

# Python File Open

Python имеет несколько функций для создания, чтения, обновления и удаления файлов.

Ключевой функцией для работы с файлами в Python является `open()` функция.

Функция `open()` принимает два параметра; имя файла и режим .

Существует четыре различных метода (режима) открытия файла:

- **"r"**- Чтение - Значение по умолчанию. Открывает файл для чтения, ошибка, если файл не существует
- **"a"**- Append - открывает файл для добавления, создает файл, если он не существует
- **"w"**- Write - открывает файл для записи, создает файл, если он не существует
- **"x"**- Create - Создает указанный файл, возвращает ошибку, если файл существует

Кроме того, вы можете указать, следует ли обрабатывать файл в двоичном или текстовом режиме.

- **"t"**- Текст - значение по умолчанию. Текстовый режим
- **"b"**- Двоичный - Двоичный режим (например, изображения)

## Синтаксис

Чтобы открыть файл для чтения достаточно указать имя файла:

```
f = open("demofile.txt")
```

Приведенный выше код аналогичен:

```
f = open("demofile.txt", "rt")
```

Поскольку `"r"` для чтения и `"t"` для текста используются значения по умолчанию, указывать их не нужно!

**Убедитесь, что файл существует, иначе вы получите сообщение об ошибке.**

## Open a File on the Server

Предположим, у нас есть следующий файл, расположенный в той же папке, что и Python:

демо-файл.txt

Hello! Welcome to demofile.txt

This file is for testing purposes.

Good Luck!

Чтобы открыть файл, используйте встроенную `open()` функцию.

Функция `open()` возвращает файловый объект, у которого есть `read()` метод для чтения содержимого файла:

```
f = open("demofile.txt", "r")
print(f.read())
```

Если файл находится в другом месте, вам нужно будет указать путь к файлу, например:

```
f = open("D:\\myfiles\\welcome.txt", "r")
print(f.read())
```

## Read Only Parts of the File

By default the `read()` method returns the whole text, but you can also specify how many characters you want to return:

Return the 5 first characters of the file:

```
f = open("demofile.txt", "r")
print(f.read(5))
```

## Read Lines

You can return one line by using the `readline()` method:

Read one line of the file:

```
f = open("demofile.txt", "r")
print(f.readline())
```

By calling `readline()` two times, you can read the two first lines:

```
f = open("demofile.txt", "r")
print(f.readline())
print(f.readline())
```

By looping through the lines of the file, you can read the whole file, line by line:

Перебирая строки файла, вы можете прочитать весь файл построчно:

```
f = open("demofile.txt", "r")
for x in f:
    print(x)
```

## Close Files

It is a good practice to always close the file when you are done with it.

```
f = open("demofile.txt", "r")
print(f.readline())
f.close()
```

---

# Python File Write

## Write to an Existing File

Для записи в существующий файл необходимо добавить в `open()` функцию параметр:

- **"a"**- Append - добавит в конец файла.
- **"w"**- Write - перезапишет любой существующий контент.

Откройте файл «demofile2.txt» и добавьте в него содержимое:

```
f = open("demofile2.txt", "a")
f.write("Now the file has more content!")
f.close()
#open and read the file after the appending:
f = open("demofile2.txt", "r")
print(f.read())
```

Откройте файл «demofile3.txt» и перезапишите содержимое:

```
f = open("demofile3.txt", "w")
f.write("Woops! I have deleted the content!")
f.close()
#open and read the file after the appending:
f = open("demofile3.txt", "r")
print(f.read())
```

Output:

Woops! I have deleted the content!

Метод «**w**» перезапишет весь файл.

## Create a New File

Чтобы создать новый файл в Python, используйте open() метод с одним из следующих параметров:

- **"x"**- Create - создаст файл, вернет ошибку, если файл существует.
- **"a"**- Добавить - создаст файл, если указанный файл не существует.
- **"w"**- Write - создаст файл, если указанный файл не существует.

Создайте файл с именем «myfile.txt»:

```
f = open("myfile.txt", "x")
```

Создайте новый файл, если он не существует:

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

## Python Delete File

### Delete a File

To delete a file, you must import the OS module, and run its `os.remove()` function:

Remove the file "demofile.txt":

```
import os
os.remove("demofile.txt")
```

## Check if File exist:

To avoid getting an error, you might want to check if the file exists before you try to delete it:

```
import os
if os.path.exists("demofile.txt"):
    os.remove("demofile.txt")
else:
    print("The file does not exist")
```

## Delete Folder

To delete an entire folder, use the `os.rmdir()` method:

Remove the folder "myfolder":

```
import os
os.rmdir("myfolder")
```

**You can only remove empty folders.**