Зображення, що містить текст

Автоматично згенерований опис

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

**Лабораторна робота №1**

Аналіз текстів з використанням мови Python

**Тема:** Робота з текстовими даними в Python

**Варіант:** 1

Виконав Перевірив:

студент групи ІП-11: Тимофєєва Ю. С

Панченко С. В.

Київ 2023

Зміст

[1 Мета лабораторної роботи 6](#__RefHeading___Toc294_2305880923)

[2 Завдання 7](#__RefHeading___Toc296_2305880923)

[3 Виконання 8](#__RefHeading___Toc298_2305880923)

[3.1 Робота з текстом 8](#__RefHeading___Toc300_2305880923)

[3.2 Робота з регулярними виразами 14](#__RefHeading___Toc302_2305880923)

[4 Висновок 16](#__RefHeading___Toc304_2305880923)

# Мета лабораторної роботи

Ознайомитись з представленням тексту Python в та регулярними виразами.

# Завдання

Створити програму, яка:

1. Зчитує текстовий файл відповідно до варіанту як рядок. За допомогою зрізів виділити частину тексту в окрему змінну-рядок та використати описані в теоретичних відомостях функції та методи для роботи з рядками.
2. За допомогою регулярних виразів:

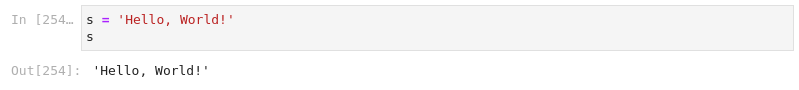
Варіант 1.

Знайти всі номери телефонів та замінити зірочками всі цифри після першої. Файл text1.

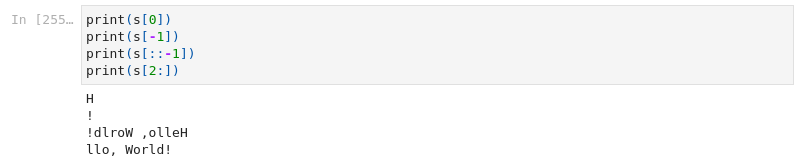
# Виконання

## Робота з текстом

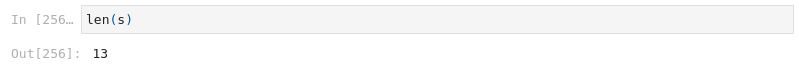
Запишемо рядок.

  
  
Рисунок 3.1 - Запис рядка

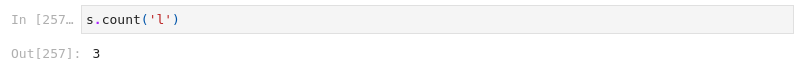
Виведемо певні символи рядка за допомогою індексів.

  
  
Рисунок 3.2 - Виведення символів та підрядків даного рядка за допомогою індексів

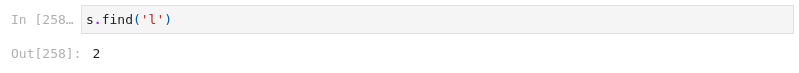
Знайдемо довжину рядка, застосувавши функцію len.

  
  
Рисунок 3.3 - Знаходження довжини рядка

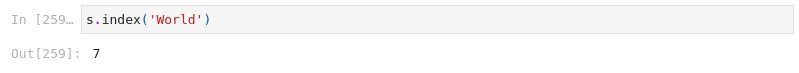
Підрахуємо кількість входжень певного елементу за допомогою методу count.

  
  
Рисунок 3.4 - Підрахунок кількості входжень певного елементу в рядку

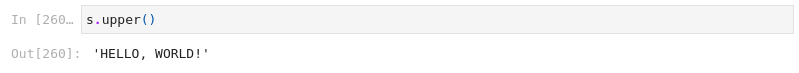
Знайдемо перший індекс входження певного елементу за допомогою методу find.

  
  
Рисунок 3.5 - Знаходження першого індексу певного елемента в рядку

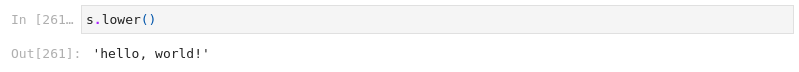
Знаходження індексу початку входження підрядка за допомогою методу index.

  
  
Рисунок 3.6 - Знаходження початку підрядка

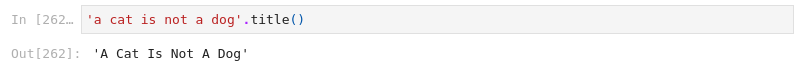
Застосуємо метод upper для виведення символів рядка у верхньому регістрі.

  
  
Рисунок 3.7 - Виведення рядка у верхньому регістрі

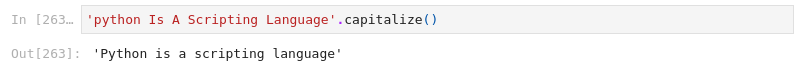
Застосуємо метод upper для виведення символів рядка у нижньому регістрі.

  
  
Рисунок 3.8 - Виведення рядка у нижньому регістрі

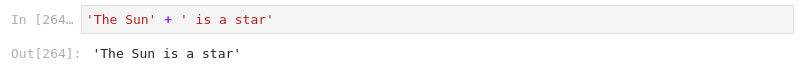
Виведення рядка, де перші літери кожного слова знаходяться у верхньому регістрі. Доречніше застосувати тут іншу фразу, що починається з маленької букви.

  
  
Рисунок 3.9 - Виведення рядка зі словами, що всі починаються з великої букви

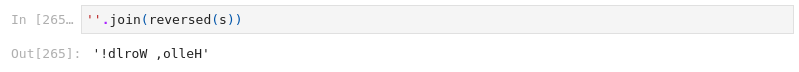
Виведення рядка, де лише перший символ буде у верхньому регістрі.

  
  
Рисунок 3.10 - Виведення рядка, у якому лише перша літера першого слова починається у верхньому регістрі

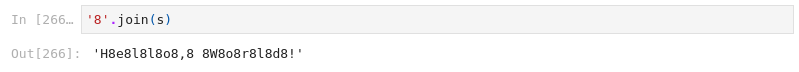
Виконаємо конкатенацію рядків.

  
  
Рисунок 3.11 - Конкатенація рядків

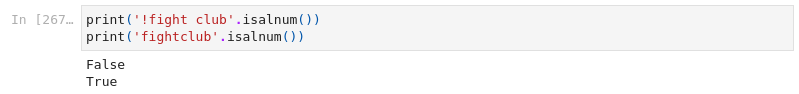
Виведення зворотного рядка за допомогою функції reversed.

  
  
Рисунок 3.12 - Виведення зворотного рядка

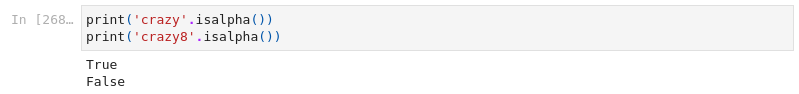
Вставка певного символу між символами рядка.

  
  
Рисунок 3.13 - Вставка символу між елементами рядка

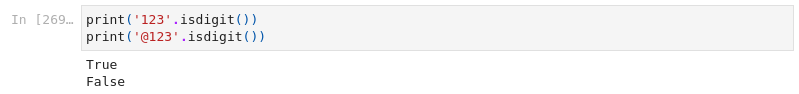
Виведення інформації про те, чи є усі символи або цифрами, або буквами, за допомогою методу isalnum.

  
  
Рисунок 3.14 - Робота методу isalnum

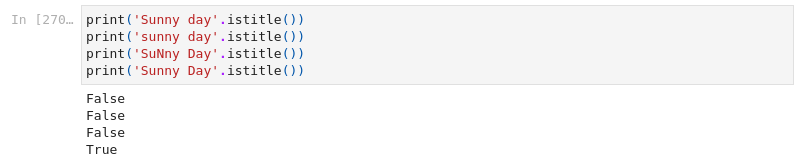
Перевірка того, чи всі символи рядка - букви. Використаємо метод isalpha.

  
  
Рисунок 3.15 - Робота методу isalpha

Перевірка того, чи всі символи рядка - цифри. Використаємо метод isdigit.

  
  
Рисунок 3.16 - Робота методу isdigit

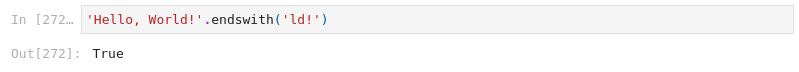
Перевірка того, чи кожне слово даного рядка починається з великої літери, а всі інші букви - з маленької. Використаємо метод istitle.

  
  
Рисунок 3.17 - Робота методу istitle

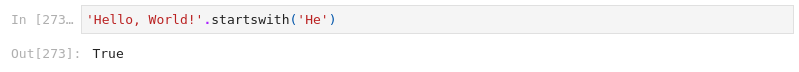
Перевірка того, чи всі символи рядка є white-space`ами. Використаємо метод isspace.

  
  
Рисунок 3.18 - Робота методу isspace

Перевірка того, чи закінчується рядок даним підрядком. Використаємо метод endswith.

  
  
Рисунок 3.19 - Робота методу endswith

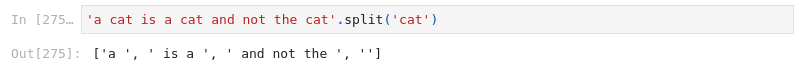
Перевірка того, чи починається рядок даним підрядком. Використаємо метод startswith.

  
  
Рисунок 3.20 - Робота методу startswith

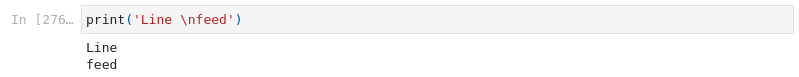
Заміна певного підрядка іншим підряком. Використаємо метод replace.

  
  
Рисунок 3.21 - Робота методу replace

Розіб'ємо рядок за підрядками. Використаємо метод split.

  
  
Рисунок 3.22 - Робота методу split

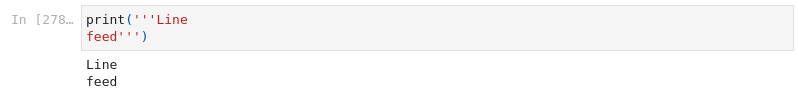
Використаємо символи форматування. Наприклад \n.

  
  
Рисунок 3.23 - Використання символів форматування

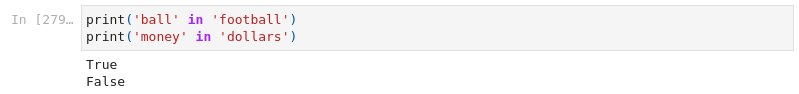
Виведення рядка з символами формутування без відображення їхньої дії. Застосуємо префікс r.

  
  
Рисунок 3.24 - Застосування префікса r

Покажемо застосування рядків з потрійними лапками.

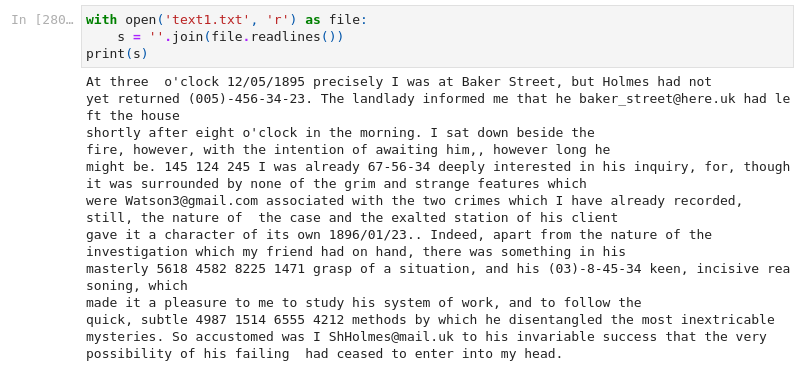
  
  
Рисунок 3.25 - Використання потрійних лапок

Виведення інформації про те, чи міститься або не міститься даний підрядок у рядку за допомогою ключових слів not та in, їхнього поєднання not in.

  
  
Рисунок 3.26 - Застосування in та not in

## Робота з регулярними виразами

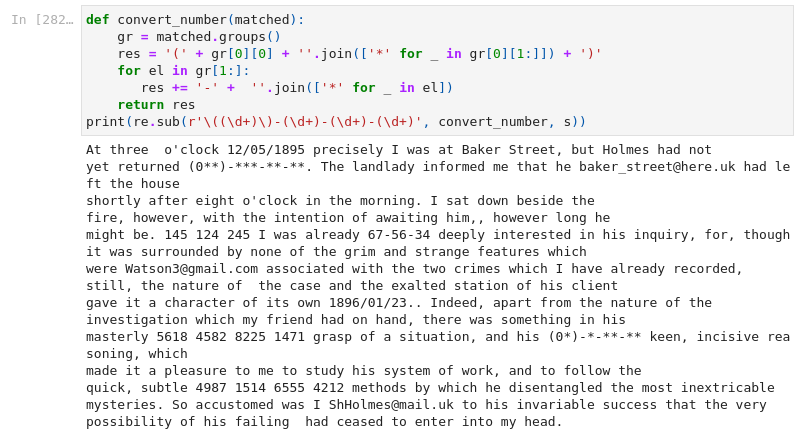
Зчитаємо файл. with - оператор контексту, який автоматично закриває файл.

  
  
Рисунок 3.27 - Зчитування файлу

Імпортуємо модуль "re" для виконання роботи з регулярними виразами.

  
  
Рисунок 3.28 - Імпортування модуля "re"

Знайдемо всі номери телефонів та замінимо зірочками всі цифри після першої у номерах телефонів. Для цього створимо функцію convert\_number, яку передамо у метод sub.

  
  
Рисунок 3.29 - Знаходження та перетворення всіх номерів телефонів

# Висновок

Під час виконання даної лабораторної роботи я ознайомився з представленням тексту Python в та регулярними виразами.

Було використано методи класу str, які показують, як працювати з текстом. Далі імпортовано модуль re та застосовано його функцію sub для перетворення номерів телефонів за допомогою відповідного регулярного виразу.