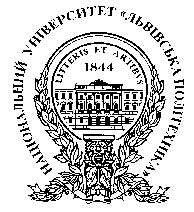
МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

Національний Університет “Львівська політехніка”



Лабораторна робота № 1

***ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.***

***ОСНОВИ ПРОГРАМУВАННЯ НА МОВІ PYTHON(частина 1).***

Виконав:

студент групи ПРЛс-11

Форманюк А.А.

Прийняв:

асистент

Дупак П.Е.

Львів 2015

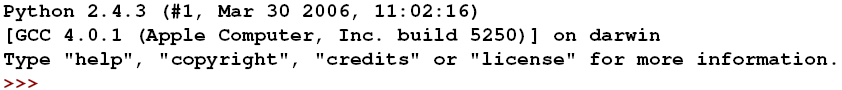
**Мета:** вивчити основи програмування на мові Python.

**Короткі 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 символьний та статистичний аналіз текстів, створювати графічні звіти та містить детальну документацію і використовується в проектах з лінгвістики, штучного інтелекту, машинного навчання, автоматизації документообігу. Його можна застосовувати як начальний 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.



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.

**Задачі**

* 1. 1. Визначити нову стрічку *hello*. Здійснити операцію *hello+ msg.* Змінити стрічку *hello* додавши в її кінці символ пробілу і знову виконати операцію *hello+ msg.*
  2. *>>> msg = 'hello'*
  3. *>>> str = ' '*
  4. *>>> msg+str*
  5. *'hello '*
  6. 2. Поясніть результат виконання *msg[::-1].*
  7. *msg = 'hello'*
  8. *>>> msg [::-1]*
  9. *'olleh'*
  10. 3. Використовуючи зрізи видаліть афікси у наступних словоформах: *dish-es, run-ning*, *nation-ality, un-do, pre-heat*.
  11. *>>> msg = 'dishes, running, nationality, undo, preheat'*
  12. *>>> msg [:4]*
  13. *'dish'*
  14. *>>> msg [:11]*
  15. *'dishes, run'*
  16. *>>> msg [:23]*
  17. *'dishes, running, nation'*
  18. *>>> msg [:32]*
  19. *'dishes, running, nationality, un'*
  20. *>>> msg [39]*
  21. *'h'*
  22. *>>> msg [:39]*
  23. *'dishes, running, nationality, undo, pre'*

4. Напишіть for цикл, який виведе на екран символи стрічки *msg* по одному на рядок.

*msg = 'Python Shell'*

*>>> for char in msg:*

*print char,*

*P y t h o n S h e l l*

5. Напишіть for цикл, який обробить *phrase1* визначивши довжину кожного елементу і результати збереже в новому списку *lengths*. (Створіть пустий список *lengths = [].* Далі використовуйте метод *append()* в тілі циклу для додавання довжин до списку).

*>>> phrase = ['washington', 'lviv', 'kiev', 'london']*

*>>> lenghts = []*

*>>> for w in phrase:*

*lenghts.append(len(w))*

*print len(w), w*

*print lenghts*

*10 washington*

*[10]*

*4 lviv*

*[10, 4]*

*4 kiev*

*[10, 4, 4]*

*6 london*

*[10, 4, 4, 6]*

* 1. 6. Використайте функцію index() наступним чином ’inexpressible’.index(’e’). Що станеться якщо виконати ’inexpressible’.index(’re’).
  3. *>>> 'inexpressible'.index('e')*
  4. *2*
  5. *>>> 'inexpressible'.index('re')*
  6. *5*

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

*>>> phrase1 = [' ', 'car', 'printer', 'smoke', 'elevator']*

*>>> phrase1.index('smoke')*

*3*

*>>> phrase1.index('car')*

*1*

*>>> phrase1.index('elevator')*

*4*

*>>> phrase1.index('printer')*

*2*

**Висновок:** на лабораторній роботі я освоїв основи програмування на мові Python.