



Работа с файлами

Открытие и закрытие файлов

Последнее обновление: 02.02.2024



Python поддерживает множество различных типов файлов, но условно их можно разделить на два вида: текстовые и бинарные. Текстовые файлы - это к примеру файлы с расширением cvs, txt, html, в общем любые файлы, которые сохраняют информацию в текстовом виде. Бинарные файлы - это изображения, аудио и видеофайлы и т.д. В зависимости от типа файла работа с ним может немного отличаться.

При работе с файлами необходимо соблюдать некоторую последовательность операций:

1. Открытие файла с помощью метода **open()**
2. Чтение файла с помощью метода **read()** или запись в файл посредством метода **write()**
3. Закрытие файла методом **close()**

Открытие и закрытие файла

Чтобы начать работу с файлом, его надо открыть с помощью функции **open()**, которая имеет следующее формальное определение:

```
1 | open(file, mode)
```

Первый параметр функции представляет путь к файлу. Путь файла может быть абсолютным, то есть начинаться с буквы диска, например, *C://somedir/somefile.txt*. Либо можно быть относительным, например, *somedir/somefile.txt* - в этом случае поиск файла будет идти относительно расположения запущенного скрипта Python.

Второй передаваемый аргумент - `mode` устанавливает режим открытия файла в зависимости от того, что мы собираемся с ним делать. 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 конец.
- **b** (Binary). Используется для работы с бинарными файлами. Применяется вместе с другими режимами - `w` или `r`, например, **rb** (чтение бинарных файлов) и **wb** (запись бинарных файлов).
- **r+**. Файл открывается одновременно для чтения и записи. Если файл не найден, то генерируется исключение `FileNotFoundError`
- **w+**. Файл открывается одновременно для чтения и записи. Если файл не существует, то он автоматически создается. Если файл существует, то он перезаписывается
- **a+**. Файл открывается одновременно для чтения и записи. Если файл не существует, то он автоматически создается. Если файл существует, то данные добавляются в конец файла

Например, открытие текстового файла на запись

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

Или открытие бинарного файла на чтение

```
1 myfile = open("image.png", "rb")
```

После завершения работы с файлом его обязательно нужно закрыть методом `close()`. Данный метод освободит все связанные с файлом используемые ресурсы.

Например, откроем для записи текстовый файл "hello.txt":

```
1 myfile = open("hello.txt", "w")
2
3 myfile.close()
```

При открытии файла или в процессе работы с ним мы можем столкнуться с различными исключениями, например, к нему нет доступа и т.д. В этом случае программа выпадет в ошибку, а ее выполнение не дойдет до вызова метода `close`, и соответственно файл не будет закрыт.

В этом случае мы можем обрабатывать исключения:

```
1 try:
2     myfile = open("hello.txt", "w")
3     try:
4         print("Работа с файлом")
5     finally:
6         myfile.close()
7 except Exception as ex:
8     print(ex)
```

В данном случае вся работа с файлом идет во вложенном блоке try. И если вдруг возникнет какое-либо исключение, то в любом случае в блоке finally файл будет закрыт.

Однако есть и более удобная конструкция - конструкция **with**:

```
1 with open(file, mode) as myfile:
2     инструкции
```

Эта конструкция определяет для открытого файла переменную myfile и выполняет набор инструкций. После их выполнения файл автоматически закрывается. Даже если при выполнении инструкций в блоке with возникнут какие-либо исключения, то файл все равно закрывается.

Так, перепишем предыдущий пример:

```
1 with open("hello.txt", "w") as myfile:
2     print("Работа с файлом myfile")
```

[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.
