



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

студент групи ІП-11:

Панченко С. В.

Перевірів:

Тимофєєва Ю. С

Київ 2023

## ЗМІСТ

1 Мета лабораторної роботи.....	6
2 Завдання.....	7
3 Виконання.....	8
3.1 Робота з текстом.....	8
3.2 Робота з регулярними виразами.....	14
4 Висновок.....	16

## 1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Ознайомитись з представленням тексту Python в та регулярними виразами.

## 2 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.

## 3 ВИКОНАННЯ

### 3.1 Робота з текстом

Запишемо рядок.

```
In [254]: s = 'Hello, World!'
s
Out[254]: 'Hello, World!'
```

Рисунок 3.1 - Запис рядка

Виведемо певні символи рядка за допомогою індексів.

```
In [255]: print(s[0])
print(s[-1])
print(s[::-1])
print(s[2:])
H
!
!dlroW ,olleH
llo, World!
```

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

Знайдемо довжину рядка, застосувавши функцію len.

```
In [256]: len(s)
Out[256]: 13
```

Рисунок 3.3 - Знаходження довжини рядка

Підрахуємо кількість входжень певного елементу за допомогою методу count.

```
In [257]: s.count('l')
Out[257]: 3
```

### Рисунок 3.4 - Підрахунок кількості входжень певного елементу в рядку

Знайдемо перший індекс входження певного елементу за допомогою методу `find`.

```
In [258]: s.find('l')  
Out[258]: 2
```

### Рисунок 3.5 - Знаходження першого індексу певного елемента в рядку

Знаходження індексу початку входження підрядка за допомогою методу `index`.

```
In [259]: s.index('World')  
Out[259]: 7
```

### Рисунок 3.6 - Знаходження початку підрядка

Застосуємо метод `upper` для виведення символів рядка у верхньому регістрі.

```
In [260]: s.upper()  
Out[260]: 'HELLO, WORLD!'
```

### Рисунок 3.7 - Виведення рядка у верхньому регістрі

Застосуємо метод `lower` для виведення символів рядка у нижньому регістрі.

```
In [261]: s.lower()  
Out[261]: 'hello, world!'
```

### 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.

```
In [262...] 'a cat is not a dog'.title()
Out[262]: 'A Cat Is Not A Dog'
```

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

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

```
In [263...] 'python Is A Scripting Language'.capitalize()
Out[263]: 'Python is a scripting language'
```

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

Виконаємо конкатенацію рядків.

```
In [264...] 'The Sun' + ' is a star'
Out[264]: 'The Sun is a star'
```

Рисунок 3.11 - Конкатенація рядків

Виведення зворотного рядка за допомогою функції `reversed`.

```
In [265...] ''.join(reversed(s))
Out[265]: '!dlrow ,olleH'
```

Рисунок 3.12 - Виведення зворотного рядка

Вставка певного символу між символами рядка.

```
In [266...] '8'.join(s)
Out[266]: 'H8e8l8l8o8,8 8W8o8r8l8d8!'
```

Рисунок 3.13 - Вставка символу між елементами рядка

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

```
In [267...] print('!fight club'.isalnum())
              print('fightclub'.isalnum())
False
True
```

Рисунок 3.14 - Робота методу `isalnum`

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

```
In [268...] print('crazy'.isalpha())
              print('crazy8'.isalpha())
True
False
```

Рисунок 3.15 - Робота методу `isalpha`

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

```
In [269...] print('123'.isdigit())
              print('@123'.isdigit())
True
False
```

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`.



```
In [270...] print('Sunny day'.istitle())
print('sunny day'.istitle())
print('SuNny Day'.istitle())
print('Sunny Day'.istitle())

False
False
False
True
```

Рисунок 3.17 - Робота методу istitle

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

```
In [271...] print('jjj\t '.isspace())
print(' \t'.isspace())

False
True
```

Рисунок 3.18 - Робота методу isspace

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

```
In [272...] 'Hello, World!'.endswith('ld!')

Out[272]: True
```

Рисунок 3.19 - Робота методу endswith

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

```
In [273...] 'Hello, World!'.startswith('He')

Out[273]: True
```

Рисунок 3.20 - Робота методу startswith

Заміна певного підрядка іншим підрядком. Використаємо метод `replace`.

```
In [274...] 'A friend in need is a friend indeed'.replace('friend', 'comrade')
Out[274]: 'A comrade in need is a comrade indeed'
```

Рисунок 3.21 - Робота методу `replace`

Розіб'ємо рядок за підрядками. Використаємо метод `split`.

```
In [275...] 'a cat is a cat and not the cat'.split('cat')
Out[275]: ['a ', ' is a ', ' and not the ', '']
```

Рисунок 3.22 - Робота методу `split`

Використаємо символи форматування. Наприклад `\n`.

```
In [276...] print('Line \nfeed')
Line
feed
```

Рисунок 3.23 - Використання символів форматування

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

```
In [277...] print(r'Line \nfeed')
Line \nfeed
```

Рисунок 3.24 - Застосування префікса `r`

Покажемо застосування рядків з потрійними лапками.

```
In [278...] print('Line
feed')
```

Line  
feed

Рисунок 3.25 - Використання потрібних 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`.

```
In [279...] print('ball' in 'football')
print('money' in 'dollars')
```

True  
False

Рисунок 3.26 - Застосування `in` та `not in`

## 3.2 Робота з регулярними виразами

Зчитуємо файл. `with` - оператор контексту, який автоматично закриває файл.

```
In [280...] with open('text1.txt', 'r') as file:
    s = ''.join(file.readlines())
    print(s)
```

At three o'clock 12/05/1895 precisely I was at Baker Street, but Holmes had not yet returned (005)-456-34-23. The landlady informed me that he baker\_street@here.uk had left the house shortly after eight o'clock in the morning. I sat down beside the fire, however, with the intention of awaiting him,, however long he might be. 145 124 245 I was already 67-56-34 deeply interested in his inquiry, for, though it was surrounded by none of the grim and strange features which were Watson3@gmail.com associated with the two crimes which I have already recorded, still, the nature of the case and the exalted station of his client gave it a character of its own 1896/01/23.. Indeed, apart from the nature of the investigation which my friend had on hand, there was something in his masterly 5618 4582 8225 1471 grasp of a situation, and his (03)-8-45-34 keen, incisive reasoning, which made it a pleasure to me to study his system of work, and to follow the quick, subtle 4987 1514 6555 4212 methods by which he disentangled the most inextricable mysteries. So accustomed was I ShHolmes@mail.uk to his invariable success that the very possibility of his failing had ceased to enter into my head.

Рисунок 3.27 - Зчитування файлу

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

```
In [281...] import re
```

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`.

```
In [282...] def convert_number(matched):
              gr = matched.groups()
              res = '(' + gr[0][0] + ''.join(['*' for _ in gr[0][1:]]) + ')'
              for el in gr[1:]:
                  res += '-' + ''.join(['*' for _ in el])
              return res
              print(re.sub(r'\((\d+)\)-(\d+)-(\d+)-(\d+)', convert_number, s))
```

At three o'clock 12/05/1895 precisely I was at Baker Street, but Holmes had not yet returned (0\*\*)-\*\*\*-\*\*-\*\*. The landlady informed me that he baker\_street@here.uk had left the house shortly after eight o'clock in the morning. I sat down beside the fire, however, with the intention of awaiting him,, however long he might be. 145 124 245 I was already 67-56-34 deeply interested in his inquiry, for, though it was surrounded by none of the grim and strange features which were Watson3@gmail.com associated with the two crimes which I have already recorded, still, the nature of the case and the exalted station of his client gave it a character of its own 1896/01/23.. Indeed, apart from the nature of the investigation which my friend had on hand, there was something in his masterly 5618 4582 8225 1471 grasp of a situation, and his (0\*)-\*--\*\* keen, incisive reasoning, which made it a pleasure to me to study his system of work, and to follow the quick, subtle 4987 1514 6555 4212 methods by which he disentangled the most inextricable mysteries. So accustomed was I ShHolmes@mail.uk to his invariable success that the very possibility of his failing had ceased to enter into my head.

Рисунок 3.29 - Знаходження та перетворення всіх номерів телефонів

## 4 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.