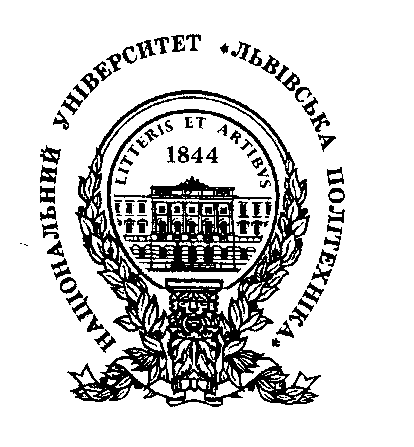
Міністерство освіти та науки України

Національний університет «Львівська Політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра САПР



Лабораторна робота №6

на тему:

ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ. ВИКОРИСТАННЯ РЕГУЛЯРНИХ ВИРАЗІВ ДЛЯ ОБРОБКИ ТЕКСТУ.

Виконала:

студентка ПРЛм-11

Михайлів Р.Б.

Перевірив:

Дупак Б.П.

Львів 2015

**МЕТА РОБОТИ**

* Вивчення основ програмування на мові *Python*.
* Використання регулярних виразів для обробки текстів.

**КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ**

## Використання регулярних виразів для виявлення слів за заданими шаблонами.

Багато задач лінгвістичних досліджень передбачають встановлення відповідності заданому шаблону. Наприклад, можна знайти слова, які закінчуються на “ed” використовуючи метод endswith('ed'). Подібні методи перевірки слів перелічені в Таблиці Методичних вказівок до лабораторної роботи №2. Регулярні вирази є більш потужним і гнучким методом опису шаблонів символів, які необхідно виявити у послідовностях символів. **Регулярний вираз (вислів)** (в [програмуванні](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) — це [рядок](http://uk.wikipedia.org/wiki/%D0%A0%D1%8F%D0%B4%D0%BE%D0%BA_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%29) що описує або збігається з [множиною](http://uk.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B8%D0%BD%D0%B0) рядків, відповідно до набору спеціальних [синтаксичних](http://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81) правил. Регулярні вислови використовуються в багатьох [текстових редакторах](http://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D0%B8%D0%B9_%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80) та допоміжних інструментах для пошуку та зміни тексту на основі заданих шаблонів.

Для роботи з регулярними виразами у Python потрібно імпортувати бібліотеку re скориставшись: import re. Корпус слів англійської мови Words Corpus буде використовуватися в якості лінгвістичних даних серед яких буде проводитися пошук. Попередня підготовка списку слів передбачає видалення власних імен.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> import re**  **>>> wordlist = [w for w in nltk.corpus.words.words('en') if w.islower()]** | |

### Використання основних метасимволів (операторів повтору).

Синтаксис регулярних висловів залежить від інтерпретатора, що використовується для їх обробки. Однак, із незначними відхиленнями, майже всі поширені механізми інтерпретатори регулярних висловів мають спільні правила.

Найпростіший регулярний вислів, з якого формуються складні, є звичайний символ. Більшість символів, включаючи усі літери та цифри, є регулярними висловами, що співпадають із відповідними символами в рядках.

Пошук слів із закінченням ed можна здійснити використовуючи регулярний вираз «ed$». Потрібно використати функцію re.search(p, s), яка перевіряє чи може зразок p бути знайдений у будь-якому місці стрічки s. Потрібно визначити символи, які шукаємо та використати символ долара , який в регулярних виразах позначає кінець слова:

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> [w for w in wordlist if re.search('ed$', w)]**  **['abaissed', 'abandoned', 'abased', 'abashed', 'abatised', 'abed', 'aborted', ...]** | |

Символ ”.” універсальний символ , якому відповідає будь-який один символ. Нехай потрібно знайти слова з восьми літер, де j – третя літера та t – шоста літера. При створенні регулярного виразу у місцях де може бути будь-який символ вказується крапка. Символ ”^” вказує на початок стрічки:

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> [w for w in wordlist if re.search('^..j..t..$', w)]**  **['abjectly', 'adjuster', 'dejected', 'dejectly', 'injector', 'majestic', ...]** | |

В мобільних телефонах для вводу тексту використовується система T9 (Рис.1.). Два або більше слів, які можуть бути введені тією самою послідовністю клавіш називають textonyms. Наприклад два слова hole та golf вводяться натисненням послідовності 4653. Які інші слова можуть бути створені натисненням тієї самої послідовності клавіш? Використовуючи регулярний вираз «^[ghi][mno][jlk][def]$» це не складно зробити:

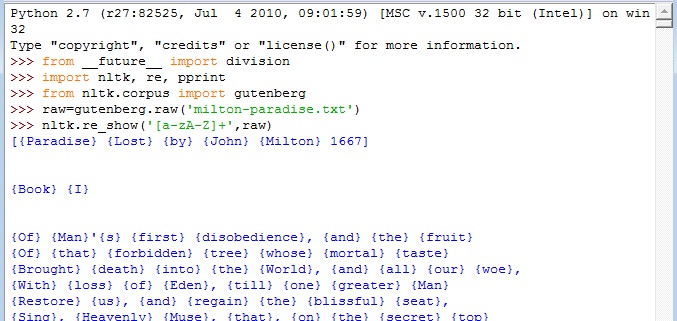
|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> [w for w in wordlist if re.search('^[ghi][mno][jlk][def]$', w)]**  **['gold', 'golf', 'hold', 'hole']** | |

Перша частина виразу, «^[ghi]», вказує на початок слова після якого йде один з символів g, h, чи i. Наступна частина виразу, «[mno]», містить другий символ m, n, або o. Обмеження на вибір третього та четвертого символів встановлюються аналогічно. Тільки чотири слова відповідають заданому шаблону. Порядок символів у квадратних дужках є довільним і не впливає на результат. could have written «^[hig][nom][ljk][fed]$» and matched the same words.

**ТЕКСТИ ПРОГРАМ НА МОВІ PYTHON**

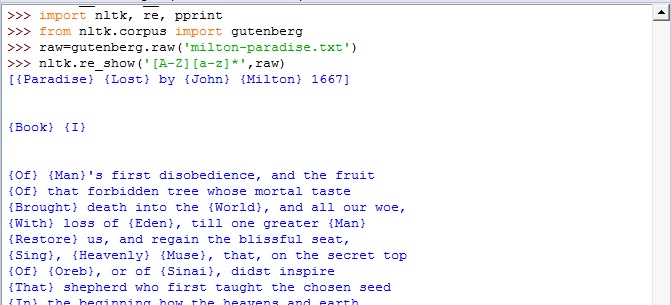
**Варіант 20**

3.1. Описати, які класи стрічок відповідають наступному регулярному виразу. [a-zA-Z]+. Результати перевірити використовуючи nltk.re\_show()



Знаходить всі слова з великої або малої літери, без цифр та розділових знаків.

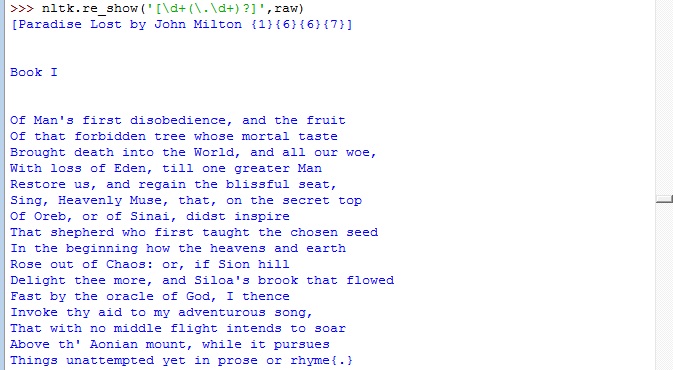
3.2. Описати, які класи стрічок відповідають наступному регулярному виразу. [A-Z][a-z]\*. Результати перевірити використовуючи nltk.re\_show()



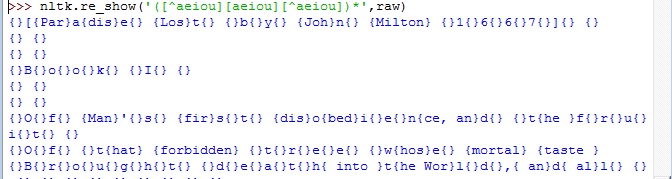
Знаходить всі слова з великої літери.

3.3. Описати, які класи стрічок відповідають наступному регулярному виразу. \d+(\.\d+)?. Результати перевірити використовуючи nltk.re\_show()

|  |
| --- |
|  |

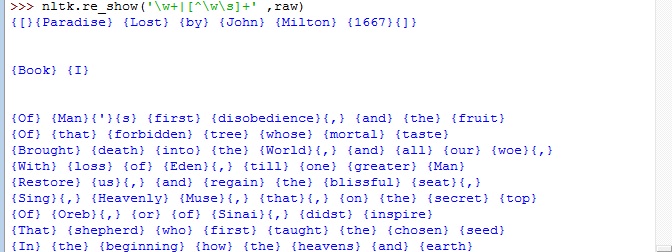


3.4. Описати, які класи стрічок відповідають наступному регулярному виразу. ([^aeiou][aeiou][^aeiou])\*. Результати перевірити використовуючи nltk.re\_show()



Послідовність букв – не голосна – голосна – не голосна.

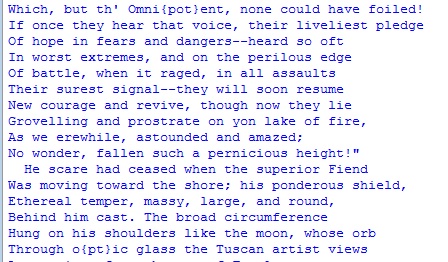
3.5. Описати, які класи стрічок відповідають наступному регулярному виразу. \w+|[^\w\s]+.. Результати перевірити використовуючи nltk.re\_show()



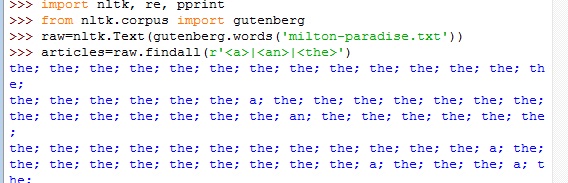
Всі слова, символи та цифри.

3.6. Описати, які класи стрічок відповідають наступному регулярному виразу. p[aeiou]{,2}t Результати перевірити використовуючи nltk.re\_show()

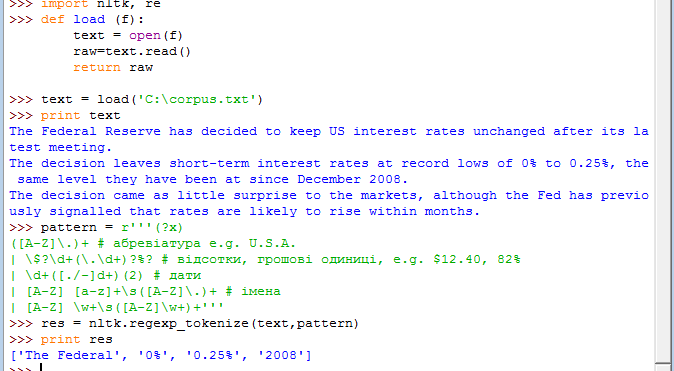
D:\Shit\5 курс\Компютерна лінгв\python labs\6.6.jpg



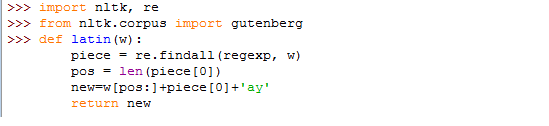
* 1. Написати регулярний вираз, який встановлює відповідність наступному класу стрічок: всі артиклі *(a, an, the*).

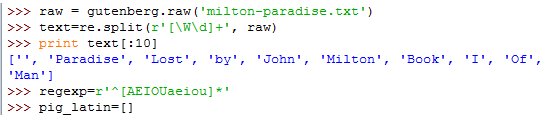


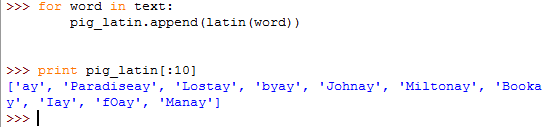
3.10. Зберегти довільний текст у файлі corpus.txt. Визначити функцію для читання з цього файлу (назва файлу аргумент функції) і повертає стрічку, яка містить текст з файлу. Використовуючи nltk.regexp\_tokenize() розробити токенізатор для токенізації різних типів виразів: грошові одиниці, дати, імена людей та організацій. Використовувати багаторядковий запис регулярного виразу з коментарями та «verbose flag».



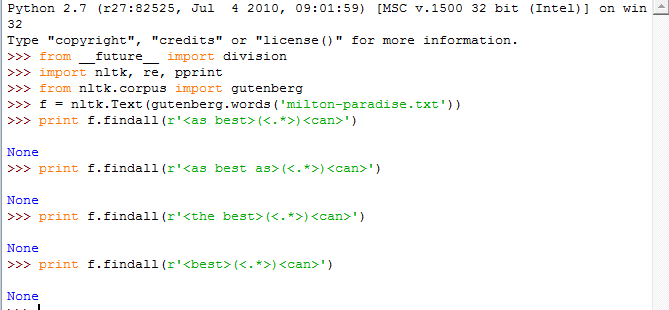
* 1. Напишіть програму, яка конвертує текст в *Pig Latin. String->ingstray, idle->idleay*. (Конвертація відбувається переміщенням приголосної або групи приголосних на початок слова та додаванням до слова ay ).







* 1. Прочитати Додаток А. Дослідити явища описані у Додатку А використовуючи корпуси текстів та метод findall()для пошуку в токенізованому тексті.



**ВИСНОВОК**

На цій лабораторній роботі, ми вивчили основи програмування на мові Python та використання регулярних виразів для обробки текстів.