



**Олексій Васильєв**

# **Мова програмування Python**

**Київ 2020**



## Лекція 8. Робота з файлами



- Зчитування вмісту файлу
- Запис даних в файл
- Порядкове зчитування файлу
- Режими доступу до файлів
- Копіювання файлів



# Зчитування файлу

- Відкрити файл: функція `open()`
- Аргументи функції `open()`:
  - [1] повний шлях до файлу (обов'язковий)
  - [2] інструкція, що визначає режим доступу (не обов'язковий)
- Файловий об'єкт: методи `write()` (запис у файл) і `read()` (зчитування з файлу)
- Метод `close()`: закриває файл

```
# Відкриваємо текстовий файл для зчитування:  
mf=open("D:\\Files\\Python\\poetry.txt")  
# Зчитується вміст файлу:  
txt=mf.read()  
print("Вміст файлу:")  
# Відображається вміст файлу:  
print(txt)  
# Закриваємо файл:  
mf.close()  
print("Файл закрито...")
```

Програма (ReadFile.py)

```
Вміст файлу:  
Вже рік старий за обрії пливе.  
Уже й Новий стоїть на виднокрузі.  
Хай у Новому буде все нове, -  
старі лиш вина, істини і друзі!  
Ліна Костенко  
Файл закрито...
```



# Порядкове зчитування

```
# Відкриваємо текстовий файл для зчитування:
mf=open("D:\\Files\\Python\\poetry.txt")
# Змінна для нумерації рядків:
k=1
# Порядокове зчитування файлу:
print("Порядкове зчитування файлу")
for L in mf:
    # Відображення номера рядка і самого рядка:
    print("[ "+str(k)+" ]",L,end="")
    # Нове значення для номера рядка:
    k+=1
# Закриваємо файл:
mf.close()
print("\nФайл закрито...")
```

Програма (ReadFile\_2.py)

```
Порядкове зчитування файлу
[1] Вже рік старий за обрії пливе.
[2] Уже й Новий стоїть на виднокрузі.
[3] Хай у Новому буде все нове, -
[4] старі лиш вина, істини і друзі!
[5] Ліна Костенко
Файл закрито...
```



# Зчитування по рядках

- Прочитати окремий рядок можна за допомогою методу `readline()`
- Якщо досягнуто кінець файлу, метод повертає пустий рядок
- Якщо файл містить пустий рядок, то результатом повертається текст `"\n"`

```
# Відкриваємо текстовий файл для зчитування:
mf=open("D:\\Files\\Python\\poetry.txt")
# Змінна для нумерування рядків:
k=1
# Порядкове зчитування файлу:
print("Порядкове зчитування файлу")
L=mf.readline()
while L!="":
    # Відображення номера рядка:
    print("[ "+str(k)+" ] ",end="")
    # Порядкове відображення символів рядка:
    for s in L:
        # Заміна пробілу на підкреслення:
        if s==' ':
            s='_'
        print(s,end="")
    # Нове значення для номера рядка:
    k+=1
    # Зчитування нового рядка:
    L=mf.readline()
# Закриваємо файл:
mf.close()
print("\nФайл закрито...")
```

Програма (ReadFile\_3.py)

```
Порядкове зчитування файлу
[1] Вже_рік_старий_за_обрії_пливе.
[2] Уже_й_Новий_стоїть_на_виднокрузі.
[3] Хай_у_Новому_буде_все_нове,_-
[4] старі_лиш_вина,_істини_і_друзі!
[5] Ліна_Костенко
Файл закрито...
```



# Режим доступу

## Символ

## Режим доступу

- 'r' Режим зчитування файлу (використовується за замовчуванням)
- 'w' Режим запису у файл (з попередньої очисткою файлу)
- 'x' Режим створення файлу з можливістю запису даних у файл. Якщо файл уже існує, виникає помилка
- 'a' Режим запису у файл. Якщо файл існує, то вміст дописується в кінець файлу
- 'b' Режим доступу до бінарного файлу
- 't' Режим доступу до текстового файлу (використовується за замовчуванням)
- '+' Символ використовується для формування шаблону, який визначає доступ до файлу в режимі оновлення, коли можливе зчитування і запис

## Можна використовувати комбінації. Наприклад:

- "rb" зчитування бінарного файлу
- "r+t" файл відкривається для зчитування і запису, але очистка файлу після відкривання не виконується, запис даних за замовчуванням починається з початку файлу (новий вміст записується "поверх" вже існуючого)
- "r+b" режим доступу до бінарного файлу для зчитування і запису без попередньої очистки вмісту файлу
- "wt" доступ до текстового файлу для запису з попередньою очисткою вмісту файлу
- "w+t" доступ до текстового файлу для зчитування і запису з попередньою чисткою файлу
- "w+b" режим, при якому бінарний файл відкривається для зчитування і запису з попередньою очисткою вмісту файлу
- "a+t" текстовий файл відкривається для зчитування і запису, дані дописуються в кінець файлу



# Запис у файл

## Програма (WriteFile.py)

```
# Зчитуємо текст:  
txt=input("Уведіть текст: ")  
# Файл відкривається для запису:  
mf=open("D:\\Files\\Python\\mytext.txt",'w')  
# Текст записується в файл:  
mf.write(txt)  
# Закривається файл:  
mf.close()  
# Повідомлення про завершення копіювання:  
print("Текст записано у файл")
```

Уведіть текст: **Вивчаємо мову Python**  
Текст записано у файл



# Зчитування і запис

- Метод **seek ()** : - визначає місце у файлі для зчитування або запису значення
  - викликається з об'єкта файлу
  - аргументи: другий - "опорна" позиція, по відношенню до якої визначається місце переходу (початку файлу відповідає значення **0**, поточній позиції у файлі відповідає значення **1**, кінцю файлу відповідає значення **2**), другий - задає зміщення відносно "опорної" позиції
  - для бінарних файлів "одиницею виміру" є байти, для текстових файлів - символи
  - якщо другий аргумент не вказаний, то за замовчуванням зміщення виконується по відношенню до початку файлу
  - для текстових файлів по відношенню до поточної позиції і кінця файлу може виконуватися тільки нульовий зсув
- Метод **tell ()** : поточна позиція в файлі
- Метод **read ()** : цілочисловий аргумент для текстового файлу визначає кількість зчитуваних символів, для бінарного файлу — кількість зчитуваних байтів





# Зчитування і запис

```
# Назва файлу:
name="D:\\Files\\Python\\mydata.txt"
# Текст для запису у файл:
txt="Python"
print("Текст для запису у файл:",txt)
# Файл відкривається для зчитування і запису:
mf=open(name, 'w+t')
# Текст записується в файл:
mf.write(txt)
# Переходимо на початок файлу:
mf.seek(0)
# Перший символ в файлі:
print(mf.tell(), "->",mf.read(1))
# Переходимо в кінець файлу:
mf.seek(0,2)
# Позиція з останнім символом у файлі:
num=mf.tell()-1
# Переходимо на позицію з останнім символом:
mf.seek(num)
# Останній символ у файлі:
print(mf.tell(), "->",mf.read(1))
# Повертаємося на начало файлу:
mf.seek(0)
# Три перших символи:
print("Три символи:",mf.read(3))
# Закривається файл:
mf.close()
print("Програма завершила виконання")
```

## Програма (ReadWriteFile.py)

```
Текст для запису у файл: Python
0 -> P
5 -> n
Три символи: Pyt
Програма завершила виконання
```



# Копіювання файлу

```
print("Починається копіювання файлу")
# Контрольований код:
try:
    # Бінарний файл відкривається для зчитування:
    A=open("D:\\Books\\animal.jpg","rb")
    # Створюється бінарний файл:
    B=open("D:\\Files\\bear.jpg","wb")
    # Вміст першого файлу зчитується
    # і записується у другий файл:
    B.write(A.read())
    # файли закриваються:
    A.close()
    B.close()
    print("Копіювання пройшло успішно")
# Якщо другий файл вже існує:
except FileNotFoundError:
    print("Помилка: такий файл вже існує")
# Всі інші помилки:
except:
    print("Помилка доступу до файлу")
print("Програма завершила виконання")
```

## Програма (CopyFile.py)

```
Починається копіювання файлу
Копіювання пройшло успішно
Програма завершила виконання
```

```
Починається копіювання файлу
Помилка: такий файл вже існує
Програма завершила виконання
```

```
Починається копіювання файлу
Помилка доступу до файлу
Програма завершила виконання
```



# Завдання

**Напишіть програму, в якій користувач вводить назву текстового файлу, а програма відображає вміст цього файлу, а також створює копію цього файлу з перенумерованими рядками**



# Домашнє завдання

**Напишіть програму, в якій створюється текстовий файл. Ім'я файлу вводиться користувачем. Текст для файлу вводиться користувачем. Під час запису тексту в файл всі маленькі букви замінюються на великі**