



Алексей Васильев

**Язык
программирования
Python**

Киев 2019



Лекция 8. Работа с файлами



- Считывание содержимого файла
- Запись данных в файл
- Построчное считывание файла
- Режимы доступа к файлам
- Копирование файлов



Чтение файла

- **Открытие файла:** функция `open()`
- **Аргументы функции `open()`:** `[1]` полный путь к файлу (обязательно)
`[2]` инструкция, определяющая режим доступа (не обязательно)
- **Файловый объект:** методы `write()` (запись в файл) и `read()` (чтение из файла)
- **Метод `close()`:** закрытие файла

```
# Открываем текстовый файл для чтения:  
mf=open("D:\\Files\\poetry.txt")  
# Считывается содержимое файла:  
txt=mf.read()  
print("Содержимое файла:")  
# Отображение содержимого файла:  
print(txt)  
# Закрываем файл:  
mf.close()  
print("Файл закрыт...")
```

Программа (ReadFile.py)

Содержимое файла:

```
Чтоб мудро жизнь прожить, знать надобно немало.  
Два важных правила запомни для начала.  
Ты лучше голодай, чем что попало есть,  
И лучше будь один, чем вместе с кем попало.  
Омар Хайям  
Файл закрыт...
```



Построчное чтение

```
# Открываем текстовый файл для чтения:
mf=open("D:\\Files\\poetry.txt")
# Переменная для нумерации строк:
k=1
# Построчное считывание файла:
print("Построчное считывание файла")
for L in mf:
    # Отображение номера строки и самой строки:
    print("[ "+str(k)+" ] ",L,end="")
    # Новое значение для номера строки:
    k+=1
# Закрываем файл:
mf.close()
print("\nФайл закрыт...")
```

Программа (ReadFile_2.py)

Построчное считывание файла

```
[1] Чтоб мудро жизнь прожить, знать надобно
    немало.
[2] Два важных правила запомни для начала.
[3] Ты лучше голодай, чем что попало есть,
[4] И лучше будь один, чем вместе с кем попало.
[5] Омар Хайям
Файл закрыт...
```



Построчное чтение - 2

- Прочитать отдельную строку можно с помощью метода `readline()`
- Если достигнут конец файла, метод возвращается пустая строка
- Если файл содержит пустую строку, то результатом возвращается текст `"\n"`

```
# Открываем текстовый файл для чтения:
mf=open("D:\\Files\\poetry.txt")
# Переменная для нумерации строк:
k=1
# Построчное считывание файла:
print("Построчное считывание файла")
L=mf.readline()
while L!="":
    # Отображение номера строки:
    print("[ "+str(k)+" ] ",end="")
    # Построчное отображение символов строки:
    for s in L:
        # Замена пробела на подчеркивание:
        if s==' ':
            s='_'
        print(s,end="")
    # Новое значение для номера строки:
    k+=1
    # Считывание новой строки:
    L=mf.readline()
# Закрываем файл:
mf.close()
print("\nФайл закрыт...")
```

Программа (ReadFile_3.py)

Построчное считывание файла

```
[1] Чтоб_мудро_жизнь_прожить,_знать_надобно_немало.
[2] Два_важных_правила_запомни_для_начала.
[3] Ты_лучше_голодай,_чем_что_попало_есть,
[4] И_лучше_будь_один,_чем_вместе_с_кем_попало.
[5] Омар_Хайям
Файл закрыт...
```



Режим доступа

Символ

Режим доступа

- 'r' Режим чтения файла (используется по умолчанию)
- 'w' Режим записи в файл (с предварительной очисткой файла)
- 'x' Режим создания файла с возможность записи данных в файл. Если файл уже существует, возникает ошибка
- 'a' Режим записи в файл. Если файл существует, то содержимое дописывается в конец файла
- 'b' Режим доступа к бинарному файлу
- 't' Режим доступа к текстовому файлу (используется по умолчанию)
- '+' Символ используется для формирования шаблона, определяющего доступ к файлу в режиме обновления, когда возможно чтение и запись

Можно использовать комбинации. Например:

- "rb" чтение бинарного файла
- "r+t" файл открывается для чтения и записи, но очистка файла после открытия не выполняется, запись данных по умолчанию начинается с начала файла (новое содержимое записывается "поверх" уже существующего)
- "r+b" режим доступа к бинарному файлу для чтения и записи без предварительной очистки содержимого файла
- "wt" доступ к текстовому файлу для записи с предварительной очисткой содержимого файла
- "w+t" доступ к текстовому файлу для чтения и записи с предварительной очисткой файла
- 'w+b' режим, при котором бинарный файл открывается для чтения и записи с предварительной очисткой содержимого файла
- "a+t" текстовый файл открывается для чтения и записи, данные дописываются в конец файла



Запись в файл

Программа (WriteFile.py)

```
# Считывается текст:  
txt=input("Введите текст: ")  
# файл открывается для записи:  
mf=open("D:\\Files\\mytext.txt", 'w')  
# Текст записывается в файл:  
mf.write(txt)  
# Закрывается файл:  
mf.close()  
# Сообщение о завершении копирования:  
print("Текст записан в файл")
```

Введите текст: **Изучаем язык Python**
Текст записан в файл



Чтение и запись

- Метод **seek ()** : - определяет место в файле для считывания или записи значения
 - вызывается из объекта файла
 - аргументы: второй - "опорная" позиция, по отношению к которой определяется место перехода (началу файла соответствует значение **0**, текущей позиции в файле соответствует значение **1**, концу файла соответствует значение **2**), второй - задает смещение относительно "опорной" позиции
 - для бинарных файлов "единицей измерения" являются байты, для текстовых файлов - символы
 - если второй аргумент не указан, то по умолчанию смещение выполняется по отношению к началу файла; для текстовых файлов по отношению к текущей позиции и концу файла может выполняться только нулевой сдвиг
- Метод **tell ()** : текущая позиция в файле
- Метод **read ()** : целочисленный аргумент для текстового файла определяет количество считываемых символов, для бинарного файла — количество считываемых байтов



Чтение и запись

```
# Имя файла:
name="D:\\Files\\mydata.txt"
# Текст для записи в файл:
txt="Python"
print("Текст для записи в файл:",txt)
# файл открывается для чтения и записи:
mf=open(name, 'w+t')
# Текст записывается в файл:
mf.write(txt)
# Переходим в начало файла:
mf.seek(0)
# Первый символ в файле:
print(mf.tell(), "->",mf.read(1))
# Переходим в конец файла:
mf.seek(0,2)
# Позиция с последним символом в файле:
num=mf.tell()-1
# Переходим на позицию с последним символом:
mf.seek(num)
# Последний символ в файле:
print(mf.tell(), "->",mf.read(1))
# Возвращаемся в начало файла:
mf.seek(0)
# Три первых символа:
print("Три символа:",mf.read(3))
# Закрывается файл:
mf.close()
print("Программа завершила выполнение")
```

Программа (ReadWriteFile.py)

```
Текст для записи в файл: Python
0 -> P
5 -> n
Три символа: Pyt
Программа завершила выполнение
```



Копирование файла

```
print("Начинается копирование файла")
# Контролируемый код:
try:
    # Бинарный файл открывается для чтения:
    A=open("D:\\Books\\animal.jpg","rb")
    # Создается бинарный файл:
    B=open("D:\\Files\\bear.jpg","wb")
    # Содержимое первого файла считывается
    # и записывается во второй файл:
    B.write(A.read())
    # Файлы закрываются:
    A.close()
    B.close()
    print("Копирование прошло успешно")
# Если второй файл уже существует:
except FileNotFoundError:
    print("Ошибка: такой файл уже существует")
# Все прочие ошибки:
except:
    print("Ошибка доступа к файлу")
print("Программа завершила выполнение")
```

Программа (CopyFile.py)

Начинается копирование файла
Копирование прошло успешно
Программа завершила выполнение

Начинается копирование файла
Ошибка: такой файл уже существует
Программа завершила выполнение

Начинается копирование файла
Ошибка доступа к файлу
Программа завершила выполнение



Задание

Напишите программу, в которой пользователь вводит имя текстового файла, а программа отображает содержимое этого файла, а также создает копию этого файла с пронумерованными строками.



Домашнее задание

Напишите программу, в которой создается текстовый файл. Имя файла вводится пользователем. Текст для файла вводится пользователем. При записи текста в файл все маленькие буквы заменяются на большие