

## Файлы

**Файл** (англ. file) — именованная область данных на носителе информации

(Материал из Википедии — свободной энциклопедии)

Имя файла = собственно имя + расширение

Путь к файлу - абсолютный и относительный.



## Открытие и закрытие файла в Python

- Перед тем как прочесть и записать что-либо в файл, его следует открыть. Чтобы открыть файл в Python, используется встроенная функция `open()`
- При вызове эта функция создает объект типа файл, с которым в дальнейшем можно работать:

```
my_file = open (file_name, mode=режим_доступа)
```

- Примеры

```
f = open ('1.txt', 'r')      // ok
f = open ('c:\1.txt', 'r')  // error !!!
f = open (r'c:\1.txt', 'r') // ok
f = open ('c:\\1.txt', 'r') // ok
```

- Метод файлового объекта `close()` автоматически закрывает файл, при этом теряется любая несохраненная информация. Работать с файлом (читать, записывать) после этого нельзя.

```
f.close()
```

## Основные режимы доступа

Режим	Значение
'r'	Открытие на чтение (является значением по умолчанию)
'w'	Открытие на запись. Содержимое файла удаляется. Если файла не существует, создается новый.
'a'	Открытие на добавление информации. Информация добавляется в конец файла. Если файла не существует, создается новый.
'b'	Открытие в двоичном режиме.
't'	Открытие в текстовом режиме (является значением по умолчанию).

'+'    Открытие на чтение и запись

Возможны комбинации, например, 'r+' 'w+' 'rb+'

## Чтение из файла

Метод `read()`

- чтение информации из всего файла целиком (вызов без аргументов)
- чтение *n* символов из файла (вызов с аргументом – целым числом *n*)
- Пример

```
f = open('data.txt', 'r')
s = f.read()
```

Метод `readline()`

- чтение очередной строки из текстового файла  
если достигнут конец файла, результат – пустая строка

Метод `readlines()`

- чтение всего содержимого файла текстового файла  
результат – список строк

"Перебор" строк текстового файла в цикле (файл – итерируемый объект!)

```
f = open('text.txt')
for line in f:    // перебор всех строк
    ...
```

**Вопрос:** Как избавиться от `\n` в конце строки?

**Ответ:** Использовать метод `rstrip()` строки:

```
for line in f:
    line = line.rstrip()
    ...
```

## Запись в файл

Метод `write()`

- Пример

```
f = open('data.txt', 'a')
f.write('hello!')
```

- Допустим только один аргумент
- Аргумент должен иметь тип `str`

- Для перевода строки добавляем '\n'

Функция `print()` с параметром `file=...`

- Пример

```
f = open('data.txt', 'a')
print('hello!', 12345, 'hello!', file = f)
```

## Доступ к функциям операционной системы: модуль `os` \*

\*Разумеется, модуль `os` содержит не две функции, а несколько десятков. Но для решения задач про файлы без упомянутых ниже двух точно не обойтись.

Удаление файла

```
os.remove(file_name)
```

- `file_name` – это строка с именем файла, а не файловая переменная
- строка `file_name` должна содержать полный путь + имя файла; для файла в текущей папке путь можно опускать
- файл не должен быть открыт в программе

Переименование файла

```
os.rename(old_name, new_name)
```

- см. все предыдущие замечания по поводу `file_name`

## Задачи по теме "Файлы"

Найдено в интернете...

**Задача 1.** Дан текстовый файл `data1.txt`. Определите сколько в нем букв (латинского алфавита), слов, строк.

Выведите три найденных числа в следующем формате:

```
Input file contains:
... letters
... words
... lines
```

In [3]:

```
1 # Решение
2 n_lines = 0
3 n_words = 0
4 n_letters = 0
5 f = open('data1.txt', 'r')
6 # lines = f.readlines()
7 # n_lines = len(lines)
8 for line in f:
9     n_lines += 1
10    n_words += len(line.split())
11    for c in line:
12        if c.isalpha():
13            n_letters += 1
14 f.close()
15 print('Input file contains:')
16 print(f'{n_letters} letters')
17 print(f'{n_words} words')
18 print(f'{n_lines} lines')
```

Input file contains:  
184 letters  
36 words  
9 lines

**Задача 2.** В файле могут быть записаны любые символы, в том числе десятичные цифры. Числом назовем последовательность цифр, идущих подряд (т.е. число всегда неотрицательно). Вычислите сумму всех чисел, записанных в файле data2.txt .

*В данной задаче удобно считывать данные посимвольно.*

In [7]:

```
1 # Решение
2 f = open('data2.txt')
3 s = 0
4 num = ''
5 c = f.read(1)
6 while c!='':
7     if c.isdigit():
8         num += c
9     else:
10        if num:
11            s += int(num)
12            num = ''
13        c = f.read(1)
14 if num:
15     s += int(num)
16 print(s)
17 f.close()
```

3594

**Задача 3.** В олимпиаде по информатике принимало участие несколько человек. Победителем олимпиады становится человек, набравший больше всех баллов. Победители определяются независимо по каждому классу. Определите количество баллов, которое набрал победитель в каждом классе. Гарантируется, что в каждом классе был хотя бы один участник.

### Входные данные:

Информация о результатах олимпиады записана в файле, `data3.txt`, каждая строка которого имеет вид:

фамилия имя класс балл

- фамилия и имя — текстовые строки, не содержащие пробелов.
- класс - одно из трех чисел 9, 10, 11.
- балл - целое число от 0 до 100.

### Выходные данные:

Три числа: баллы победителя олимпиады по 9, по 10 и по 11 классу соответственно.

*В этой задаче файл необходимо считывать построчно, не сохраняя содержимое файла в памяти целиком.*

In [8]:

```
1 # Решение
2 f = open('data3.txt')
3 res = [0,0,0] #9, 10, 11 класс
4 for line in f:
5     _, _, klass, ball = line.rstrip().split()
6     klass = int(klass)
7     ball = int(ball)
8     res[klass-9] = max(res[klass-9], ball)
9 f.close()
10 print(res)
11
```

[100, 99, 98]

### Задачи с сайта [ptaskbook.com](http://ptaskbook.com)

**Text1.** Дано имя файла и целые положительные числа N и K. Создать текстовый файл с указанным именем и записать в него N строк, каждая из которых состоит из K символов «\*» (звездочка).

In [9]:

```
1 # Решение
2 name = 'stars.txt'
3 n = int(input('n = '))
4 k = int(input('k = '))
5 f = open(name, 'w')
6 for i in range(n):
7     print('*'*k, file=f)
8 f.close()
```

n = 15

k = 7

**Text5.** Дана строка S и текстовый файл. Добавить строку S в конец файла.

In [10]:

```
1 # Решение
2 s = 'hello'
3 file_name = 'stars.txt'
4 f = open(file_name, 'a')
5 print(s, file=f)
6 f.close()
```

**Text6.** Даны два текстовых файла. Добавить в конец первого файла содержимое второго файла.

In [11]:

```
1 # Решение
2 name1 = 'stars.txt'
3 name2 = 'data1.txt'
4 f1 = open(name1, 'a')
5 f2 = open(name2)
6 for line in f2:
7     f1.write(line)
8 f1.close()
9 f2.close()
```

**Text7.** Дана строка S и текстовый файл. Добавить строку S в начало файла.

In [13]:

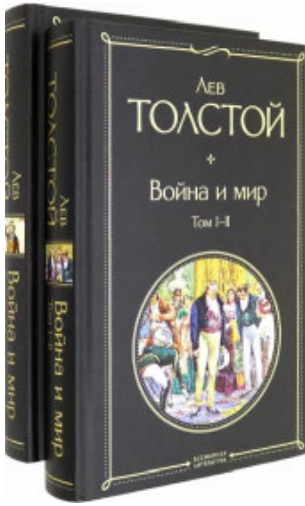
```
1 # Решение
2 import os
3 s = 'hello!!!'
4 name = 'stars.txt'
5 f = open(name)
6 g = open(name+'.tmp.$$$', 'w')
7 g.write(s+'\n')
8 for line in f:
9     g.write(line)
10 f.close()
11 g.close()
12 os.remove(name)
13 os.rename(name+'.tmp.$$$', name)
14
```

## Питонский стиль / Pythonic style

Знайки Питона работают с файлами, используя конструкцию `with ... as`:

```
with open("x.txt") as f:
    data = f.read()
    ## обрабатываем строчку и тп
```

Такая конструкция с одной стороны не требует явного закрытия файла (не нужно писать `f.close()`), а с другой – гарантирует закрытие файла в любом случае, какие бы ситуации не возникли в процессе его обработки.



#### Задача 4.

В файле `book.txt` содержатся два тома знаменитого романа Л.Н.Толстого "Война и мир". Определите, имя какого персонажа встречается в этом файле чаще всего?

In [2]:

```
1 # Решение
2 f = open('book1.txt')
3 names = {}
4 for line in f:
5     words = line.rstrip().split()
6     words = [w for w in words if len(w)>2 and w.istitle()]
7     for i in range(len(words)):
8         if not words[i][-1].isalpha():
9             words[i] = words[i][: -1]
10    for w in words:
11        names[w] = names.get(w, 0) + 1
12 res = sorted(names.items(), key=lambda x:x[1], reverse=True)
13 print(res[:25])
14
15
16 f.close()
```

```
[('Пьер', 613), ('Андрей', 587), ('Наташа', 497), ('Ростов', 430), ('Она', 385), ('Князь', 310), ('Что', 251), ('Николай', 240), ('Марья', 235), ('Как', 221), ('Соня', 194), ('Это', 188), ('Пьера', 187), ('Ну', 185), ('Анна', 184), ('Нет', 171), ('Да', 156), ('Долохов', 155), ('Денисов', 154), ('Андрея', 151), ('Борис', 148), ('Вот', 141), ('Василий', 138), ('Анатолий', 135), ('Ростова', 120)]
```



## Финальная задача

*Посвящается закончившемуся чемпионату мира и наступившему перерыву в чемпионате России по футболу.*

Дан текстовый файл, содержащий результаты матчей футбольного турнира в формате

<u>Команда-1</u>	<u>Команда-2</u>	<u>Счет</u>
Сочи	Химки	1 : 1
Спартак	Ахмат	2 : 0
Ротор	Зенит	0 : 2

На основании этого файла создать таблицу турнира в следующем формате:

№	Команда	И	В/Н/П	М	О
1	Зенит	17	13/3/1	47-7	42
2	Спартак	17	11/3/3	40-18	36
3	Ростов	17	10/5/2	32-23	35
4	Динамо М	17	8/5/4	33-21	29



In [15]:

```
1 # Решение
2 # 1. Структура данных
3 # [И, Поб, Н, Пор, Заб, Проп]
4 # 0 1 2 3 4 5
5 # {'НазваниеКоманды' : [...], ...}
6
7 # 2. Ввод из файла и обработка
8 table = {}
9 f = open('soccer_rus.txt')
10 for line in f:
11     team1, team2, score = line.rstrip().split()
12     data1 = table.get(team1, [0]*6)
13     data2 = table.get(team2, [0]*6)
14     goal1, goal2 = map(int, score.split(':'))
15     data1[0] += 1
16     data2[0] += 1
17     data1[4] += goal1
18     data2[4] += goal2
19     data1[5] += goal2
20     data2[5] += goal1
21     if goal1 > goal2:
22         data1[1] += 1
23         data2[3] += 1
24     elif goal1 < goal2:
25         data2[1] += 1
26         data1[3] += 1
27     else:
28         data1[2] += 1
29         data2[2] += 1
30     table[team1] = data1
31     table[team2] = data2
32 f.close()
33
34 # 3. Вывод "красивой" таблички
35
36 def score(data):
37     return data[1]*3 + data[2]*1
38
39 def print_team(t):
40     g = open('table_res.txt', 'a')
41     d = table[t]
42     print(f'{t:15} {d[0]} {d[1]:2}/{d[2]:2}/{d[3]:2} {d[4]:2}-{d[5]:2} {score(d)}', fi
43     g.close()
44
45 teams = table.keys()
46 teams = sorted(teams, key=lambda x:-score(table[x]))
47 for t in teams:
48     print_team(t)
49 print('Done. Cm. file')
```

Done. Cm. file

In [6]:

```
1 table
```

Out[6]:

```
{'Химки': [17, 3, 3, 11, 16, 36],  
'Зенит': [17, 13, 3, 1, 47, 7],  
'ЦСКА': [17, 8, 5, 4, 30, 18],  
'Урал': [17, 6, 3, 8, 21, 28],  
'Оренбург': [17, 8, 1, 8, 33, 37],  
'КрыльяСоветов': [17, 5, 5, 7, 20, 25],  
'Ахмат': [17, 8, 2, 7, 27, 27],  
'Спартак': [17, 11, 3, 3, 40, 18],  
'ТорпедоМ': [17, 1, 3, 13, 7, 30],  
'Сочи': [17, 7, 4, 6, 27, 32],  
'Локомотив': [17, 3, 4, 10, 20, 34],  
'ПариНН': [17, 5, 4, 8, 22, 29],  
'Краснодар': [17, 7, 4, 6, 31, 29],  
'Факел': [17, 1, 10, 6, 21, 33],  
'ДинамоМ': [17, 8, 5, 4, 33, 21],  
'Ростов': [17, 10, 5, 2, 32, 23]}
```

In [ ]:

```
1 Содержимое файла table_res.txt  
2  
3 Зенит          17 13/ 3/ 1 47- 7 42  
4 Спартак       17 11/ 3/ 3 40-18 36  
5 Ростов        17 10/ 5/ 2 32-23 35  
6 ЦСКА          17  8/ 5/ 4 30-18 29  
7 ДинамоМ       17  8/ 5/ 4 33-21 29  
8 Ахмат         17  8/ 2/ 7 27-27 26  
9 Оренбург      17  8/ 1/ 8 33-37 25  
10 Сочи         17  7/ 4/ 6 27-32 25  
11 Краснодар    17  7/ 4/ 6 31-29 25  
12 Урал          17  6/ 3/ 8 21-28 21  
13 КрыльяСоветов 17  5/ 5/ 7 20-25 20  
14 ПариНН       17  5/ 4/ 8 22-29 19  
15 Локомотив    17  3/ 4/10 20-34 13  
16 Факел        17  1/10/ 6 21-33 13  
17 Химки        17  3/ 3/11 16-36 12  
18 ТорпедоМ     17  1/ 3/13  7-30  6
```

In [ ]:

```
1
```