**Завдання 1.** Напишіть функцію, яка приймає адресу URL, як аргумент, і повертає те що міститься за цією адресою з видаленням HTML розмітки. Використовувати urllib.urlopen для доступу до контенту наступним чином raw\_contents = urllib.urlopen('http://www.nltk.org/').read().

*Програма:*

import urllib

from urllib import urlopen

def cleanhtml (url):

raw = urlopen(url).read()

raw\_contents = nltk.clean\_html(raw)

tokens = nltk.word\_tokenize(raw\_contents)

return tokens

url = "http://www.gutenberg.org/files/1342/1123.txt"

s = cleanhtml(url)

print (s[10:70])

*Результат:*

['denied', 'he', 'knew', 'AWB', 'was', 'paying', 'kickbacks', 'to', 'Iraq', 'despite', 'writing', 'to', 'the', 'wheat', 'exporter', 'asking', 'to', 'be', 'kept', 'fully', 'informed', 'on', 'Iraq', 'wheat', 'sales', '.', 'Letters', 'from', 'John', 'Howard', 'and', 'Deputy', 'Prime', 'Minister', 'Mark', 'Vaile', 'to', 'AWB', 'have', 'been', 'released', 'by', 'the', 'Cole', 'inquiry', 'into', 'the', 'oil', 'for', 'food', 'program', '.', 'In', 'one', 'of', 'the', 'letters', 'Mr', 'Howard', 'asks']

**Завдання 2.** Збережіть деякий текст у файлі corpus.txt. Визначити функцію load(f) для читання файлу, назва якого є її аргументом і повертає стрічку, яка містить текст з файлу.

*Програма:*

from \_\_future\_\_ import division

import nltk, re, pprint

def load(f):

raw=open(f)

return(raw.read())

print(load('austen-emma.txt'))

**Завдання 3.** Перепишіть наступний цикл як list comprehension:

|  |  |
| --- | --- |
|  | **>>> sent = ['The', 'dog', 'gave', 'John', 'the', 'newspaper']**  **>>> result = []**  **>>> for word in sent:**  **... word\_len = (word, len(word))**  **... result.append(word\_len)**  **>>> result**  **[('The', 3), ('dog', 3), ('gave', 4), ('John', 4), ('the', 3), ('newspaper', 9)]** |

*Програма:*

sent = ['The', 'dog', 'gave', 'John', 'the', 'newspaper']

result = [(word, len(word)) for word in sent]

print(result)

**Завдання 4.** Перевірити різницю між стрічками і цілим виконавши наступні дії: "3" \* 7 та 3 \* 7. Спробуйте здійснити конвертування між стрічками і цілими використавши int("3") та str(3).

*Програма:*

print('3'\*7)

print(3\*7)

print(int('3')\*7)

print(str(3)\*7)

*Результат:*

3333333

21

21

3333333

**Завдання 5.** Що станеться, коли стрічки форматування %6s та %-6s використовується для відображення стрічки довшої ніж 6 символів?

Ширина стрічки в результаті залишиться рівною ширині вхідної стрічки, якщо вона містить більше ніж 6 символів.

**Завдання 6.** Створіть файл, який буде містити слова та їх частоту записані в окремих рядках через пробіл ( fuzzy 53). Прочитайте цей файл використовуючи open(filename).readlines(). Розділіть кожну стрічку на дві частини використовуючи split(), і перетворіть число в ціле значення використовуючи int(). Результат повинен бути у вигляді списку: [['fuzzy', 53], ...].

*Програма:*

import nltk

from nltk.corpus import wordnet as wn

from nltk.corpus import brown

from nltk.probability import FreqDist

text=open('text.txt', 'r').readlines()

list=[]

pair2=0

for pair in text:

pair=pair.split()

pair[1]=int(pair[1])

list.append(pair)

print (list)

*Результат:*

[['text', 45], ['girl', 12], ['boy', 34], ['simple', 1], ['clue', 3], ['spoon', 2], ['man', 32]]

**Завдання 13.** Використовуючи Porter стемер нормалізуйте будь-який токенізований текст . До того самого тексту застосуйте Lancaster стемер. Результати порівняйте та поясніть*.*

*Програма:*

import nltk

from nltk.corpus import gutenberg

from nltk.probability import FreqDist

raw = open('austen-emma.txt').read()

tokens = nltk.word\_tokenize(raw)

porter = nltk.PorterStemmer()

lancaster = nltk.LancasterStemmer()

print ([porter.stem(t) for t in tokens][:50], '\n')

print ([lancaster.stem(t) for t in tokens][:50])

*Результат:*

['[', 'Emma', 'by', 'Jane', 'Austen', '1816', ']', 'VOLUM', 'I', 'CHAPTER', 'I', 'Emma', 'Woodhous', ',', 'handsom', ',', 'clever', ',', 'and', 'rich', ',', 'with', 'a', 'comfort', 'home', 'and', 'happi', 'disposit', ',', 'seem', 'to', 'unit', 'some', 'of', 'the', 'best', 'bless', 'of', 'exist', ';', 'and', 'had', 'live', 'nearli', 'twenty-on', 'year', 'in', 'the', 'world', 'with']

['[', 'emm', 'by', 'jan', 'aust', '1816', ']', 'volum', 'i', 'chapt', 'i', 'emm', 'woodh', ',', 'handsom', ',', 'clev', ',', 'and', 'rich', ',', 'with', 'a', 'comfort', 'hom', 'and', 'happy', 'disposit', ',', 'seem', 'to', 'unit', 'som', 'of', 'the', 'best', 'bless', 'of', 'ex', ';', 'and', 'had', 'liv', 'near', 'twenty-one', 'year', 'in', 'the', 'world', 'with']

**Завдання 12.** Міра оцінки читабельності використовується для оцінки складності тексту для читання. Нехай, μw - середня кількість літер у слові, та μs – середнє значення кількості слів у реченні в певному тексті. Automated Readability Index (ARI) тексту визначається згідно виразу: 4.71 μw + 0.5 μs - 21.43. Визначити значення ARI для різних частин корпуса Brown Corpus, включаючи частину f (popular lore) та j (learned). Використовуйте nltk.corpus.brown.words() для знаходження послідовності слів та nltk.corpus.brown.sents() для знаходження послідовності речень.

*Програма:*

import nltk

from nltk.corpus import brown

from nltk.probability import FreqDist

mw=0 #av.number of letters in a word

ms=0 #av. number of words in a sent.

n=m=0

part1words=nltk.corpus.brown.words(categories='lore')

part2words=nltk.corpus.brown.words(categories='learned')

part1sents=nltk.corpus.brown.sents(categories='lore')

part2sents=nltk.corpus.brown.sents(categories='learned')

for word in part1words:

n=n+len(word)

for sent in part1sents:

m=m+len(sent)

mw=n/len(part1words)

ms=m/len(part1sents)

print ('ARI for genre LORE is:', '%g'%(4.71\*mw+0.5\*ms-21.43))

n=m=0

for word in part2words:

n=n+len(word)

for sent in part2sents:

m=m+len(sent)

mw=n/len(part2words)

ms=m/len(part2sents)

print ('ARI for genre LEARNED is:', '%g'%(4.71\*mw+0.5\*ms-21.43))

*Результат:*

ARI for genre LORE is: 10.2548

ARI for genre LEARNED is: 11.926

**Завдання 12.** Доступіться до текстів ABC Rural News та ABC Science News з корпуса (nltk.corpus.abc). Знайдіть значення для оцінки читабельності текстів (аналогічно до задачі №12). Використовуйте Punkt для поділу тексту на окремі речення.

*Програма:*

import nltk

from nltk.corpus import brown

from nltk.probability import FreqDist

mw=0 #av.number of letters in a word

ms=0 #av. number of words in a sent.

n=0

sent\_tokenizer=nltk.data.load('tokenizers/punkt/english.pickle')

part1=nltk.corpus.abc.raw('rural.txt')

part2=nltk.corpus.abc.raw('science.txt')

part1sents=sent\_tokenizer.tokenize(part1)

part2sents=sent\_tokenizer.tokenize(part2)

part1words=nltk.corpus.abc.words('rural.txt')

part2words=nltk.corpus.abc.words('science.txt')

for word in part1words:

n=n+len(word)

mw=n/len(part1words)

for sent in part1sents:

sent=sent.split()

ms=ms+len(sent)

ms=ms/len(part1sents)

print ('ARI for text RURAL is:', '%g'%(4.71\*mw+0.5\*ms-21.43))

n=0

for word in part2words:

n=n+len(word)

mw=n/len(part2words)

for sent in part2sents:

sent=sent.split()

ms=ms+len(sent)

ms=ms/len(part2sents)

print ('ARI for text SCIENCE is:', '%g'%(4.71\*mw+0.5\*ms-21.43))

*Результат:*

ARI for text RURAL is: 10.6607

ARI for text SCIENCE is: 10.7047