# Урок 18.02. Базовые операции с файлами (open, close, read)

Мы уже хорошо поработали с путями и имеем представление как с ними взаимодействовать. Однако для нас еще остается загадкой, как работать с самими файла, как их открывать и читать. И сегодня мы займемся именно этим.

Рассмотрит вот такую простую задачку «Участники конференции».

Нам передали список, выступающих на конференции, участников в виде файла. В каждой строек содержится фамилия, имя и возраст выступающего. Наша задача проста: нам необходимо вывести на экран содержимое файла. И для работы программы (чтения файла) нам нужен сам файл, для этого давайте создадим такой тестовый файл speakers.txt:

Иванов Иван 15

Петров Петр 17

Степанова Лена 14

Кузнецова Анна 16

И теперь можем приступать к написанию программы. И тут нам нужно понять, что для работы с файлом мы всегда должны его открыть. А в программировании все объекты хранятся в переменных, так и тут нам нужно создать переменную, в которой мы будем как бы «открывать» наш файл. Напишем такой код:

speakers\_file = open('speakers.txt')

Вот так мы открыли файл. Однако, для разграничения доступа и исключения ошибок при открытии файла чаще всего указывается назначение, с которым производится открытие файла. В нашем случае мы хотим прочитать файл, по-английски это read или сокращенно r, именно это сокращенное назначение и нужно указать как режим открытия файла.

speakers\_file = open('speakers.txt', 'r')

Запуск такой программы ни к чему не привело. И как вы уже догадались (и я об этом говорил) наша переменная это объект, который хранит открытый наш файл, конечно, не сам файл, а ссылку на него на диске. И для получения из этого обхекта наших заветных данных используется метод read, который сам за себя говорит. Правда этот метод читает только одну строку.

speakers\_file = open('speakers.txt', 'r')

data = speakers\_file.read()

print(data)

Запустим нашу программу… И получим либо ошибку, либо набор непонятных символов. Всё дело в кодировке. Возможно вы знаете, что существует несколько кодировок символов кодами, например, Win-1251, KOI8-R, UTF-8 и другие. Так вот причина такого поведения нашей программы это и есть несовпадение кодировок. И для прямого указания кодировки нам нужно дописать в команде open именованный параметр encoding='UTF-8'.

speakers\_file = open('speakers.txt', 'r', encoding='UTF-8')

data = speakers\_file.read()

print(data)

Запускаем и наша программа работает. Но мы забыли одну важную вещь, а именно закрыть файл. Это очень важное и обязательное условие работы с файлами, если файл не закрывать, то мы можем получить множество не приятных эффектов (начиная от ненужного расхода памяти и отсутствия доступа других программ к этому файлу, до полной потери файла).

speakers\_file = open('speakers.txt', 'r', encoding='UTF-8')

data = speakers\_file.read()

print(data)

speakers\_file.close()

Тут я открою вам небольшую тайну, что объект файлового типа это итератор, а значит его можно считывать в цикле вот так:

speakers\_file = open('speakers.txt', 'r', encoding='UTF-8')

# data = speakers\_file.read()

# print(data)

for i\_line in speakers\_file:

 print(i\_line)

speakers\_file.close()

Однако наши строки стали писаться через одну, тут дело в том, что в файле стоят символы окончания строки и наш print также добавляет свои концы строк. А еще, как вы, думаю, уже поняли, метод read читает сразу весь файл, и второй раз он уже не сработает. Однако если нам нужно работать с файлом построчно, когда файл очень большой, то мы можем использовать цикл или использовать метод readline (чтение файла построчно).