

# Работа с файлами



## Открытие

Напишем простой код:

```
F = open("text.txt", "w")
print("Имя файла: ", F.name)
print("Файл закрыт (True или False): ", F.closed)
print("В каком режиме файл открыт: ", F.mode)
```

Пояснение:

F – дескриптор, содержит  
информацию про:

- Имя файла: F.name;
- Режим открытия: F.mode;
- Открыт/Закрыт: F.closed

(если закрыт, возвращает True  
False, если открыт)

# Работа с файлами



## Режимы и типы

Файл открывается для:

**ЧТЕНИЯ** или **ЗАПИСИ** (РЕЖИМ - MODE)

Файлы бывают двух **ТИПОВ**:

**ТЕКСТОВЫЕ** или **БИНАРНЫЕ** (двоичные)

## Этапы работы

Последовательность работы с файлом такая:

1. **ОТКРЫТИЕ** (`open(...)`) – файл открывается в определённом режиме (для **чтения** или **записи**);
2. **ОПЕРАЦИИ** (`read(...)`, `write(...)` и т.д.)
3. **ЗАКРЫТИЕ** (`close ()`)

# Работа с файлами



## Режимы (способы) работы

Режимы (в основном) определяются по начальным буквам английских слов:

РЕЖИМ (MODE)	СЛОВО	ОПИСАНИЕ
"r"	read	Открывает файл только для чтения. Если файл не существует, выводится сообщение об ошибке
"w"	write	Открывает файл только для записи. Если файл не существует, то он создается.
"a"	append	Открывает файл для дозаписи, добавления информации.
"t"	text	Текстовый файл
"b"	binary	Двоичный файл. Например, картинка (изображение).
"x"	XOR исключение	Файл открывается для записи, но если файл уже существует, выводится сообщение об ошибке

Возможны сочетания режимов. Например: "wt" – текстовый файл для записи. Если режим не указан, то открывается текстовый файл для чтения

# Работа с файлами



## Запись

Запишем в наш файл небольшой текст. Для этого немного перепишем код:

```
F = open("text.txt", "w")
f.write("Мне нравится программировать на Python")
print("Имя файла: ", F.name)
print("В каком режиме файл открыт: ", F.mode)
print("Файл закрыт? (True или False): ", F.closed)
F.close()
print("А теперь закрыт? (True или False): ", F.closed)
```

В результате работы программы в файле `text.txt` появится наш введенный текст. **Python** автоматически закрывает файл по завершении работы программы. Однако, хорошей практикой будет вручную закрывать файл командой.

Как мы уже успели заметить, это делается командой `close()`.

# Работа с файлами



## Чтение

Теперь прочитаем текст из нашего файла. Для этого еще раз немного перепишем код:

```
F = open("text.txt")
data = f.read (3)
print("Имя файла: ", F.name)
print("Из него прочитано первые 3 символа: ", data)
print("В каком режиме файл открыт: ", F.mode)
print("Файл закрыт? (True или False): ", F.closed)
F.close()
print("А теперь закрыт? (True или False): ", F.closed)
```

Можно прочитать и весь текст. Для этого вторую и четвертую строки из кода выше перепишем так (остальное без изменений):

```
data = f.read ()
print("Из него прочитано: ", data)
```

# Работа с файлами



## Построчное чтение

Теперь вручную добавим в наш файл вторую строку: «Потому что это мощный язык.». И читаем каждую строку отдельно. Иногда такой способ может быть полезным:

```
F = open("text.txt")
line1 = f.readline ()
line2 = f.readline()
print("Имя файла: ", F.name)
print("Из него прочитано:")
print("Строка №1: ", line1)
print("Строка №2: ", line1)
print("Имя файла: ", F.name)
print("В каком режиме файл открыт: ", F.mode)
print("Файл закрыт? (True или False): ", F.closed)
F.close()
print("А теперь закрыт? (True или False): ", F.closed)
```



# Работа с файлами



## Удаление

Файл можно удалить программным кодом. Для этого нужно вызвать модуль (объект) **os**:

```
import os
```

**os** (Operation System) – означает «Операционная система». Методами модуля **os** можно:

- Определить имя системы:

```
Sys = os.uname  
print("Имя системы", Sys.sysname())
```

- Получить список файлов в текущей папке:

```
os.listdir()
```

- Удалить наш файл `text.txt`:

```
os.remove("text.txt")
```

# Работа с файлами



С помощью `os` можно проверить существование файла:

```
import os

if os.path.exists("text.txt"):
    print("Файл с именем 'text.txt' существует:")
else:
    print("Файл с именем 'text.txt' не существует:)
```

Это бывает полезно в том случае, когда необходимо, например, прочитать информацию из файла, т.е. перед тем как начать операции с ним.

## Контрольное задание

Создать текстовый файл `text.txt`.  
Записать в него текст, введенный с клавиатуры. После прочитать этот текст из файла и вывести на экран

**заглавными буквами, используя функцию работы со строками `upper()`**