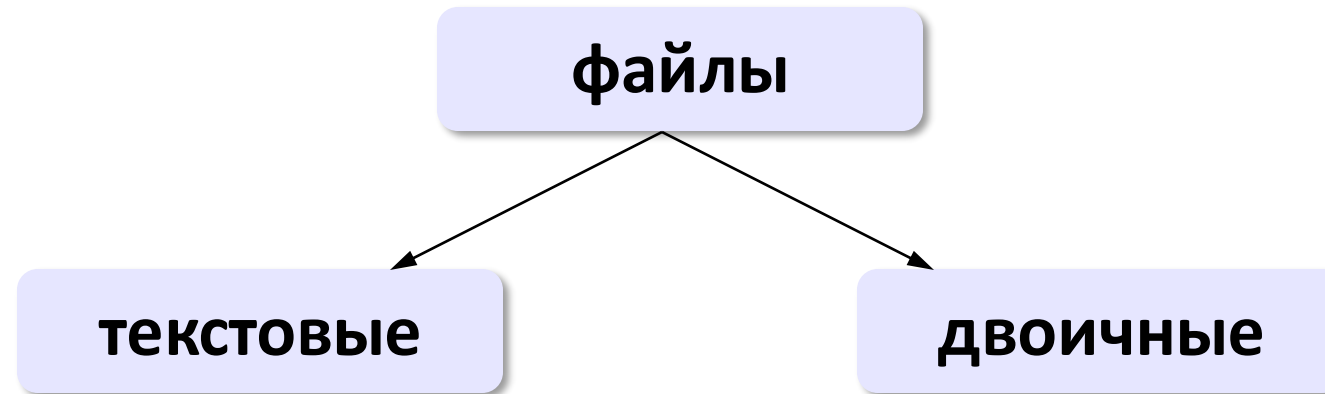


Файловый ввод вывод

Как работать с файлами?



«*plain text*»:

- текст, разбитый на строки;
- из специальных символов только символы перехода на новую строку

- любые символы
- рисунки, звуки, видео, ...

Метод `split()`

Преобразование строки в список осуществляется с помощью метода `split()`, по умолчанию, в качестве разделителя, используется пробел.

```
line = input().split()
# вводим числа 1 2 3 4 5 6
print(line)
# результат ['1', '2', '3', '4', '5', '6']
```

Для считывания списка чисел с одновременным приведением их к типу `int` можно воспользоваться вот такой конструкцией.

```
nums = map(int, input().split())
# вводим числа 1 2 3 4 5 6
print(list(nums))
# результат [1, 2, 3, 4, 5, 6]
```

Открытие и закрытие файла

Для открытия файла используется функция **open()**, которая возвращает файловый объект. Наиболее часто используемый вид данной функции выглядит так:

open(имя_файла, режим_доступа).

Для указания режима доступа используются следующие символы:

'r' – открыть файл для чтения;

'w' – открыть файл для записи;

'x' – открыть файл с целью создания, если файл существует, то вызов функции **open** завершится с ошибкой;

'a' – открыть файл для записи, при этом новые данные будут добавлены в конец файла, без удаления существующих;

'b' – бинарный режим;

't' – текстовый режим;

'+' – открывает файл для обновления.

По умолчанию файл открывается на чтение в текстовом режиме.

Открытие и закрытие файла

У файлового объекта есть следующие атрибуты.

file.closed – возвращает *true* если файл закрыт и *false* в противном случае;

```
f = open("test.txt", "w")
# создаёт test.txt
print("file.mode: " + f.mode)
#file.mode: w
f = open("test.txt", "r")
print("file.closed: " + str(f.closed))
#file.closed: False
print("file.mode: " + f.mode)
#file.mode: r
print("file.name: " + f.name)
#file.name: test.txt
```

Для закрытия файла используется метод *close()*.

Принцип работы

← открыть файл

← работа с файлом

← закрыть файл

файловые переменные-
указатели

по умолчанию – на чтение
(режим **"r"**)

```
Fin = open ( "input.txt" )  
Fout = open ( "output.txt", "w" )  
# здесь работаем с файлами  
Fin.close()  
Fout.close()
```

"r" - чтение
"w" – запись
"a" – добавление

Чтение данных из файла

Чтение данных из файла осуществляется с помощью методов *read(размер)* и *readline()*.

Метод *read(размер)* считывает из файла определенное количество символов, переданное в качестве аргумента. Если использовать этот метод без аргументов, то будет считан весь файл.

```
f = open("test.txt", "w")  
# создаёт test.txt  
for i in range(10):  
    s=str(i)+" "  
    f.write(s)  
f = open("test.txt", "r")  
text=f.readlines()  
print(text)  
#['0 1 2 3 4 5 6 7 8 9 ']
```

Чтение данных из файла

В качестве аргумента метода можно передать количество символом, которое нужно считать.

```
f = open("test.txt", "r")
text=f.read (10)
print(text)
#0 1 2 3 4
```

Метод *readline()* позволяет считать строку из открытого файла.

```
f = open("test.txt", "w")
text=f.readline()
# text=f.read() и text=f.readline()
# обе функции выполняют одну и ту же действию
print(text)
#0 1 2 3 4
```


Чтение данных из файла

Построчное считывание можно организовать с помощью оператора *for*.

```
f = open("test.txt", "r")
text=f.read().split()
for simvol in text:
    print(simvol)
```

#0

#1

#2

#3

#4

#5

#6

#7

#8

#9

Запись данных в файл

Для записи данных файл используется метод **write**(строка), при успешной записи он вернет количество записанных символов.

```
f = open("test.txt", "a")
f.write(" This text would be written in the text.txt")
f = open("test.txt", "r")
text=f.read()
print(text)
#0 1 2 3 4 5 6 7 8 9 This text would be written in the text.txt
```

Чтение неизвестного количества данных

Задача. В файле записано в столбик неизвестное количество чисел. Найти их сумму.

пока не конец файла
прочитать число из файла
добавить его к сумме

```
Fin = open ( "input.txt" )  
sum = 0  
while True:  
    s = Fin.readline()  
    if not s: break  
    sum += int(s)  
Fin.close()
```

если конец файла, вернёт
пустую строку

Чтение неизвестного количества данных

Задача. В файле записано в столбик неизвестное количество чисел. Найти их сумму.

```
sum = 0
Fin = open ( "input.txt" )
lst = Fin.readlines()
for s in lst:
    sum += int(s)
Fin.close()
```

прочитать все строки в
список строк

Чтение неизвестного количества данных

Задача. В файле записано в столбик неизвестное количество чисел. Найти их сумму.

```
sum = 0
with open ( "input.txt" ) as Fin:
    for s in Fin:
        sum += int(s)
```

или так:

```
sum = 0
for s in open ( "input.txt" ):
    sum += int(s)
```




Не нужно закрывать файл!


Задачи

- «А»: Напишите программу, которая находит среднее арифметическое всех чисел, записанных в файле в столбик, и выводит результат в другой файл.
- «В»: Напишите программу, которая находит минимальное и максимальное среди чётных положительных чисел, записанных в файле, и выводит результат в другой файл. Учтите, что таких чисел может вообще не быть.
- «С»: В файле в столбик записаны целые числа, сколько их – неизвестно. Напишите программу, которая определяет длину самой длинной цепочки идущих подряд одинаковых чисел и выводит результат в другой файл.

Обработка массивов

Задача. В файле записаны в столбик целые числа. Вывести в другой текстовый файл те же числа, отсортированные в порядке возрастания.

 В чем отличие от предыдущей задачи?

 Для сортировки нужно удерживать все элементы в памяти одновременно.

Задачи

- «А»: В файле в столбик записаны числа. Отсортировать их по возрастанию последней цифры и записать в другой файл.
- «В»: В файле в столбик записаны числа. Отсортировать их по возрастанию суммы цифр и записать в другой файл. Используйте функцию, которая вычисляет сумму цифр числа.
- «С»: В двух файлах записаны отсортированные по возрастанию массивы неизвестной длины. Объединить их и записать результат в третий файл. Полученный массив также должен быть отсортирован по возрастанию.

Обработка строк

Задача. В файле записано данные о собаках: в каждой строчке кличка собаки, ее возраст и порода:

Мухтар 4 немецкая овчарка

Вывести в другой файл сведения о собаках, которым меньше 5 лет.

пока не **конец файла** Fin

прочитать строку из файла Fin

разобрать строку – выделить возраст

если возраст < 5 то

записать строку в файл Fout