

26. Базовые операции для работы с файлами в Python. Использование инструкции with ... as для работы с файлами.

Файл — именованная область данных на носителе информации. Работа с файлами реализуется средствами операционных систем. Операционная система предоставляет приложениям набор функций и структур (API) для работы с файлами.

Можно выделить два типа операций с файлом — связанные с его открытием и выполняющиеся без его открытия.

- Операции первого типа обычно служат для чтения и записи информации или подготовки к чтению или записи.
- Операции второго типа выполняются с файлом как с «объектом» файловой системы, т.е. для его переименования, копирования, перемещения и т.д.

Для первого типа работы с файлами используется конструкция with open .. as название переменной.

Пример конструкции:

```
with open("./text.txt", "w") as file:  
    file.write("какие-то там данные")
```

В функцию open можно передать три параметра: имя файла, режим открытия файла и кодировка при чтении файла, последняя по-умолчанию utf-8.

Режимы открытия файла:

r — открытие на чтение;

w — открытие на запись, содержимое файла удаляется, если файла нет, создается новый;

a — открытие на дозапись, информация добавляется в конец файла.

Также можно добавить b для работы с файлами в побитовом режиме, те:

rb - r для побитовой записи

wb - w для побитовой записи

ab - a для побитовой записи

При работе с файлами используется 2 метода: write и read. Как следствие, write используем с аргументом w и a, a read - r

Пример чтения из файла:

```
with open ("./kot.txt", "r") as file:  
    this_str = file.read()  
    print(this_str)
```

Пример дозаписи в файл:

```
with open ("./kot.txt", "a") as file:  
    file.write("\nпривет\n")
```

Также существует альтернатива с чтением/записью файлов без использования

конструкции with open .. as ...:

Пример чтения:

```
this_file = open("./kot.txt", "r")
this_str = this_file.read()
this_file.close() #ОБЯЗАТЕЛЬНО, даже при r!
print(this_str)
```

Пример записи:

```
this_file = open("./kot.txt", "w")
this_file.write("meow meow")
this_file.close() #ОБЯЗАТЕЛЬНО, иначе потеря данных
```

Недостатком является то, что необходимо вручную закрывать файл с помощью метода file.close(), когда конструкция with open .. as ... делает это автоматически.

ПРОГРАММИСТ ПОМНИ! Всегда закрывай файл с помощью f.close, если не используешь конструкцию with open .. as:

Для второго типа работы в Python используется модуль os. Во несколько примеров работы:

```
import os
os.remove("text.txt") #удаление файла
os.mkdir("DIR") #создание директории DIR
os.rename("OLD", "NEW") #переименование файла OLD в NEW
```