

Работа с файлами

Файлы, в общем случае, делятся на два вида:

- **Текстовые** файлы – текст, разбитый на строки, состоящий из различных символов. Из специальных символов данный тип файлов содержит только символы перехода на новую строку ‘\n’
- **Двоичные** файлы – может содержать любые символы, также может являться рисунками, звуками, видео, в общем, любой файл можно открыть как двоичный.

Работа с файлами на любом языке программирования строится по принципу сэндвича: хлебом (верхней и нижней частью) являются процедуры открытия и закрытия файлов, а начинкой (то, что находится между хлебом) является работой с файлом. Именно в начинке происходит чтение информации из файла и её обработка



Файлы можно открыть для следующих манипуляций:

- **Чтение** информации из файла (для того, чтобы файл можно было прочесть, он должен существовать)
- **Запись** информации в файл (вся информация, которая находилась в файле удаляется и записывается новая информация, если файл до этого не существовал, то он создается)
- **Добавление** информации в файл (информация в файл добавляется в самый конец, после старой информации, которая была в файле)

На языке Python открытие файла делается при помощи функции `open(file_name, mode, encoding)`, где параметрами являются:

- **file_name** – название файла и путь к нему
- **mode** – режим открытия файла (чтение, запись, добавление и т.п.)
- **encoding** – кодировка, в которой будет происходить считывание файла (utf-8, utf-16, ASCII и т.п.)

Закрытие файла выполняется при помощи метода `close()`, вызываемой без параметров.

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 осуществляется несколькими способами:

Ввод данных

```
Fin = open( "input.txt" )
```

Чтение строки:

```
s = Fin.readline()      # "1 2"
```

Чтение строки и разбивка по пробелам:

```
s = Fin.readline().split()      # ["1","2"]
```

Чтение целых чисел:

```
s = Fin.readline().split()      # ["1","2"]  
a, b = int(s[0]), int(s[1])
```

или так:

```
a, b = [int(x) for x in s]
```

или так:

```
a, b = map( int, s )
```

Запись данных в файл осуществляется следующим образом:

```
a = 1  
b = 2  
Fout = open( "output.txt", "w" )  
Fout.write( "{:d} + {:d} = {:d}\n".format(  
            a, b, a+b) )  
Fout.close()
```

Следует отметить, что перед записью информации в файл, необходимо преобразовать всю записываемую информацию в строковый формат.

Выше был описан способ, когда нам заранее известно количество строк, которые необходимо прочитать из файла. Но чаще всего перед нами стоит задача **читать информацию из файла, в котором заранее неизвестно количество строк**. Для решения данной задачи можно использовать следующую конструкцию (в ней решается задача, в которой необходимо посчитать сумму цифр, записанных в файл по одному в строке и их количество неизвестно)

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

Есть **ещё один способ**, считать все числа из файла разом и добавить их в список, затем перевести все строки, записанные в массив, в числовой тип данных и найти их сумму.

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

Также, есть **ещё один способ** решения данной задачи. Для этого циклу for в качестве итерируемого элемента можно передать результат работы функции open(). После считывания информации данным способом, не нужно закрывать файл, он закроется автоматически.

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

Если данные из файла необходимо записать в список и обработать их как единый набор данных (например, отсортировать), то можно использовать следующую конструкцию

Ввод массива:

```
A = []
while True:
    s = Fin.readline()
    if not s: break
    A.append ( int(s) )
```

Ввод в стиле Python:

```
s = Fin.read().split()
A = list ( map(int, s) )
```

Если же необходимо записать информацию из списка в файл, то можно использовать следующие конструкции

Вывод результата:

```
Fout = open ( "output.txt", "w" )
Fout.write ( str(A) )
Fout.close()
```

[1, 2, 3]

или так:

```
for x in A:
    Fout.write ( str(x)+"\n" )
```

1

2

3

или так:

```
for x in A:
    Fout.write ( "{:4d}".format(x) )
```

1 2 3

Задачи:

№1

Получить 10 случайных четных чисел и записать их в файл chet.txt

№2

Найти сумму всех чисел, записанных в файл chet.txt

№3

Переписать все числа из файла chet.txt, кратные 6 в файл chet6.txt

№4

Напишите программу, которая находит среднее арифметическое всех чисел, записанных в файле в столбик, и выводит результат в другой файл.

№5

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.

№6

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.