



Олексій Васильєв

Мова програмування Python

Київ 2020



Лекція 5. Робота з текстом



- Текстовий літерал
- Операції з текстом
- Методи для роботи з текстом
- Приклади використання тексту



Текстовий літерал

Текстовий літерал є послідовністю символів, які розміщені у одинарних і подвійних лапках

Для включення в літерал одинарних і подвійних лапок можна використовувати інструкції `'` та `"`

Обернена нахилена риска `\` використовується для розбивки літералу на декілька рядків у вікні редактора коду

Подвійна нахилена риска `\\` використовується для вставки нахиленої риски у літерал

Керуючі символи:

- інструкція `\n` - перехід до нового рядка
- інструкція `\t` - табуляція

Три пари подвійних або одинарних лапок: такий текстовий літерал можна вводити (у вікні редактора) в декількох рядках (без використання оберненої риски як індикатора переносу рядка), і відобразатися він буде так, як уведений у вікні редактора



Текстові літерали

```
A="Мова 'Python' відрізняється від мови \"Java\"."
print(A)
B='Мова "Java" відрізняється від мови \'C++\''
print(B)
C="Сірий\tЖовтий\tЧервоний\nСиній\
\tБілий\tЗелений"
print(C)
print("\\ Ліна Костенко \\")
D="""Усе було - і сум, і самота,
    і горе втрат, і дружба нефальшива.
    А ця любов - як нитка золота,
    що й чорні дні життя мого прошила."""
print(D)
```

Програма (Strings.py)

```
Мова 'Python' відрізняється від мови "Java".
Мова "Java" відрізняється від мови 'C++'.
Сірий    Жовтий    Червоний
Синій    Білий    Зелений
\ Ліна Костенко \
Усе було - і сум, і самота,
    і горе втрат, і дружба нефальшива.
    А ця любов - як нитка золота,
    що й чорні дні життя мого прошила.
```



Префікси

Префікс — це спеціальний символ (або символи), котрий вказується безпосередньо перед текстовим літералом

Префікс **r** або **R**: літерал (сирий, необроблений) обробляється в режимі, в якому нахилена риска \ інтерпретується як звичайний символ

Префікси **f** або **F** використовують для створення форматуваних літералів. Форматовані літерали можуть містити поля заміни: спеціальні інструкції (у фігурних дужках), котрі визначають, які значення і в якому форматі мають бути добавлені у текстовий літерал у відповідному місці.

Всередині фігурних дужок вказується назва змінної, значення якої вставляється у відповідне місце в текстовому літералі. Також там можуть міститися додаткові коди, що визначають формат (вид і спосіб) представлення значення в літералі



Префікси

```
# Текстовий літерал без префіксу:  
A="\Java\"\n\"Python\""  
print(A)  
print("Символів:", len(A))  
# Текстовий літерал з префіксом:  
B=r"\Java\"\n\"Python\""  
print(B)  
print("Символів:", len(B))  
# Змінна з текстовим значенням:  
name="Python"  
# Текстовий літерал з префіксом:  
C=f"Мова {name} - прості і зрозуміла"  
print(C)  
C=f"Мова {name!r} - прості і зрозуміла"  
print(C)  
# Змінна з числовим значенням:  
num=12.34567  
# Текстовий літерал з префіксом:  
txt=f"Число: {num:9.3f}"  
print(txt)  
txt=f"Число: {num:09.3f}"  
print(txt)  
# Нове числове значення змінної:  
num=42  
# Формат для відображення цілого числа:  
txt=f"Число: {num:*>9d}"  
print(txt)
```

Програма (UsingPrefix.py)

```
# Формат для відображення  
# шістнадцяткового числа:  
txt=f"Число: {num:#09x}"  
print(txt)  
txt=f"Число: {num:9x}"  
print(txt)  
txt=f"Число: {num:*<9x}"  
print(txt)  
# Формат для відображення вісімкового числа:  
txt=f"Число: {num:*^#09o}"  
print(txt)  
# Формат для відображення двійкового числа:  
txt=f"Число: {num:#9b}"  
print(txt)
```

```
"Java"  
"Python"  
Символів: 15  
\Java\"\n\"Python\"  
Символів: 20  
Мова Python - проста і зрозуміла  
Мова 'Python' - проста і зрозуміла  
Число: 12.346  
Число: 00012.346  
Число: *****42  
Число: 0x000002a  
Число: 2a  
Число: 2a*****  
Число: **0o52***  
Число: 0b101010
```



Інструкції

Інструкція !r означає, що перед вставкою значення використовується спеціальне перетворення: викликається функція `repr()`, за допомогою якої розраховується текстове представлення і воно вставляється в літерал. В результаті значення "обростає" лапками

Символ **f** в полі заміни означає, що відображається число у форматі з плаваючою крапкою.

Інструкція **9.3f**: під змінну виділяється поле шириною в **9** позицій, після коми відображається **3** цифри

Інструкція **09.3f**: всі "зайві" позиції, виділені під число, будуть заповнені нулями

Інструкція *>9d: відображається ціле число (символ **d**), під яке виділяється **9** позицій.

Символ **>** означає, що вирівнювання значення виконується по правому краю.

Символ ***** означає, що "зайві" позиції будуть заповнені символом "зірочка" *****

Вирівнювання смісту по лівому краю: використовують символ **<**. Вирівнювання по центру: символ **^**. Символ **=** задає режим: якщо число відображається зі знаком, то між знаком і цифрами в представленні числа виконується розрив (так щоб число займало всю виділену під нього область), і цей розрив заповнюється відповідним символом (вказаним перед символом, який визначає спосіб вирівнювання)



Інструкції - 2

Символ **#** визначає спеціальний режим для відображення числових значень. Для кожного числового типу (цілі числа, числа з плаваючою точкою, комплексні числа) цей режим має свої особливості. Наприклад, для цілих чисел в двійковому, вісімковому і шістнадцятковому представленні даний режим означає, що перед власне значенням числа буде відображатися відповідний префікс **0b**, **0o** і **0x**

Інструкція **#09x**: відображення значення в шістнадцятковому вигляді з виділенням **9** позицій під число і заповненням "зайвих" позицій нулями. Через наявність символу **#** перед шістнадцятковим кодом відображається префікс **0x**

Інструкція ***<9x**: значення відображається в шістнадцятковому форматі, під число виділяється **9** позицій, вирівнювання виконується по лівому краю, "зайві" символи заповнюються "зірочкою" *****

Інструкція ***^#09o**: число відображається у вісімковому представленні, під число виділяється **9** позицій, в представленні числа відображається префікс **0o** (символ **#** в інструкції), вирівнювання виконується по центру (символ **^** в інструкції), "зайві" позиції заповнюються "зірочкою" *****

Інструкція **#9b**: числове значення відображається в двійковому представленні (символ **b** в інструкції), під значення виділяється поле шириною в **9** символів, а в представленні використовується префікс **0b** (символ **#** в інструкції)



Інструкції - 3

Параметри інструкцій:

- [1] Символ, котрим заповнюються "зайві" позиції.
- [2] Символ (<, >, ^ або =), визначає спосіб вирівнювання.
- [3] Режим відображення знаку для числових значень: якщо вказано "плюс" +, то знак буде відображатися і для додатних, і для від'ємних чисел. Якщо вказано "мінус" -, то знак буде відображатися тільки для від'ємних чисел.
- [4] Можна вказати пробіл: в цьому випадку для додатних чисел перед числом робиться відступ, а від'ємні числа відображаються зі знаком.
- [5] Символ #: спеціальний режим відображення значень.
- [6] Якщо для числових значень вказаний 0, то "зайві" позиції заповнюються нулями.
- [7] Числове значення: ширина поля для відображення значення.
- [8] Символ (кома , символ підкреслення _) для виділення тисячних розрядів.
- [9] Через крапку вказується точність відображення числового значення.
- [10] Символ, який задає тип відображуваного значення: **s** (текст), **b** (двійкове представлення числа), **c** (символ — цілочислове значення інтерпретується як код символу), **d** або **n** (ціле число), **o** (вісімкове представлення для числа), **x** або **X** (шістнадцяткове представлення для числа), **e** або **E** (експоненційне представлення для дійсного числа), **f** або **F** (число в форматі з плаваючою крапкою), **g** або **G** (загальний формат — спосіб відображення числового значення залежить від фактичного значення числа), % (проценти).



Метод format()

Програма (UsingFormat.py)

```
A="Число {}, текст {} і знову число {}"  
txt=A.format(123,"Python",321)  
print(txt)  
txt="Число {0} - це {0:b} або {0:x}".format(42)  
print(txt)  
txt="Код: {0:05d}, символ: {0:*^5c}".format(65)  
print(txt)  
txt="Число: {:_>+20.3E}".format(123.468)  
print(txt)  
B="{0:_{2}{1}s}"  
num=6  
for k in range(1,num+1):  
    print(B.format("*",k,">"),end="")  
    print(" "*(2*(num-k)),end="")  
    print(B.format("*",k,"<"))
```

```
Число 123, текст Python і знову число 321  
Число 42 - це 101010 або 2a  
Код: 00065, символ: **A**  
Число: _____+1.235E+02  
*           *  
_ *         * _  
_ *         * _  
_ *         * _  
_ *         * _  
_ *         * _  
_ **        _
```



Операції з текстом

Операції з текстом (A і B - текст):

[1] Конкатенація: **A+B**

[2] Конкатенація літералів через пробіл:
"Вивчаємо " "Python" (результат "Вивчаємо Python")

[3] Множення тексту на число: **3*"A"** (результат "AAA")

[4] Функція **ord()**: визначення коду символу

[5] Функція **chr()**: визначення символу за кодом

[6] Текст можна використовувати в операторі циклу **for**

[7] Для тексту можна виконувати зріз



Операції з текстом

```
# Вихідний текст:
txt="Hello Python"
print(txt)
# Текст в оберненому порядку:
A=txt[::-1]
print(A)
# Перше слово в тексті:
B=txt[:5]
print(B)
# Останнє слово в тексті:
C=txt[6:]
print(C)
# Змінна з текстовим значенням:
new_txt=""
# Змінна з цілочисловим значенням:
delta=ord("a")-ord("A")
# Перебір символів у тексті:
for s in txt:
    # Якщо буква в діапазоні від "a" до "z":
    if(ord(s)>=ord("a") and ord(s)<=ord("z")):
        s=chr(ord(s)-delta)
    # Додавання символу до тексту:
    new_txt+=s
# Текст з великих букв:
print(new_txt)
```

Програма (UsingText.py)

```
Hello Python
nohtyP olleH
Hello
Python
HELLO PYTHON
```



Завдання - 1

Напишіть програму, в якій на основі уведеного тексту створюється новий текст, в якому сусідні символи міняються місцями: перший з другим, третій з четвертим, и так далі

Завдання - 2

Напишіть програму для шифрування тексту: кожен символ замінюється на наступну букву в алфавіті (а остання буква алфавіту - на першу)



Методи

upper() - текст, в якому всі букви великі

lower() - текст, з маленьких букв

swapcase() - великі букви замінені на маленькі, а маленькі — на великі

title() - кожне слово починається з великої букви

capitalize() - перше слово починається з великої букви

isalnum() - результат **True** якщо текст непустий і складається з букв та/або чисел

isalpha() - результат **True** якщо текст непустий і складається з букв

isascii() - значення **True** якщо текст пустий або містить символи кодової таблиці

isdecimal() - результат **True**, якщо текст непустий і містить лише десяткові цифри

isdigit() - результат **True**, якщо текст непустий і містить лише цифри

isidentifier() - результат **True**, якщо текст містить назву зареєстрованого в мові ідентифікатору або ключового слова

islower() - значення **True**, якщо текст непустий і складається з маленьких букв

isnumeric() - результат **True**, якщо текст непустий і складається з числових символів

isprintable() - значення **True**, якщо текст пустий або складається з друкованих (тих, які відображаються в області виводу) символів

isspace() - значення **True**, якщо текст непустий і складається з пробілів

istitle() - результат **True**, якщо текст непустий і кожне слово в тексті починається з великої букви

isupper() - значення **True**, якщо текст непустий і складається з великих букв



Методи - 2

Пошук символу: методи **find()**, **rfind()**, **index()** і **rindex()**. Аргумент - шуканий підрядок.

Методи **find()** і **index()**: результат - індекс першого вхождення підрядка у текст. Якщо підрядка в тексті немає: метод **find()** повертає значення **-1**, метод **index()** генерує виключення класу **ValueError**.

Методи **rfind()** і **rindex()** відрізняється від методів **find()** і **index()**: виконується пошук не першого, а останнього вхождення підрядка у текст.

count() - кількість входжень підрядка або символу в тексті

endswith() - чи закінчується текст підрядком, переданим аргументом

startswith() - чи починається текст підрядком, переданим аргументом

expandtabs() - текст, в якому інструкції табуляції замінені на відповідну кількість пробілів

strip(), **lstrip()** і **rstrip()**: дозволяють видалити початкові та/або кінцеві символи в тексті

replace() - заміна в тексті одного підрядка на інший підрядок



Методи - 3

join() - створює рядок на основі списку

partition() - розбиває текст на три частини. Аргумент - роздільник для розбивки тексту на блоки. Точка розбивки - місце, де роздільник зустрічається перший раз. Результат - кортеж: текст до роздільника, роздільник, текст після роздільника

rpartition() - аналогічний методу **partition()**, але розбивка - в місці, де роздільник останній раз зустрічається в тексті

split() - результат - список слів у тексті

rsplit() - аналогічний методу **split()**, але розбивка на слова виконується справа наліво

splitlines() - список підрядків, на які розбивається вихідний текст. Роздільник - інструкція переходу до нового рядка

center() - рядок, вирівняний по центру поля, виділеного для відображення тексту. Ширина поля (в символах) - аргумент методу

ljust() і **rjust()** - аналогічні методу **center()**, але методом **ljust()** вирівнювання виконується за лівим краєм, а методом **rjust()** текст вирівнюється за правим краєм



Приклади

```
txt="Мова PYTHON простіша за мову JAVA!"  
print(txt)  
print(txt.upper())  
print(txt.lower())  
print(txt.swapcase())  
print(txt.title())  
print(txt.capitalize())
```

Програма (Text_01.py)

```
Мова PYTHON простіша за мову JAVA!  
MOBA PYTHON ПРОСТИША ЗА МОВУ JAVA!  
мова python простіша за мову java!  
MOBA python ПРОСТИША ЗА МОВУ java!  
Мова Python Простіша За Мову Java!  
Мова python простіша за мову java!
```

Програма (Text_02.py)

```
txt=input("Уведіть текст: ")  
symb=input("Яку букву знайти? ")  
num=txt.count(symb)  
if num==0:  
    print("Такої букви в тексті немає!")  
else:  
    print(f"В тексті {num} букв(и) '{symb}'")
```

Уведіть текст: **Програмувати слід правильно**
Яку букву знайти? **p**
В тексті 3 букв(и) 'p'

Уведіть текст: **Програмувати слід правильно**
Яку букву знайти? **i**
Такої букви в тексті немає!



Приклади - 2

Програма (Text_03.py)

```
txt=input("Уведіть текст: ")
symb=input("Яку букву знайти? ")
num=txt.find(symb)
L=[]
while num!=-1:
    L.append(num)
    num=txt.find(symb,num+1)
if len(L)==0:
    print("Такої букви в тексті немає!")
else:
    print(f"Позиції букви '{symb}' в тексті: {L}")
```

Уведіть текст: **Програмувати слід правильно**
Яку букву знайти? **p**
Позиції букви 'p' в тексті: [1, 4, 19]

Уведіть текст: **Програмувати слід правильно**
Яку букву знайти? **і**
Такої букви в тексті немає!



Приклади - 3

Програма (Text_04.py)

```
txt="Ми вивчаємо мову Python"  
print(txt)  
A=txt.replace(" ","_*_")  
print(A)  
B=txt.replace(" ","\n")  
print(B)  
C=txt.replace(" "," не ",1).replace("Python","Java")  
print(C)  
D=txt.replace("мову ","")  
print(D)
```

```
Ми вивчаємо мову Python  
Ми_*_вивчаємо_*_мову_*_Python  
Ми  
вивчаємо  
мову  
Python  
Ми не вивчаємо мову Java  
Ми вивчаємо Python
```



Приклади - 4

Програма (Text_05.py)

```
A=["Alpha","Bravo","Charlie"]
print("Список:",A)
B=", ".join(A)
print("Текст:",B)
C=B.split(", ")
print("Знову список:",C)
txt="""Десь там планети в просторі безмеж
    Яка сумна у безвісті ночівля!
    А може ми їм світимося теж?
    А може ми їм зіронька вечірня?"""
print(txt)
D=txt.splitlines()
print(D)
```

```
Список: ['Alpha', 'Bravo', 'Charlie']
Текст: Alpha, Bravo, Charlie
Знову список: ['Alpha', 'Bravo', 'Charlie']
Десь там планети в просторі безмеж
    Яка сумна у безвісті ночівля!
    А може ми їм світимося теж?
    А може ми їм зіронька вечірня?
['Десь там планети в просторі безмеж',
 '    Яка сумна у безвісті ночівля!', '    А може ми їм
світимося теж? ',
 '        А може ми їм зіронька вечірня?']
```



Приклади - 5

Програма (Text_06.py)

```
txt="PYTHON"  
num=20  
A=txt.ljust(num, "_")  
B=txt.center(num)  
C=txt.rjust(num, "*")  
print("|",A,"|")  
print("|",B,"|")  
print("|",C,"|")
```

```
| PYTHON_____ |  
|          PYTHON          |  
| *****PYTHON***** |
```



Домашнє завдання

[1] Напишіть програму, в якій користувач уводить два текстових значення, і на їх основі створюється новий текст. В цей новий текст по черзі додаються букви з текстів, уведених користувачем. Коли один з текстів закінчується, як символ з цього тексту використовується "зірочка" *

[2] Напишіть програму, в якій на основі тексту, уведеного користувачем, створюється новий текст. У порівнянні з вихідним, в ньому слова розміщені в зворотному порядку. Під словами маються на увазі блоки тексту, розділені пробілами.