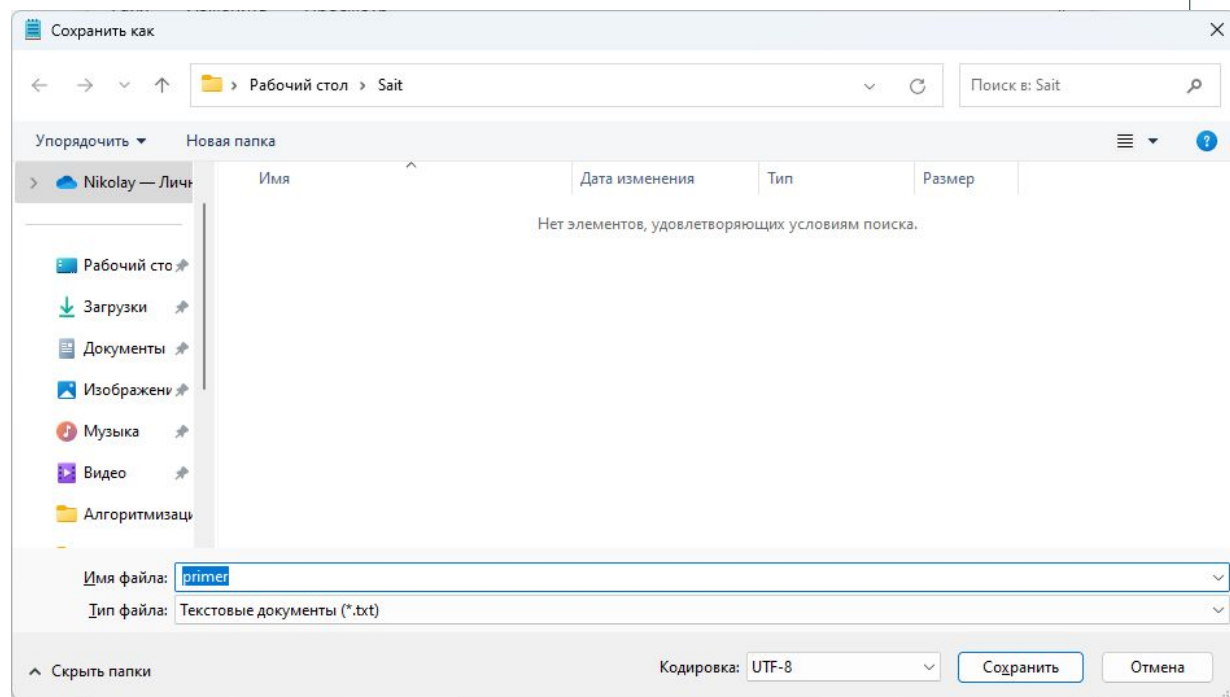
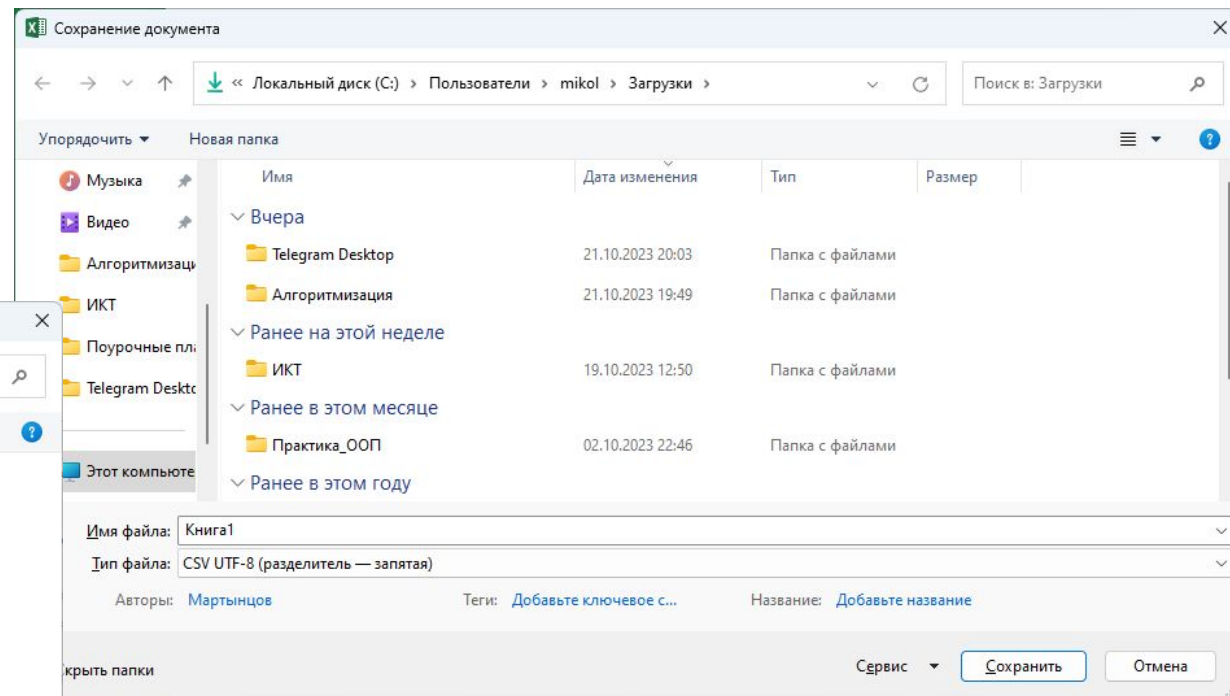


Работа с файлами в Python

Типы файлов



txt – файлы, создаются через Блокнот



CSV – файлы, создаются через Excel

Открытие и закрытие файлов

Чтобы начать работу с файлом, его надо открыть с помощью функции **open()**

Формат:

open(file, mode)

Существует 4 общих режима:

r (Read). Файл открывается для чтения. Если файл не найден, то генерируется исключение `FileNotFoundError`

w (Write). Файл открывается для записи. Если файл отсутствует, то он создается. Если подобный файл уже есть, то он создается заново, и соответственно старые данные в нем стираются.

a (Append). Файл открывается для дозаписи. Если файл отсутствует, то он создается. Если подобный файл уже есть, то данные записываются в его

Пример:

```
myfile = open("hello.txt", "w")
```

```
myfile.close()
```

Конструкция with:

```
with open(file, mode) as file_obj:  
    инструкции
```

Пример:

```
with open('example.txt', 'w') as file:  
    file.write('Hello, World!')
```

Чтение и запись в текстовый файл

Чтение данных из файла:

read(size): читает size байтов из файла. Если size не указан, читает весь файл.

readline(): читает одну строку из файла.

readlines(): читает все строки из файла и возвращает их в виде списка.

```
content = file.read()    # читает весь файл
```

```
line = file.readline()   # читает одну строку
```

```
lines = file.readlines() # читает все строки и  
возвращает список
```

Запись данных в файл:

write(str)

Пример:

```
with open("hello.txt", "w") as file:  
    file.write("hello world")
```

Пример с использованием \n:

```
with open("hello.txt", "a") as file:  
    file.write("\ngood bye, world")
```

метод print():

```
with open("hello.txt", "a") as hello_file:  
    print("Hello, world", file=hello_file)
```

Примеры работы с текстовыми файлами

Пример: Считает количество гласных и согласных букв в тексте из файла

```
def count_vowels_and_consonants(text):
    vowels = "aeiouAEIOU"
    consonants = "bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ"

    num_vowels = sum(1 for char in text if char in vowels)
    num_consonants = sum(1 for char in text if char in consonants)

    return num_vowels, num_consonants

# Читаем текст из файла
with open('example.txt', 'r') as file:
    text = file.read()

# Подсчитываем количество гласных и согласных
num_vowels, num_consonants = count_vowels_and_consonants(text)

print(f"Количество гласных: {num_vowels}")
print(f"Количество согласных: {num_consonants}")
```

Пример: Сгенерирует случайный текст из 500 символов и запишет его в текстовый файл

```
import random
import string

# Генерируем случайный текст
random_text = ''.join(random.choices(string.ascii_letters +
string.digits + string.punctuation + ' ', k=500))

# Записываем его в файл
with open('random_text.txt', 'w') as file:
    file.write(random_text)

print("Случайный текст записан в файл 'random_text.txt'.")
```

Чтение и запись в csv - файл

Чтение данных из файла:

`import csv` – подключаем модуль csv

Чтение данных из файла:

Формат:

`csv.reader(Файловая переменная)`

Пример чтения файла:

```
import csv
```

```
FILENAME = "users.csv"
```

```
with open(FILENAME, "r", newline="") as file:
    reader = csv.reader(file)
    for row in reader:
        print(row[0], " - ", row[1])
```

Запись данных в файл:

writer (ФП) - Создает объект для записи данных в CSV файл.

```
writer = csv.writer(file)
```

writerow(row) - Записывает одну строку в CSV файл.

```
user = ["Sam", 31]
writer.writerow(user)
```

writerows(rows) - Записывает несколько строк в CSV файл.

```
data = [
    ['Имя', 'Возраст', 'Город'],
    ['Анна', '25', 'Москва']
]
writer.writerows(data)
```

Примеры работы с csv - файлом

Пример: Считывает данные из CSV файла, фильтрует и выводит информацию о людях, возраст которых больше 30

```
import csv

# Считываем данные из CSV файла в список
with open('example.csv', 'r', newline='') as file:
    reader = csv.reader(file)
    data = list(reader)

# Выводим информацию о людях, возраст которых больше 30
for row in data:
    if int(row[1]) > 30:
        print(f"{row[0]} {row[1]} лет")
```

Пример: Сгенерирует случайный текст из 500 символов и запишет его в текстовый файл

```
import csv
# Создаем список данных
data = [
    ['Иванов', '1990-05-15'],
    ['Петров', '1985-10-20']
]
# Открываем CSV файл для записи
with open('output.csv', 'w', newline='') as file:
    writer = csv.writer(file)

# Записываем заголовок
writer.writerow(['Фамилия', 'Дата рождения'])

# Проверяем и записываем данные
for record in data:
    if record[1].startswith('1990'):
        writer.writerow(record)
```

Самостоятельная работа текстовые файлы

1. Чтение файла и подсчет слов. Напишите программу, которая читает текстовый файл и подсчитывает количество слов в нем.
2. Копирование файла. Напишите программу, которая копирует содержимое одного текстового файла в другой.
3. Замена слова в файле. Напишите программу, которая читает файл, заменяет все вхождения определенного слова на другое и записывает результат в новый файл.
4. Объединение файлов. Напишите программу, которая объединяет содержимое нескольких текстовых файлов в один.
5. Поиск ключевого слова. Напишите программу, которая содержит текст, по ключевому слову необходимо найти позиции в тексте.
6. Анализ логов. Напишите программу, которая анализирует лог-файл и выводит статистику, например, сколько раз каждый IP-адрес пытался получить доступ к ресурсу.

Самостоятельная работа с csv - файлами

- **Задача: Работа с инвентаризацией.** Создайте CSV файл, представляющий инвентаризацию товаров в магазине. Укажите название товара, количество и стоимость. Напишите программу, которая считает общую стоимость всех товаров.
- **Задача: Анализ данных о студентах.** Создайте CSV файл с данными о студентах (ФИО, возраст, средний балл). Напишите программу, которая определит средний возраст и средний балл студентов.
- **Задача: Фильтрация заказов.** Создайте CSV файл с данными о заказах (название товара, количество, цена). Напишите программу, которая выведет все заказы с суммой выше 1000.
- **Задача: Управление сотрудниками.** Создайте CSV файл с информацией о сотрудниках (ФИО, должность, зарплата). Напишите программу, которая подсчитает общую сумму зарплаты всех сотрудников.
- **Задача: Работа с книгами.** Создайте CSV файл с информацией о книгах (название, автор, год издания). Напишите программу, которая найдет все книги, изданные после 2000 года.
- **Задача: Сравнение продуктов.** Создайте два CSV файла с информацией о продуктах (название, цена) и сравните их, определив, есть ли одинаковые продукты в обоих списках.