МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра САПР

ЗВІТ

ДО ЛАБОРАТОРНОЇ РОБОТИ №7

З ДИСЦИПЛІНИ «КОМП’ЮТЕРНА ЛІНГВІСТИКА»

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

***СТРУКТУРНЕ ПРОГРАМУВАННЯ МОВОЮ PYTHON (частина1)***

Виконала:

Студентка групи ПРЛм-12

Василишин В. В.

Перевірив:

Викладач кафедри САПР

Дупак Б. П.

2015

**Мета роботи:** Вивчення основ програмування на мові Python. Вивчення основ структурного програмування мовою Python. Повторення та закріплення знань отриманих при виконанні попередніх лабораторних робіт. Покращення загальних навичок у програмуванні.

**Теоретичні відомості**

Присвоювання – найпростіше поняття програмування, але навіть і йому властиві певні тонкощі.

За допомогою операції присвоєння відбувається копіювання значення виразу іншій змінній, хоча насправді, значення структурованого об’єкту, наприклад такого, як список, це є посилання на цей об’єкт.

Python підтримує два способи порівняння. Оператор is перевіряє об’єкти на ідентичність. Створивши список з декількох копій одного і того самого об’єкту не складно переконатися, що елементи цього списку не тільки ідентичні, згідно ==, але і є одним і тим самим об’єктом.

Серед послідовностей, в попередніх прикладах, використовувалися наступні: стрічки, списки та кортежі. Змінна, тип якої – кортеж, створюється за допомогою ком (#1) і переважно обмежується дужками. В попередніх лабораторних роботах кортежі використовувалися для представлення пар значень (елементів послідовності з двох членів). Зазвичай, кортежі можуть містити будь-яку кількість елементів та членів.

Кортежі створюються за допомогою ком (оператор - кома). Для створення кортежу, який містить єдиний елемент 'snark' , кома ставиться безпосередньо після цього елементу 'snark',. Пустий кортеж створюється за допомогою пустих дужок (t=()).

Способи ітерування елементів послідовностей Таблиця 1:

| **Вираз Python** | **Пояснення** |
| --- | --- |
| for item in s | Проітерувати елементи s |
| for item in sorted(s) | Проітерувати впорядковані елементи s |
| for item in set(s) | Проітерувати унікальні елементи s |
| for item in reversed(s) | Проітерувати зворотньо впорядковані елементи s |
| for item in set(s).difference(t) | Проітерувати елементи s, які не входять в t |
| for item in random.shuffle(s) | Проітерувати випадково впорядковані елементи s |

ТЕКСТИ ПРОГРАМ НА МОВІ PYTHON

ВАРІАНТ №2

1. Знайти в Python's help додаткову інформацію про послідовності. В інтерпретаторі, набрати по черзі help(str), help(list), та help(tuple). На екрані буде відображено повний список функцій властивих кожному з типів. Деякі функції мають спеціальні імена з подвійними підкреслюваннями. Кожній такій функції відповідає і інший запис показаний в документації. Наприклад x.\_\_getitem\_\_(y) відповідає x[y].

**>>> help (str)**

Help on class str in module \_\_builtin\_\_:

class str(basestring)

| str(object) -> string

|

| Return a nice string representation of the object.

| If the argument is a string, the return value is the same object.

|

| Method resolution order:

| str

| basestring

| object

|

| Methods defined here:

|

| \_\_add\_\_(...)

| x.\_\_add\_\_(y) <==> x+y

|

| \_\_contains\_\_(...)

| x.\_\_contains\_\_(y) <==> y in x

|

| \_\_eq\_\_(...)

| x.\_\_eq\_\_(y) <==> x==y

|

| \_\_format\_\_(...)

| S.\_\_format\_\_(format\_spec) -> string

|

...

**>>> help (list)**

Help on class list in module \_\_builtin\_\_:

class list(object)

| list() -> new empty list

| list(iterable) -> new list initialized from iterable's items

|

| Methods defined here:

|

| \_\_add\_\_(...)

| x.\_\_add\_\_(y) <==> x+y

|

| \_\_contains\_\_(...)

| x.\_\_contains\_\_(y) <==> y in x

|

| \_\_delitem\_\_(...)

| x.\_\_delitem\_\_(y) <==> del x[y]

|

...

**>>> help (tuple)**

Help on class tuple in module \_\_builtin\_\_:

class tuple(object)

| tuple() -> empty tuple

| tuple(iterable) -> tuple initialized from iterable's items

|

| If the argument is a tuple, the return value is the same object.

|

| Methods defined here:

|

| \_\_add\_\_(...)

| x.\_\_add\_\_(y) <==> x+y

|

| \_\_contains\_\_(...)

| x.\_\_contains\_\_(y) <==> y in x

|

| \_\_eq\_\_(...)

| x.\_\_eq\_\_(y) <==> x==y

|

...

2. Знайти три операції, які можна здійснювати і зі списками та із кортежами. Знайти три операції, які не можна здійснювати над кортежами. Знайдіть коли використання списку замість кортежу приводить до Python помилки.

- операції, які можна здійснювати і зі списками, і з кортежами

>>> kortezh = (2, 10, 3, 'hello', 'autumn', 'october')

>>> spysok = [2, 10, 3, 'hello', 'autumn', 'october']

>>> len (kortezh)

6

>>> len (spysok)

6

>>> kortezh\*2

(2, 10, 3, 'hello', 'autumn', 'october', 2, 10, 3, 'hello', 'autumn', 'october')

>>> spysok\*2

[2, 10, 3, 'hello', 'autumn', 'october', 2, 10, 3, 'hello', 'autumn', 'october']

>>> 2 in kortezh

True

>>> 'hello' in spysok

True

- операції, які не можна здійснювати з кортежами

>>> del (kortezh[0])

Traceback (most recent call last):

File "<pyshell#6>", line 1, in <module>

del (kortezh[0])

TypeError: 'tuple' object doesn't support item deletion

>>> kortezh.append ('go')

Traceback (most recent call last):

File "<pyshell#15>", line 1, in <module>

kortezh.append ('go')

AttributeError: 'tuple' object has no attribute 'append'

>>> kortezh.sort()

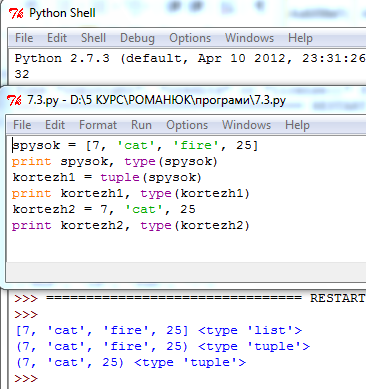
Traceback (most recent call last):

File "<pyshell#19>", line 1, in <module>

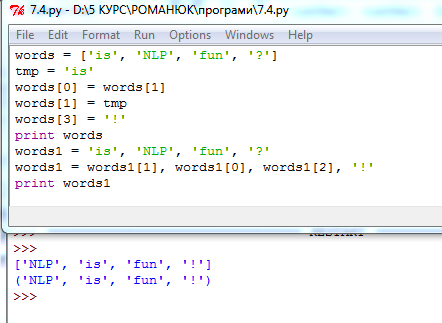
kortezh.sort()

AttributeError: 'tuple' object has no attribute 'sort'

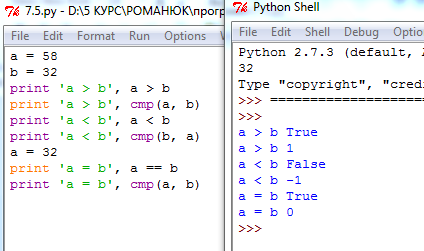
3. Яким чином можна створити кортеж з одного елемента. Продемонструвати два різні способи.



4. Створити список words = ['is', 'NLP', 'fun', '?']. Використовуючи операції присвоювання подібні до words[1] = words[2] та тимчасову змінну tmp перетворити цей список в список ['NLP', 'is', 'fun', '!']. Здійснити аналогічні перетворення використовуючи присвоювання в кортежах.

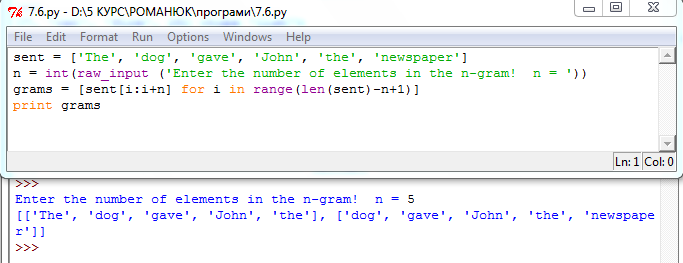


5. Прочитати про вбудовану функцію здійснення порівнянь cmp, набравши help(cmp). Продемонструвати чим поведінка цієї функції відрізняється від поведінки операторів порівняння.

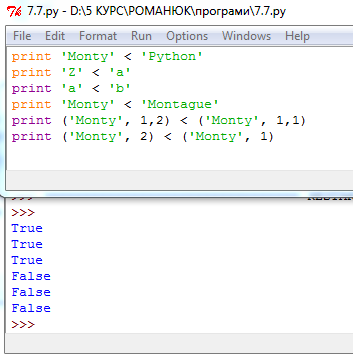


Навідміну від операторів порівнянь, які дають результат True або False, функція здійснення порівнянь cmp показує результат порівнння у вигляді -1, 0, 1.

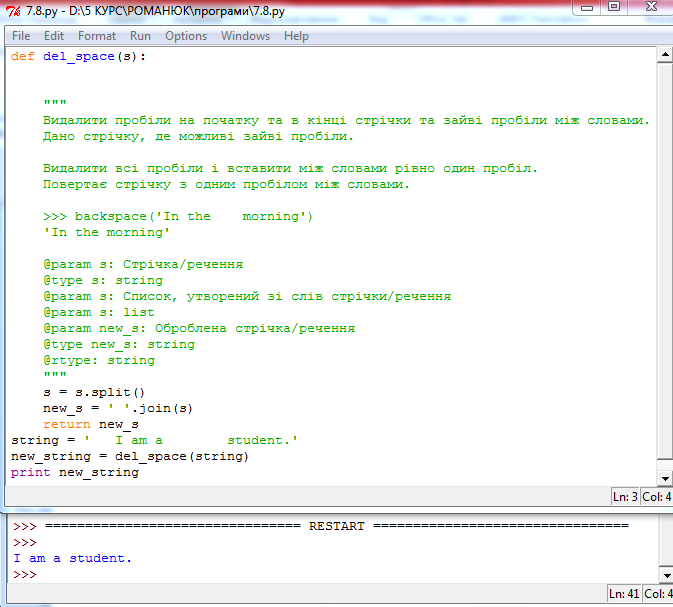
6. Написати програму для коректного виділення в тексті n-грамів з врахуванням граничних випадків: n = 1, та n = len(sent).



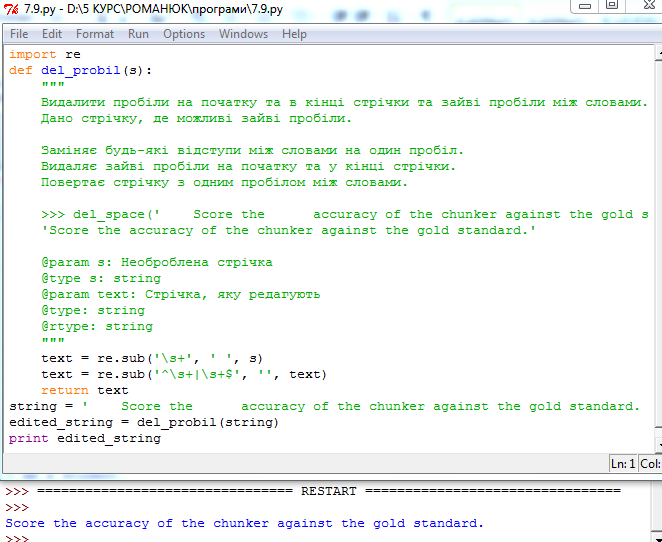
7. Використати оператори нерівності для порівняння стрічок, наприклад. 'Monty' < 'Python'. Що станеться, якщо виконати 'Z' < 'a'? Порівняти стрічки,як мають однаковий префікс, наприклад 'Monty' < 'Montague'. Спробувати порівняти структуровані об’єкти ,наприклад. ('Monty', 1) < ('Monty', 2). Чи отримали очікувані результати?



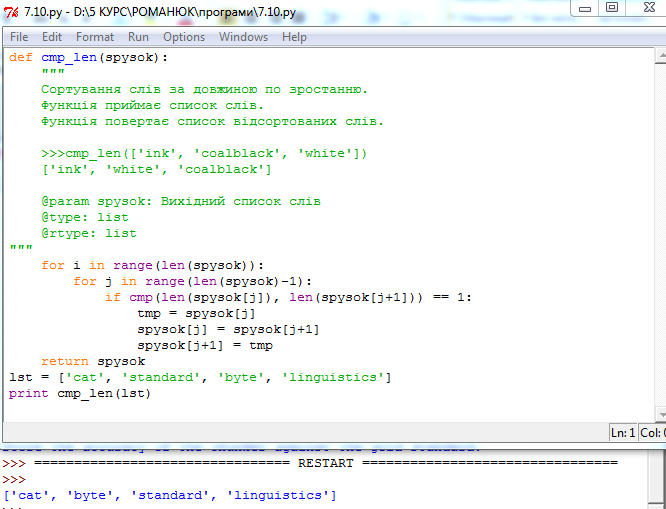
8. Написати програму видалення пробілів на початку і в кінці стрічки та для видалення зайвих пробілів між словами. Використовувати split() та join(). Оформити у вигляді функції. Функція повинна містити повну стрічку документування.



9. Написати програму видалення пробілів на початку і в кінці стрічки та для видалення зайвих пробілів між словами. Використовувати re.sub() . Оформити у вигляді функції. Функція повинна містити повну стрічку документування



10. Написати програму сортування слів за їх довжиною. Визначити допоміжну функцію cmp\_len, яка буде використовувати функцію cmp для порівняння довжин слів. Функція повинна містити повну стрічку документування.



**Висновки:** під час виконання даної лабораторної роботи були вивчені основи програмування мови Python, а саме основи структурного програмування.