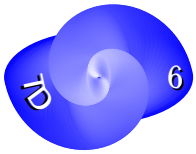

Importations d'autres contenus dans NLTK

NLTK

Exercice 1

Utilisations intéressantes de NLTK

1. http://nltk.org/book3/. C'est le site du livre *Natural Language Processing with Python* aux éditions O'reilly. Bien prendre la version Python 3 si c'est la version de Python dont vous disposez.
2. http://streamhacker.wordpress.com/tag/nltk/ site présentant des exemples intéressants de Web mining.
3. https://github.com/ptwobrussell/Mining-the-Social-Web-2nd-Edition

Code Python 1 – Texte brut importé dans NLTK

```
# Souvenirs d'un entomologiste de JH Fabre
from urllib import request
url='http://www.gutenberg.org/cache/epub/16825/pg16825.txt'
response = request.urlopen(url)
texte_brut = response.read().decode('utf8')
tokens = nltk.word_tokenize(texte_brut)
text = nltk.Text(tokens)
text.collocations()
# etc.
```

Exercice 2

Récupération de textes externes dans NLTK

— Suivez le code ci-dessus pour récupération d'un livre sur le Web
— On le *tokenize* pour l'importer dans NLTK

Exercice 3

Etude de contenus Web

Pour examiner des contenus Web :
— On récupère la ou les pages à étudier
— La nouvelle version de NLTK ne nettoie plus le html et charge BeautifulSoup de cette tâche.
— Il faut un peu bricoler en plus pour nettoyer les résultats. Les pages Web d'aujourd'hui peuvent être très complexes comme celles des grands journaux ou portails Web.
— Etudier le code ci-dessous

Code Python 2 – Contenu Web

```
from urllib import request
url='http://www.lemonde.fr/le-magazine/article/2014/01/16/le-dessein-anime-de-miyazaki_4348993_1616923.html'
html = request.urlopen(url).read().decode('utf8')
html[:200]
```

Code Python 3 – Nettoyage html avec BeautifulSoup

```
import nltk
from bs4 import BeautifulSoup
brut = BeautifulSoup(html).get_text()
tokens = nltk.word_tokenize(brut)
miyazaki = nltk.Text(tokens)
```

Code Python 4 – Nettoyage html amélioré

```
# on peut faire mieux:
import nltk
from bs4 import BeautifulSoup
import re
soup = BeautifulSoup(html)
texts = soup.findAll(text=True)

def vis_text(element):
    if element.parent.name in ['style', 'script', '[document]', 'head', 'title']:
        return ''
    result = re.sub('<!--.*-->|\r|\n', '', str(element), flags=re.DOTALL)
    result = re.sub('\s{2,}| ', ' ', result)
    return result

visible_elements = [vis_text(elem) for elem in texts]
texte_visible = ''.join(visible_elements)
tokens = nltk.word_tokenize(texte_visible)
miyazaki = nltk.Text(tokens)
```