

[ХОЧУ ПОМОЧЬ
ПРОЕКТУ](#)

Работа с файлами в Python



EON eon-energydrink.com РЕКЛАМА

Больше информации на сайте
рекламодателя

[Узнать больше](#)[Справочник по языку Python3.](#) / Работа с файлами в Python

Чтение, запись и обработка файлов в Python

В разделе подробно разобраны [основные методы работы с файлами](#). Чтение данных и запись данных в файл, управление указателем в файле при чтении/записи и многое другое.

Дополнительно смотрите:

- встроенную [функцию open\(\)](#);
- встроенный тип - [файловый объект](#);

Для удобной работы с путями файловой системы, рекомендуем использовать встроенный [модуль pathlib](#).

[Составление пути к файлу в Unix и Windows в Python](#)

Различия в оставления пути к файлу в Unix и Windows. При доступе к файлу в операционной системе требуется правильно указать путь к файлу.

[Открытие/закрытие файла для чтения/записи в Python](#)

Прежде чем начать работать с файлом, первое, что нужно сделать, это открыть его. Это делается путем вызова встроенной функции `open()`. Она имеет единственный обязательный аргумент, который представляет путь к файлу `filename`

[Типы обрабатываемых данных и файлов в Python](#)

Существуют три типа файлов которые чаще всего обрабатываются на практике. Текстовые файлы. Буферизованные двоичные типы файлов. Необработанный тип файлов Raw.

[Способы чтения открытого файла в Python](#)

Существует несколько методов, которые могут быть вызваны для чтения открытого файла. `read(size=-1)`, `readline(size=-1)`, `readlines()`

[Способы записи в открытый файл в Python](#)

Как и при чтении файлов, файловые объекты имеют несколько методов, которые полезны для записи в файл. `fp.write(string)`, `fp.writelines(sequence)`

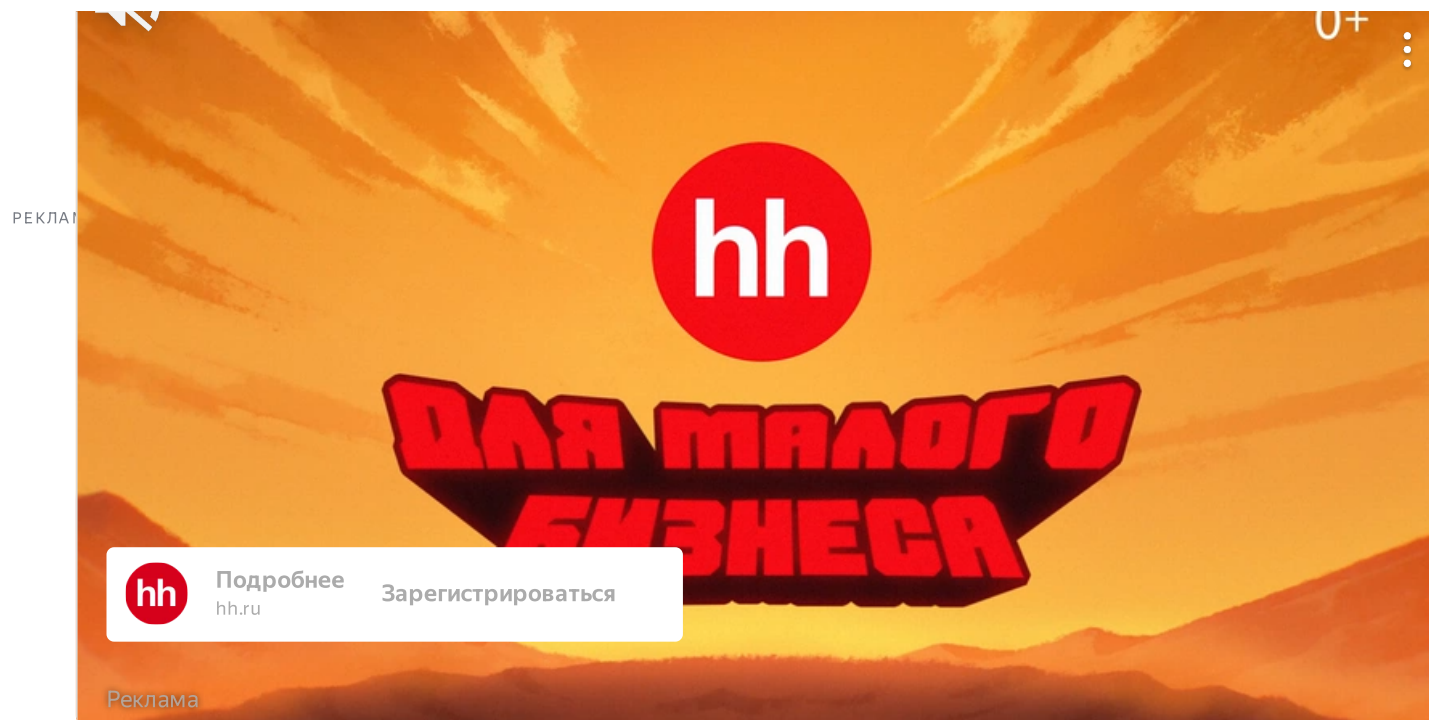
[Одновременное чтение из одного и запись в другой файл в Python](#)

Есть моменты, когда необходимо читать файл, на лету обрабатывать и сразу писать данные в другой файл или например объединить нескольких фалов из каталога в один общий файл.

[Добавление данных в открытый файл в Python](#)

Иногда может понадобиться добавить данные в файл или начать запись в конце уже заполненного файла. Это легко сделать, используя символ 'a' для аргумента `mode` функции `open()`:

[Вверх](#)



[Управление указателем чтения/записи в файле в Python](#)

Краткий обзор методов управления указателем чтения/записи в файле.

[Создание менеджера для обработки файла в Python](#)

У менеджера контекста есть два "магических метода". `__enter__()` - вызывается при вызове оператора `with`. `__exit__()` вызывается при выходе из блока оператора `with`.

[Сохранение словарей в формат JSON в Python](#)

Для сохранения сложных типов данных в файлы, Python позволяет использовать популярный формат обмена данными, называемый JSON. `pickle` - это протокол, который позволяет сериализовать произвольно сложные объекты Python.

[Встроенные модули для работы с разными форматами в Python](#)

Есть общие ситуации, с которыми вы можете столкнуться при работе с файлами. Большинство из этих случаев могут быть обработаны с помощью других модулей и библиотек.