

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра “Системи автоматизованого проектування”



Звіт

до лабораторної роботи №1

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

ОСНОВИ ПРОГРАМУВАННЯ НА МОВІ PYTHON(частина 1) ”

з дисципліни “Комп’ютерна лінгвістика”

Виконала:

студентка групи ПРЛм-11

Липак О.В.

Прийняв:

викладач

Дупак Б.П.

Львів-2015

Мета роботи: вивчити основи програмування на мові *Python*.

Теоретичні відомості.

Python - це проста і потужна об'єктно-орієнтована мова програмування високого рівня з чудовими можливостями для обробки лінгвістичних даних.

Natural Language Toolk (NLTK) – набір *Python* бібліотек, які призначені для аналізу текстів природною мовою. *NLTK* дозволяє здійснювати символічний та статистичний аналіз текстів, створювати графічні звіти та містить детальну документацію і використовується в проєктах з лінгвістики, штучного інтелекту, машинного навчання, автоматизації документообігу. Його можна застосовувати як початковий комплекс, готовий аналітичний інструмент або платформу для створення прикладних систем опрацювання текстів. *NLTK* вільно розповсюджується (<http://www.nltk.org>) і всі бажаючі можуть його встановити згідно інструкції розробників.

Python – інтерпретаційна мова, яка дозволяє зекономити час, що витрачається на компіляцію. Інтерпретатор можна використовувати інтерактивно, що дозволяє експериментувати з можливостями мови і створювати фрагменти програм або тестувати окремі функції. Інтерпретатор – це програма яка виконує *Python* програми.

```
Python 2.4.3 (#1, Mar 30 2006, 11:02:16)
[GCC 4.0.1 (Apple Computer, Inc. build 5250)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

При запуску інтерпретатора ми бачимо інформацію про його версію, додаткову інформацію і запрошення >>> вводити оператори *Python*. У випадку використання *Interactive DeveLopment Environment (IDLE)* нам доступні додаткові зручності, зокрема у відображенні тексту програми на екрані.

Текст або частини тексту в програмах на *Python* представляються за допомогою *стрічок (string)* і повинен відділятися від решти програми лапками (одинарними, подвійними або потрійними).

Якщо стрічка містить одинарні лапки необхідно використовувати лівий слеш перед апострофом для того, щоб символ апострофа не розглядався як

символ завершення стрічки або використовувати подвійні лапки. Якщо цього не зробити то отримаємо помилку.

Деколи стрічки можуть складатися з декількох рядків. Python забезпечує декілька способів роботи з ними. В наступному прикладі послідовність з двох стрічок об'єднується в одну. Потрібно використовувати лівий слеш або круглі дужки для того щоб інтерпретатор знав що ввід стрічки ще не завершився після введення першого рядка.

Lists(списки) – тип даних для опису послідовності значень. В *Python* списки представляються як послідовність записана через кому і у квадратних дужках.

Tuples(кортежі) – подібний до списків тип даних але подібно до стрічок кортежі також не можемо змінювати.

Тексти програм на мові *Python*.

Варіант №7

3. Здійснити арифметичні операції зі стрічкою *msg*.

```
>>> msg = 'Olena'
>>> num = 5
>>> msg * num
'OlenaOlenaOlenaOlenaOlena'
>>> msg2 = 'Lypak'
>>> msg = msg + msg2
>>> msg
'OlenaLypak'
```

5. Використовуючи зрізи та операцію поєднання змінити стрічку *msg* до вигляду ім'я , по батькові, прізвище студента.

```
>>> msg = 'OlenaLypak'
>>> msg = msg [0:5] + ' ' + ' Volodymyrivna' + ' ' + msg[5:10]
>>> msg
'Olena Volodymyrivna Lypak'
```

6. Визначити стрічку *s*='colorless'. Використовуючи зрізи та операцію поєднання змінити стрічку до вигляду 'colourless'.

```
>>> s = 'colorless'
>>> s = s[0:4] + 'u' + s [4:9]
>>> s
'colourless'
```

12. Представити прізвище, ім'я та по батькові як список стрічок. Розділити речення на окремі елементи, межа розділу голосна буква.

```
>>> list1 = ['Lypak', 'Olena', 'Volodymyrivna']
>>> string1 = ''.join(list1)
>>> string1
'LypakOlenaVolodymyrivna'
>>> string2 = string1.split('a')
>>> string2
['Lyp', 'kOlen', 'Volodymyrivn', '']
```

17. Створіть файл *test.py*, який містить стрічку *msg* . Використайте наступні оператори і поясніть отримані результати.

```
>>> from test import msg
>>> msg
```

```
test.py - C:/Python27/test.py
File Edit Format Run Options Windows Help
msg = 'Olena Volodymyrivna Lypak'

>>> from test import msg
>>> msg
'Olena Volodymyrivna Lypak'
```

Було створено файл *test.py*, який містить стрічку `msg = 'Olena Volodymyrivna Lypak'`. В IDLE було введено дані оператори, результатом є виведення вмісту змінної *msg* на екран.

20. Визначіть змінну *silly*, яка буде містити стрічку 'newly formed bland ideas are inexpressible in an infuriating way' і напишіть програму перетворення стрічки на список стрічок (кожне слово окрема стрічка). Результати збережіть в змінній *bland*.

```
>>> silly = 'newly formed bland ideas are inexpressible in an infuriating way'
>>> bland = silly.split()
>>> bland
['newly', 'formed', 'bland', 'ideas', 'are', 'inexpressible', 'in', 'an', 'infuriating', 'way']
```

25. Визначіть позиції всіх слів в списку *phrase1* використовуючи метод *index()*.

```
>>> phrase1 = ['Lypak', 'Olena', 'Volodymyrivna']
>>> for word in phrase1:
    print "Word = %s\nIndex = %s\n*****" % (word, phrase1.index(word))

Word = Lypak
Index = 0
*****
Word = Olena
Index = 1
*****
Word = Volodymyrivna
Index = 2
*****
```

Висновок: у цій лабораторній роботі я вивчила основи програмування на мові *Python*.