**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**іНСТИТУТ КОМП’ютерних НАУК та ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

### Кафедра “Системи автоматизованого проектування”



Звіт

до лабораторної роботи №2

на тему: “ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.

ОСНОВИ ПРОГРАМУВАННЯ НА МОВІ PYTHON(частина 2) ”

з дисципліни “Комп’ютерна лінгвістика”

Виконала:

студентка групи ПРЛм-11

Павлів І.О.

Прийняв:

викладач

Дупак Б.П.

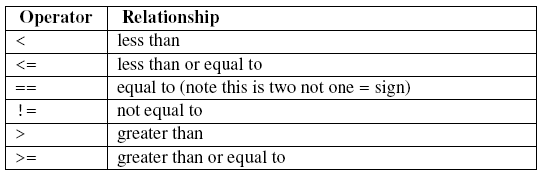
Львів-2015

**Мета роботи:** вивчити основи програмування на мові *Python*,ознайомитись з контрольними структурами та класом FreqDist.

**Теоретичні відомості.**

Python підтримує широкий набір операторів для встановлення взаємозв’язків між змінними (значеннями). Повний набір цих операторів наведений у таблиці 1.

Таблиця 1.



Загальна схема роботи цих прикладів ([w for w in text if *condition* ]), де *condition* умова, яка справджується або ні (приймає значення True або False ).

Звичайно ми використовуємо умовні оператори, як частину *If* операторів. Для перевірки властивостей окремих слів існує набір наступних функцій (Таблиця2.).

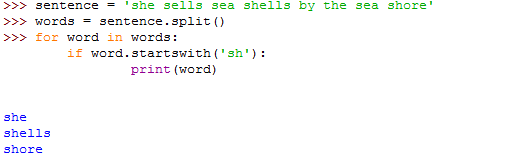
| **Функція** | **Пояснення** |
| --- | --- |
| s.startswith(t) | чи починається s з t |
| s.endswith(t) | чи закінчується s на t |
| t in s | Чи t міститься в s |
| s.islower() | Чи всі символи в s є малі |
| s.isupper() | Чи всі символи в s є великі |
| s.isalpha() | Чи всі символи в s є букви |
| s.isalnum() | Чи всі символи в s є букви і цифри |
| s.isdigit() | Чи всі символи в s є цифри |
| s.istitle() | Чи всі слова в s є з великої літери |

В даних прикладах наступні вирази: [f(w) for ...] or [w.f() for ...], де f це функція, яка або визначає довжину слова або перетворює малі літери на великі. В кожному з цих прикладів здійснюється обробка кожного елемента списку. Змінній W послідовно присвоююся значення слів з текста і над цією змінною виконуються передбачені програмою дії. Такий запис [f(w) for ...] називається "list comprehension." (включення списків або спискові висловлювання) і є важливим для написання та розуміння програм на Python.

Для автоматичного визначення слів, які є найбільш інформативними для текстів певного жанру або певної тематики спочатку інтуітивно виникає думка побудувати частотний список або частотний розподіл. Частотний розподіл вказує на частоту з якою в тексті зустрічається кожне зі слів. Такий частотний список називають розподілом тому, що він вказує яким чином загальна кількість слів розподіляється між словниковими статями (оригінальні слова) в тексті. Враховуючи що побудова частотних розподілів часто необхідна при обробці природної мови в NLTK реалізовано окремий клас FreqDist в модулі nltk.probability . Застосуємо цей клас для знаходження 50 найчастотніших слів в тексті *Moby Dick*.

**Варіант №13**

* 1. Створіть змінну *sentence* і присвойте їй значення *‘she sells sea shells by the sea shore’* та напишіть фрагмент програми для виведення на екран всіх слів які починаються з *‘sh’*.



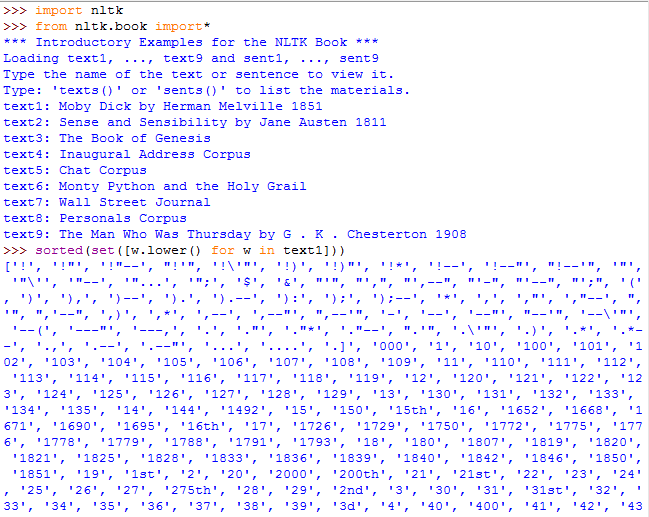
* 1. Напишіть програму, яка видаляє всі голосні зі стрічки, яка відповідає імені, по батькові та прізвищу студента. Програма повинна здійснювати наступну послідовність дій: створення початкової стрічки; створення стрічки, у якій буде зберігатися результат; *for* цикл для обробки стрічки символ за символом і запису неголосних символів в результуючу стрічку.

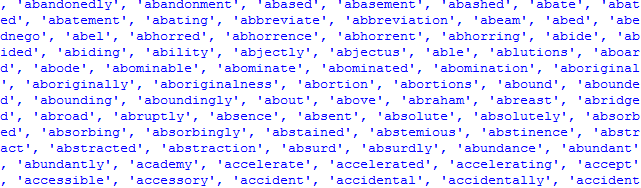


* 1. Виконати наступні приклади і пояснити чому отримані різні результати (різні значення змінних)

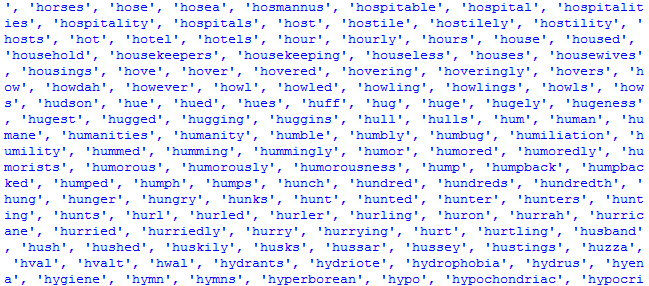
sorted(set([w.lower() for w in text1]))

sorted([w.lower() for w in set(text1)])



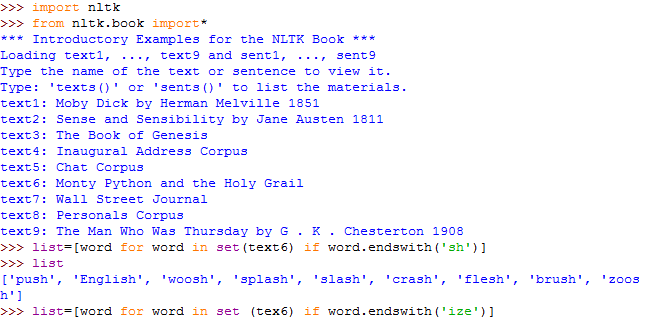




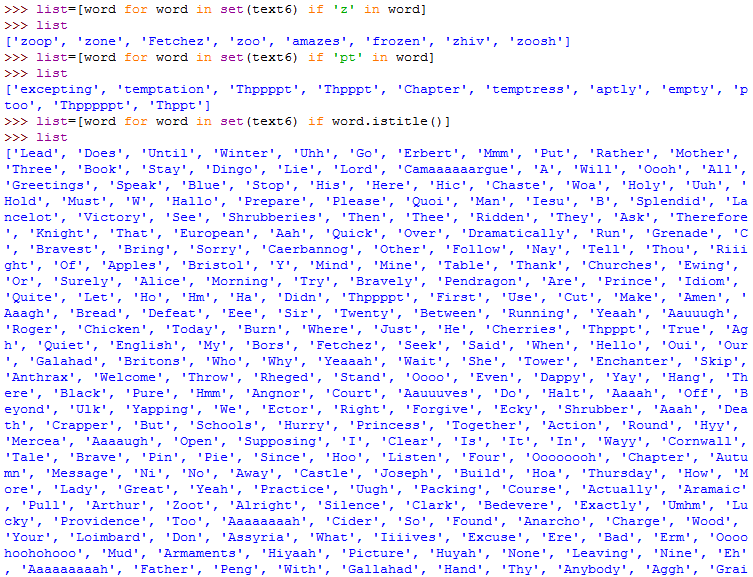


Так як в першому випадку set вжито перед умовою, тому і рахувало без повторів.

1.11. Напишіть вираз для знаходження в тексті №6 всіх слів які відповідають наступним вимогам: закінчуються на ize; містять літеру z; містять послідовність літер pt; написані з великої літери . Результат представити, як список слів.

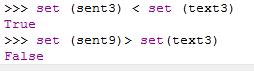






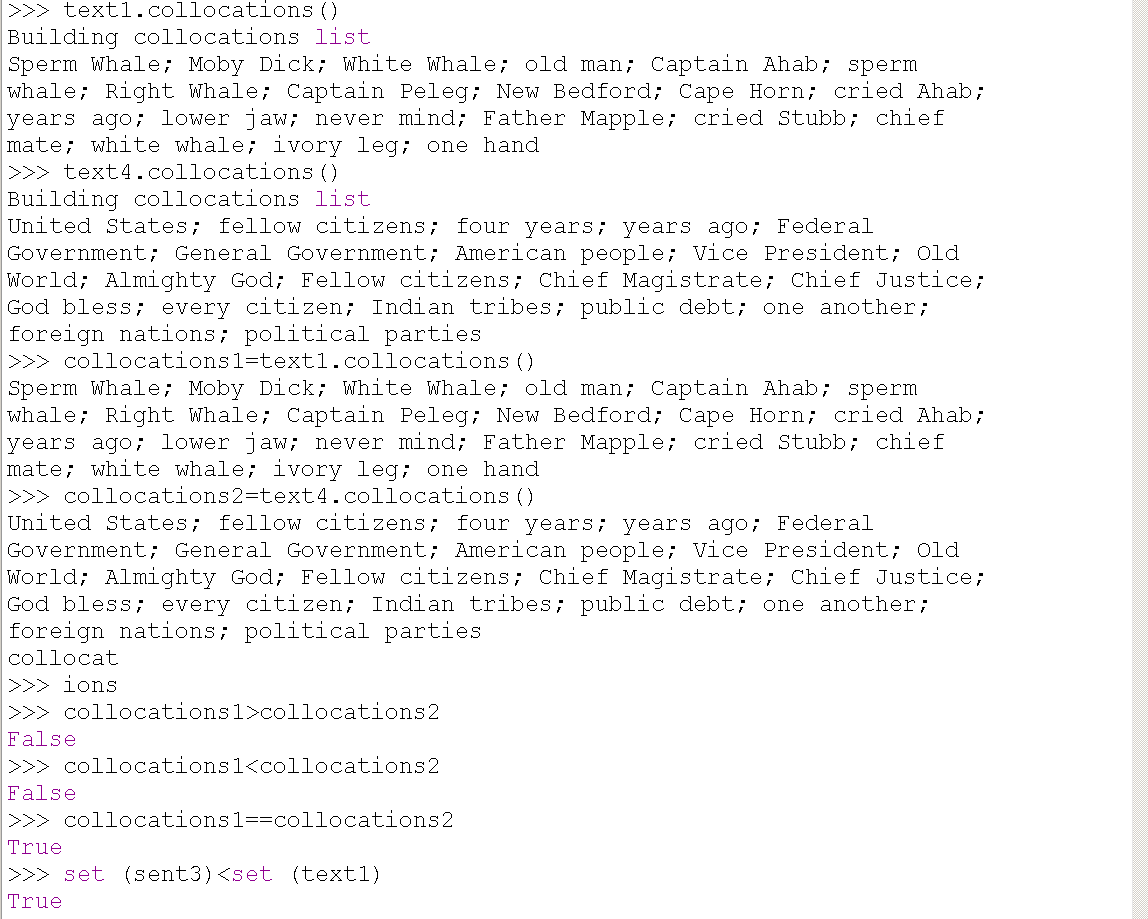
* 1. Перевірте виконання виразу set(sent3) < set(text1). Змініть аргументи функції. Результати поясніть.





В першому випадку задали умову, що третє речення є меншим, ніж перший текст. Це виявилось вірним твердженням. В другому випадку я змінила умову: третє речення є меншим, ніж третій текст. Теж вірно. В третьому випадку: дев’яте речення більше, ніж третій текст. Є хибним твердженням.

* 1. Побудуйте колокації для текстів №1 та №4. Результати порівняйте.



**Висновок:** у цій лабораторній роботі я ознайомилась з основами програмування на мові Python, та з контрольними структурами та класом FreqDist.