Лабораторна робота №1

Робота з текстовими даними в Python

Мета роботи: Ознайомитись з представленням тексту **Python** в та регулярними виразами.

Короткі теоретичні відомості

Python можна завантажити на https://www.python.org/downloads/

Або можна скористатись Google Colab, щоб виконувати роботи онлайн на хмарних серверах

https://colab.research.google.com/

У Python текст представляється у вигляді рядка, який є об'єктом класу str. Рядок є незмінною послідовністю кодових точок або символів Unicode.

Під час аналізу даних може виникнути потреба роботи з об'єктами str, наприклад, виправлення помилок, тому корисно знати основні методи цього класу.

```
text="It's a cat!"
```

До елементів рядку можна звертатись так само, як до елементів масиву, а також робити зрізи за схемою [початок:кінець:крок]:

```
text[0]          'I'
text[:4]         "It's"
text[::2]        "I'  a!"
text[-1]         '!'
```

Можна знайти довжину рядка:

```
len(text)
```

Та використати інші методи:

`text.count("t")` кількість літер t

`text.find("s")` індекс літери s

`text.index("cat")` індекс, з якого починається cat

text.upper(), text.lower(), text.title(), text.capitalize() і т.д.

За замовчанням початок=0, крок=1. Якщо вказано від'ємний крок, то елементи починаються з кінця і закінчуються початком.

Метод join об'єднує передані йому елементи, використовуючи вказаний рядок як роздільник:

```
"?".join(text)    "I?t?'?s? ?a? ?c?a?t?!"
```

Також існують методи, що перевіряють наявність літер, цифр, пробілів і т.д.:

text.isalnum(), a.isalpha(), text.isdigit(), text.istitle(), text.isspace(),text.endswith('!'), a.startswith('c').

Можна замінити елемент рядка:

```
text.replace("cat", "dog")
```

Або розбити рядок на вказаному символі:

```
text.split("a")
```

Також під час роботи з рядками допомагають регулярні вирази, які описують шаблони тексту. Потрібно імпортувати відповідний модуль, і створити вираз за допомогою методу compile, який приймає «сирий» рядок (перед рядком зазначається літера r):

```
import re

n = re.compile(r'(\d\d\d)-(\d\d)')
```

Символ \d представляє будь-яку цифру: \D - навпаки, будь-який символ, що не є цифрою; \w - літери чи цифри; \W – навпаки; \s - пробіл, табуляцію чи новий рядок; \S - навпаки.

Далі у створеного об’єкту – регулярного виразу можна викликати метод search та передати рядок, в якому необхідно знайти вираз. Метод search поверне об’єкт Match, у якого викликається метод group(), що й поверне знайдений вираз. Також можна використовувати метод findall, що поверне всі збіги.

```
room = n.search('Room number is 304-18.')

print('Room ' + room.group())
```

Якщо відділити дужками групи у регулярному виразі, то можна передати в метод group() номер групи.

```
room.group(0) '304-18'
room.group(1) '304'
room.group(2) '18'
```

Символ | в регулярному виразі означає «або», ? - необов’язковий збіг.

```
n = re.compile(r'(\d\d\d-)?\d\d')

room = n.search('Room number is 18.')

print('Room ' + room.group())     Room 18
```

Символ * означає нуль або більше збігів, + - один або більше. Фігурні дужки {} означають кількість або діапазон повторень.

```
n = re.compile(r'(\d){2}')
```

```
room = n.search('Number is 183.')
print('Number ' + room.group())
n = re.compile(r'(\d){1,4}')
```

```
Number 18
```

У квадратних дужках можна вказувати всі символи, для яких шукати збіги.

```
n = re.compile(r'[a-zA-Z0-9]')
room = n.findall('Room number is 304-18')
```

```
['R', 'o', 'o', 'm', 'n', 'u', 'm', 'b', 'e', 'r', 'i', 's', '3', '0', '4', '1', '8']
```

[^a-zA-Z0-9] означає всі символи, окрім літер та цифр.

Символ ^ означає, що збіг з регулярним виразом повинен бути на початку рядка, $ - в кінці. Символ . означає будь-який символ, окрім переносу рядка. re.IGNORECASE або re.I дозволяє ігнорувати регістр.

Метод sub дозволяє замінювати регулярні вирази.

```
n = re.compile(r'black (\\w)\\w*')
n.sub(r'\\1****', \"It's a black cat!\")
```

Можна прочитати текстовий файл як рядок:

```
text_file = open("test.txt", "r")
text = text_file.read()
text_file.close()
```

Завдання до лабораторної роботи

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.
   Варіант 2. Знайти та вивести всі адреси електронної пошти. Файл text1.
   Варіант 3. Знайти всі дати та перевести в один формат. Файл text1.
   Варіант 4. Знайти всі банківські картки і замінити будь-яким символом всі цифри, крім двох перших. Файл text1.
   Варіант 5. Знайти і видалити такі помилки, як подвійні пробіли та знаки пунктуації. Файл text1.
   Варіант 6. Знайти всі номери телефонів та замінити зірочками всі цифри, крім останньої. Файл text2.

Варіант 7. Знайти та вивести всі адреси електронної пошти. Файл text2.
Варіант 8. Знайти всі відмітки часу та перевести в один формат. Файл text2.
Варіант 9. Знайти всі номери будинків та перевернути їх (записати у зворотному порядку). Файл text2.
Варіант 10. Знайти і видалити такі помилки, як зайві пробіли та знаки пунктуації. Файл text2.
Варіант 11. Знайти всі номери телефонів та замінити будь-яким символом всі цифри, крім перших двох. Файл text3.
Варіант 12. Знайти всі адреси електронної пошти та замінити зірочками їхню частину після @. Файл text3.
Варіант 13. Знайти всі дати та перевести в один формат. Файл text3.
Варіант 14. Знайти всі номери будинків та замінити будь-яким символом все після першої цифри. Файл text3.
Варіант 15. Знайти і видалити голосні літери та цифри, крім 3. Файл text3.

Оформити звіт. Звіт повинен містити:

— титульний лист;
— код програми;
— результати виконання коду;

Продемонструвати роботу програми та відповісти на питання стосовно теоретичних відомостей та роботи програми.