

Программирование, лекция 12. Файлы. Текстовые файлы

Кафедра ИУ7 МГТУ им. Н. Э. Баумана,
2022 год

Модуль `сору`

2 метода:

`сору(х)` - создаёт “мелкую” копию объекта

`деерсору(х, [мето])` - создаёт глубокую копию

Файлы

Файл - поименованное место на носителе данных (внешняя память).

В языках программирования обычно применяется концепция, в которой файл является абстракцией, не привязанной к конкретному типу носителя и файловой системе, а работа с файлами осуществляется подобно обработке массива данных.

Однако, в зависимости от технических возможностей, ряд файлов поддерживает **произвольный** доступ, а остальные - только **последовательный**.

Виды файлов

- текстовые файлы
- структурированные (типизированные) форматы
- бинарные файлы

Формат файла определяется его содержимым. Расширение файла обычно соответствует формату файла, но в общем случае никак на него не влияет.

Дескриптор

Файловый дескриптор — целое число, которое присваивается операционной системой каждому потоку ввода-вывода при его создании.

Открытие файла

```
open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None,
closefd=True, opener=None)
```

- `file` - путь к файлу (полный или относительный) или *иной адрес, подобный пути*
- `mode` - режим доступа к файлу
- `buffering` - метод буферизации
- `encoding` - кодировка
- `errors` - реакция на ошибки декодирования файла
- `newline` - формат перевода на следующую строку для текстовых файлов
- `closefd` - обработка файлового дескриптора, если он указан
- `opener` - функция открытия файла

Режимы доступа к файлам

Character - Meaning

'r' open for reading (default)

'w' open for writing, truncating the file first

'x' open for exclusive creation, failing if the file already exists

'a' open for writing, appending to the end of file if it exists

'b' binary mode

't' text mode (default)

'+' open for updating (reading and writing)

Свойства и методы файловых объектов

f.buffer	f.flush()	f.readable()	f.truncate()
f.close()	f.isatty()	f.readline()	f.writable()
f.closed	f.line_buffering	f.readlines()	f.write()
f.detach()	f.mode	f.reconfigure()	f.write_through
f.encoding	f.name	f.seek()	f.writelines()
f.errors	f.newlines	f.seekable()	
f.fileno()	f.read()	f.tell()	

Методы текстовых файловых объектов

`f.close`

`f.read()`

`f.readline()`

`f.readlines()`

`f.write()`

`f.writelines(lines)`

`f.truncate(size)`

`f.seek(offset)`

`f.tell()`

Способы чтения текстовых файлов

```
f = open('in.txt')  
content = f.read() # 1  
lines_list = f.readlines() # 2  
for i in (range(5)): # 3  
    string = f.readline()  
for s in f: # 4  
    print(s)  
f.close()
```

Способы записи текстовых файлов

```
f = open('in.txt', 'w')
```

```
f.write('abcd') # 1
```

```
f.writelines(['1','abcd']) # 2
```

```
print('def', file=f)
```

```
f.close()
```

Перевод на новую строку - спец. символ!

Исключения при работе с файлами

Ошибки возможны:

- при открытии файла
- при записи
- и вообще при любых операциях

Оператор with

with open(...) as f1, open(...) as f2:

операторы