class="tag" href="/tag/aliteracy/page/1/">aliteracy</a>
      <a class="tag" href="/tag/books/page/1/">books</a>
      <a class="tag" href="/tag/classic/page/1/">classic</a>
      <a class="tag" href="/tag/humor/page/1/">humor</a>
      </div>
      </div>,
      <div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
      <span class="text" itemprop="text">“Imperfection is beauty, madness is gen
      <span>by <small class="author" itemprop="author">Marilyn Monroe</small>
      <a href="/author/Marilyn-Monroe">(about)</a>
```

```
#Affichage des citations
citation=doc.find_all(class_='text')
```

```
citation
```

```
[<span class="text" itemprop="text">“The world as we have created it is a p
 <span class="text" itemprop="text">“It is our choices, Harry, that show wh
 <span class="text" itemprop="text">“There are only two ways to live your l
 <span class="text" itemprop="text">“The person, be it gentleman or lady, w
 <span class="text" itemprop="text">“Imperfection is beauty, madness is gen
 <span class="text" itemprop="text">“Try not to become a man of success. Ra
 <span class="text" itemprop="text">“It is better to be hated for what you
 <span class="text" itemprop="text">“I have not failed. I've just found 10,
 <span class="text" itemprop="text">“A woman is like a tea bag; you never k
 <span class="text" itemprop="text">“A day without sunshine is like, you kn
```

Afficher le nombre citations de la page web

```
print(len(articles))
```

```
10
```

Afficher les auteurs

```
#Affichage de toute les occurences des balises de class author
auteurs=doc.find_all(class_='author')


all_auteur=[auteur.text.strip() for auteur in auteurs]
all_auteur
```

```
['Albert Einstein',
 'J.K. Rowling',
 'Albert Einstein',
 'Jane Austen',
 'Marilyn Monroe',
 'Albert Einstein',
 'André Gide',
 'Thomas A. Edison',
 'Eleanor Roosevelt',
 'Steve Martin']
```

Creer une fonction pour afficher tout les tags

```
def aff_tags():
    tags=doc.find_all(class_='tag')
    all_tags=[tag.text.strip() for tag in tags]
    return all_tags
```

```
aff_tags()
```

```
['change',
 'deep-thoughts',
 'thinking',
 'world',
 'abilities',
 'choices',
 'inspirational',
 'life',
 'live',
 'miracle',
 'miracles',
 'aliteracy',
 'books',
 'classic',
 'humor',
 'be-yourself',
 'inspirational',
 'adulthood',
 'success',
 'value',
 'life',
 'love',
 'edison',
 'failure',
 'inspirational',
 'paraphrased',
 'misattributed-eleanor-roosevelt',
 'humor',
 'obvious',
 'simile',
 'love',
 'inspirational',
 'life',
 'humor',
 'books',
 'reading',
 'friendship',
 'friends',
 'truth',
 'simile']
```

Creer une fonction qui affiche une citation son auteur et ses tags

```
def Cit_aut_tag(article):

    """
    Cette fonction prend en entree une variable article
    contenant l'ensembles des donnees  pour retourner en sortie un dictionnaire
    de contenant une citation son auteur et son tag`
    """

    a_tags = article.find_all('span')
    citation = a_tags[0].text.strip()
    author = a_tags[1].text.strip()
    stars_tag = article.find_all('a',class_='tag')
    tag=[x.text.strip() for x in stars_tag ]

    return {
        'citation':  citation,
        'auteur': author,
        'tag': tag,

    }


Cit_aut_tag(articles[1])
```

```
{'citation': '“It is our choices, Harry, that show what we truly are, far m
 'auteur': 'by J.K. Rowling\n(about)',
 'tag': ['abilities', 'choices']}
```

Creer une fonction pour afficher la liste de toutes les citations auteurs et tags

```
def citation():
    top_citation=[Cit_aut_tag(article) for article in articles]
    return  top_citation
```

```
citation()
```

```
[{'citation': '“The world as we have created it is a process of our thinkin
  'auteur': 'by Albert Einstein\n(about)',
  'tag': ['change', 'deep-thoughts', 'thinking', 'world']},
 {'citation': '“It is our choices, Harry, that show what we truly are, far
  'auteur': 'by J.K. Rowling\n(about)',
  'tag': ['abilities', 'choices']},
 {'citation': '“There are only two ways to live your life. One is as though
  'auteur': 'by Albert Einstein\n(about)',
  'tag': ['inspirational', 'life', 'live', 'miracle', 'miracles']},
 {'citation': '“The person, be it gentleman or lady, who has not pleasure i
  'auteur': 'by Jane Austen\n(about)',
  'tag': ['aliteracy', 'books', 'classic', 'humor']},
 {'citation': "“Imperfection is beauty, madness is genius and it's better t
  'auteur': 'by Marilyn Monroe\n(about)',
  'tag': ['be-yourself', 'inspirational']},
 {'citation': '“Try not to become a man of success. Rather become a man of
  'auteur': 'by Albert Einstein\n(about)',
  'tag': ['adulthood', 'success', 'value']},
 {'citation': '“It is better to be hated for what you are than to be loved
  'auteur': 'by André Gide\n(about)',
  'tag': ['life', 'love']},
 {'citation': "“I have not failed. I've just found 10,000 ways that won't w
  'auteur': 'by Thomas A. Edison\n(about)',
  'tag': ['edison', 'failure', 'inspirational', 'paraphrased']},
 {'citation': "“A woman is like a tea bag; you never know how strong it is
  'auteur': 'by Eleanor Roosevelt\n(about)',
  'tag': ['misattributed-eleanor-roosevelt']},
 {'citation': '“A day without sunshine is like, you know, night.”',
  'auteur': 'by Steve Martin\n(about)',
  'tag': ['humor', 'obvious', 'simile']}]
```

creer une fonction qui permet de d'enregistrer dans un fichier csv l'ensemble des citations leurs auteurs et tags de la page web

```
def write_csv(items, path):

    """
    Cette fonction prend en entree un dictionnaire
    contenant les citations,auteurs et tags
    dans une variable appelee items puis
    l'enregistre dans un fichier csv avec la varible path
    """
    # Open the file in write mode
    with open(path, 'w') as f:
        # Return if there's nothing to write
        if len(items) == 0:
            return

        # Write the headers in the first line
        headers = list(items[0].keys())
        f.write(','.join(headers) + '\n')

        # Write one item per line
        for item in items:
            values = []
            for header in headers:
                values.append(str(item.get(header, "")))
            f.write(','.join(values) + "\n")


write_csv(top_citation, 'citation_gratos.csv')
```

Creer une fonction pour afficher les citations d'un auteur donné

```
def cit_auth(author):
    """
    cette fonction prend en entree le nom de l'auteur
    et vous affiche en sortie les citation qu'il a produit
    dans la page

    """
    cita=[]
    for citation in top_citation:
        if (citation['auteur']==author):
            cita.append(citation)

    return cita
```

```
cit_auth('by Albert Einstein\n(about)')
```

```
[{'citation': '“The world as we have created it is a process of our thinkin
  'auteur': 'by Albert Einstein\n(about)',
  'tag': ['change', 'deep-thoughts', 'thinking', 'world']},
 {'citation': '“There are only two ways to live your life. One is as though
  'auteur': 'by Albert Einstein\n(about)',
  'tag': ['inspirational', 'life', 'live', 'miracle', 'miracles']},
 {'citation': '“Try not to become a man of success. Rather become a man of
  'auteur': 'by Albert Einstein\n(about)',
  'tag': ['adulthood', 'success', 'value']}]
```