



Текстовые файлы

Последнее обновление: 02.02.2024



Запись в текстовый файл

Чтобы открыть текстовый файл на запись, необходимо применить режим **w** (перезапись), **a** (дозапись) или **r+ / w+** (чтение и запись). Затем для записи применяется метод **write(str)**, в который передается записываемая строка. Стоит отметить, что записывается именно строка, поэтому, если нужно записать числа, данные других типов, то их предварительно нужно конвертировать в строку.

Запишем некоторую информацию в файл "hello.txt":

```
1 with open("hello.txt", "w") as file:
2     file.write("hello world")
3
4 print("Файл записан")
```

Если мы откроем папку, в которой находится текущий скрипт Python, то увидим там файл "hello.txt". Этот файл можно открыть в любом текстовом редакторе и при желании изменить.

Теперь дозапишем в этот файл еще одну строку:

```
1 with open("hello.txt", "a") as file:
```

```
2     file.write("\nhello work")
3
4     print("Файл изменен")
```

Дозапись выглядит как добавление строки к последнему символу в файле, поэтому, если необходимо сделать запись с новой строки, то можно использовать эскейп-последовательность `"\n"`. В итоге файл `hello.txt` будет иметь следующее содержимое:

```
1 hello world
2 hello work
```

Для записи списка строк имеется метод **writelines()**. В качестве параметра он принимает список строк:

```
1 lines = ["Hello Word\n", "Hello Work\n", "Hello World\n"]
2 with open("hello2.txt", "a") as file:
3     file.writelines(lines)
4
5     print("Список строк записан в файл")
```

Стоит учитывать, что, если мы хотим, чтобы каждая строка из списка располагалась на отдельной строке в файле, то строка в списке должна заканчиваться на символ `"\n"`.

Еще один способ записи в файл представляет стандартный метод **print()**, который применяется для вывода данных на консоль:

```
1 with open("hello.txt", "a") as myfile:
2     print("\nhello metanit.com", file=myfile)
```

Для вывода данных в файл в метод `print` в качестве второго параметра передается название файла через параметр `file`. А первый параметр представляет записываемую в файл строку.

Чтение файла

Для чтения файла он открывается с режимом **r** (Read), и затем мы можем считать его содержимое различными методами:

- **readline()**: считывает одну строку из файла
- **read()**: считывает все содержимое файла в одну строку
- **readlines()**: считывает все строки файла в список

Например, считаем выше записанный файл построчно:

```
1 with open("hello.txt", "r") as file:
2     for line in file:
3         print(line)
```

Несмотря на то, что мы явно не применяем метод `readline()` для чтения каждой строки, но в при переборе файла этот метод автоматически вызывается для получения каждой новой строки. Поэтому в цикле вручную нет смысла вызывать метод `readline`. Консольный вывод:

```
hello world

hello work

hello metanit.com
```

Однако поскольку функция `print` неявно добавляет к каждой строке перевод строки `"\n"`, а строки в записанном файлетакже разделяются символом перевода строки `"\n"`, то получается, что каждая строка оканчивается двойным `"\n"`. и чтобы исключить излишнего переноса на другую строку в функцию `print` можно передать значение `end=""`.

```
1 with open("hello.txt", "r") as file:
2     for line in file:
3         print(line, end="")
```

Теперь явным образом вызовем метод `readline()` для чтения отдельных строк:

```
1 with open("hello.txt", "r") as file:
2     str1 = file.readline()      # считываем первую строку
3     print(str1, end="")
4     str2 = file.readline()      # считываем вторую строку
5     print(str2)
```

Консольный вывод:

```
hello world
hello work
```

Метод `readline` можно использовать для построчного считывания файла в цикле `while`:

```
1 with open("hello.txt", "r") as file:
2     line = file.readline()
3     while line:
4         print(line, end="")
5         line = file.readline()
```

Если файл небольшой, то его можно разом считать с помощью метода `read()`:

```
1 with open("hello.txt", "r") as file:
2     content = file.read()
3     print(content)
```

И также применим метод `readlines()` для считывания всего файла в список строк:

```
1 with open("hello.txt", "r") as file:
2     contents = file.readlines()
3     str1 = contents[0]
4     str2 = contents[1]
5     print(str1, end="")
6     print(str2)
```

При чтении файла мы можем столкнуться с тем, что его кодировка не совпадает с ASCII. В этом случае мы явным образом можем указать кодировку с помощью параметра `encoding`:

```
1 filename = "hello.txt"
2 with open(filename, encoding="utf8") as file:
3     text = file.read()
```

Чтение и запись

Для открытия файла сразу для чтения и записи применяются флаги:

- **r+**: если файл не найден, то генерируется исключение `FileNotFoundError`
- **w+**: если файл не существует, то он автоматически создается. Если файл существует, то он перезаписывается

Например, запишем и считаем из файла данные:

```
1 with open("hello.txt", "w+") as file:
2     file.write("Hello world\nHello work\n") # сначала записываем данные
3     file.seek(0) # перемещаемся к первому байту в файле
4     content = file.read() # считываем данные
5     print(content)
```

Перемещение по файлу

Для перемещения по файлу после его открытия применяется метод **seek()**. В него передается индекс байта, к которому надо переместить указатель в файле. Например, в примере выше после записи указатель в файле автоматически устанавливается на конец файла. И считывания данных идет с позиции, на которой стоит указатель. И чтобы его сдвинуть в самое начало, применяется вызов

```
1 file.seek(0)
```

Аналогичным образом мы можем перемещаться по файлу на нужную позицию. Например, считаем данные, начиная с 5-го символа:

```
1 with open("hello.txt", "w+") as file:
2     file.write("Hello world\n") # сначала записываем данные
3     file.seek(6)                # перемещаемся к шестому байту в файле
4     content = file.read()       # считываем данные
5     print(content)             # world
```

Пример работы с файлами

Теперь напишем небольшой скрипт, в котором будет записывать введенный пользователем массив строк и считывать его обратно из файла на консоль:

```
1 # имя файла
2 FILENAME = "messages.txt"
3
4 # запись строки в файл
5 def write():
6     message = input("Введите строку: ")
7     with open(FILENAME, "a") as file:
8         file.write(message + "\n")
9
10 # чтение файла файл
11 def read():
12     with open(FILENAME, "r") as file:
13         for message in file:
14             print(message, end="")
15     print() # перевод строки для разделения меню и вывода
16
17 while(True):
18     selection = int(input("1.Запись в файл\t\t2.Чтение файла\t\t3.Выход\nВыберите д
19     match selection:
20         case 1: write()
21         case 2: read()
22         case 3: break
23         case _: print("Некорректный ввод")
24
25 print("Программа завершена")
```

Пример работы программы:

```
1.Запись в файл          2.Чтение файла          3.Выход
Выберите действие: 1
Введите строку: Hello World

1.Запись в файл          2.Чтение файла          3.Выход
Выберите действие: 1
Введите строку: Hello METANIT.COM

1.Запись в файл          2.Чтение файла          3.Выход
Выберите действие: 2
Hello World
Hello METANIT.COM

1.Запись в файл          2.Чтение файла          3.Выход
Выберите действие: 3
Программа завершена
```

[Назад](#) [Содержание](#) [Вперед](#)



Помощь сайту

[Помощь сайту](#)

Юмани:

410011174743222

Номер карты:

4048415020898850

[Телеграмм](#)

[Вконтакте](#) | [Телеграм](#) | [Донаты/Помощь сайту](#).

Contacts: metanit22@mail.ru

Copyright © Евгений Попов, metanit.com, 2025. Все права защищены.