Лекция 12

Чтение и запись данных в файл

Переменные указывающие на файл

В Python для чтения и записи строковой информации в файл сначала нужно создать переменную, указывающую на этот файл на диске.

Для создания переменных такого типа используется встроенная функция **open**.

Синтаксис ее использования таков:

variable_name = open (file_address, access_mode)

Где

variable_name - имя переменной указывающей на файл

file_address - абсолютный или относительный адрес файла на диске

access_mode - режим доступа в виде строки

Адресация файла

Для указания, где на диске расположен или должен быть создан файл, используется его адрес в виде строки.

Адрес бывает :

- Абсолютный — вы указываете расположение файла относительно верха дисковой иерархии. Например, в ОС Windows это может выглядеть как «C:\temp\a.txt»
- Относительный — адрес вычисляется относительно места, из которого запускается ваша программа. Так адрес вида «a.txt» указывает на то, что файл «a.txt» расположен в той же папке, что и ваша программа.

Внимание! Если вы используете абсолютную адресацию, то перед строковым литералом адреса нужно поставить букву **r**. Такая запись позволит Python корректно обработать адрес.
Например, open(r"C:/temp/a.txt","r"))

Режим доступа

Для указания того, с применением каких атрибутов (режимов доступа) будет открыт файл, используется строка, в которой буквой указывается тип применяемого атрибута.

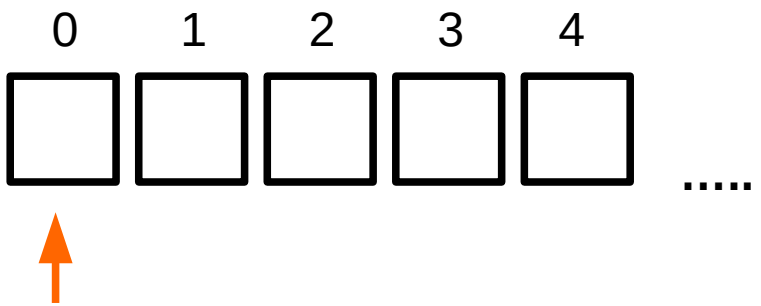
- «**r**» - Открывает файл только для чтения. Указатель стоит в начале файла.
- «**r+**» - Открывает файл для чтения и записи. Указатель стоит в начале файла.
- «**w**» - Открывает файл только для записи. Указатель стоит в начале файла. Создает файл, если его не существует.
- «**w+**» - Открывает файл для чтения и записи. Указатель стоит в начале файла. Создает файл, если его не существует.
- «**a**» - Открывает файл для добавления информации в файл. Указатель стоит в конце файла. Создает файл, если его не существует.
- «**a+**» - Открывает файл для добавления и чтения. Указатель стоит в конце файла. Создает файл, если его не существует.

Внимание! Эти режимы применяются только для символьных данных! Существуют режимы и для двоичных данных, но в этом курсе мы их не рассматриваем.

python

Указатель в файле

Вы можете представлять файл, как последовательность символов. Каждому символу присваивается порядковый номер, начиная с 0, и далее по возрастанию с шагом 1.



Например, пусть файл состоит из 5 символов. Тогда квадраты обозначают символы в файле. Цифры над квадратами обозначают индексы (номера по порядку) этих символов. Оранжевая стрелка, это - указатель. При чтении или записи, он считывает или добавляет символ и передвигается на одну позицию вправо.

Пример создания переменной указывающей на файл

```
file_one = open("a.txt","w")
```

Имя переменной Адрес файла Режим доступа

В данном примере переменная с именем file_one указывает на файл с адресом «a.txt». Это - относительный адрес, следовательно этот файл появится в той же папке, из которой запущена ваша программа. Тут выбран режим доступа «w», что означает, что файл открыт для записи, а если такого файла нет, то он будет создан.

Запись данных в файл

Для записи данных в файл нужно выполнить следующую последовательность действий:

1) Открыть файл в режиме записи
2) С помощью методов **write** и **writelines** записать требуемые данные в файл
3) С помощью метода **close** закрыть файл

Что такое метод? До сих пор не было такого термина.

Метод - это функция, которая принадлежит классу.

Класс - это пользовательский или встроенный, производный тип данных. Классы, их создание и использование будет рассмотрено в курсе для продвинутых. На данном моменте важно понимать, что ,кроме этой особенности, метод ведет себя точно также, как и функция.

python

Методы для записи write и writelines

Как использовать эти методы?

Для их использования нужно указать имя переменной, которая ссылается на файл, поставить точку и написать имя того или иного метода.

В чем разница между ними?

write (str) запишет в файл строку str. **Внимание!** В отличие от print() он не переводит строку автоматически.

writelines(list_str) запишет в файл все строки, расположенные в списке list_str

python

Пример записи одной строки в файл

```
file_one = open("Hello.txt","w")

file_one.write("Hello world")

file_one.close()
```

Файл открыли на запись

Записали в него одну строку - «Hello world»

Закрыли файл

После выполнения данного кода в каталоге, в котором расположен файл вашей программы, появится файл с именем «Hello.txt», в котором будет одна строка текста «Hello world».

Более сложный пример записи

```
students_list = ["Olga","Sergiy","Petr","Lidia"]

file_one = open("Students.txt","w")

for i in range(len(students_list)):
    file_one.write(str(i+1)+") "+str(students_list[i])
+"\n")

file_one.close()
```

В этом примере в файл запишется список имен студентов, перед которыми следуют их порядковые номера. Обратите внимание на то, что для выполнения перевода строки приходится указывать символ перевода строки «**\n**» в явном виде.

Чтение данных из файла

Для чтения данных из файла нужно выполнить следующую последовательность действий:

1) Открыть файл в режиме чтения
2) С помощью методов **read, readline** и **readlines** считать требуемые данные из файла
3) С помощью метода **close** закрыть файл

Методы для чтения из файла

read() - возвращает строку содержащую все символы, хранящиеся в файле.

read(N) - возвращает строку содержащую очередные N символов из файла.

readline() - читает содержимое файла до ближайшего символа \n и возвращает строку.

readlines() - читает файл целиком и возвращает список строк.

python

Пример чтения из файла

```
file_one = open("cars.txt","r")

text = file_one.read()

print(text)

file_one.close()
```

Файл открыли на чтение

Считали все содержимое в одну строку

Закрыли файл

Удобный способ чтения из файла

В случае, когда нужно построчно вычитать текстовый файл, существует удобный способ. Для этого нужно получить переменную связанную с файлом, и использовать цикл for для построчной вычитки из файла.

Синтаксис такого приема примерно таков:

```
file_1 = open(file_address, access_mode)

for text in file_1:
    ....
```

В переменную цикла text будет вычитано построчно все содержимое текстового файла.

Пример построчного считывания из файла

```
file_one = open("cars.txt","r")

for text in file_one:
    print(text)

file_one.close()
```

Как перемещаться по файлу

При чтении или записи действие всегда происходит от текущего положения указателя. После записи или чтения из файла n символов, указатель сдвигается на n позиций. Т.е. получается последовательное продвижение файлового указателя от начала файла к его концу.

В случае возникновения необходимости, вы можете передвинуть указатель. Для этого используется метод **seek**.

Синтаксис его использования:

seek(N) — установит указатель на N-й символ от начала файла. Например, seek(0) установит указатель в начало файла. seek(5) установит указатель на 5 символ.

python

Пример перемещения по файлу

```
file_one = open("a.txt","r")
```

Файл открыли на чтение

```
text = file_one.read(4)
print(text)
```

Вычитали первые 4 символа из файла

```
file_one.seek(0)
```

Вернулись в начало файла

```
text = file_one.read(5)
print(text)
file_one.close()
```

Вычитали первые 5 символов из файла

В этом примере из файла сначала вычитываются четыре символа. Потом, с помощью метода seek, указатель возвращается в нулевой символ файла, т. е. в самое начало. Затем повторно вычитываются, но теперь уже первые пять символов файла.

Список использованной литературы

1) Лутц М. Программирование на Python том-I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011 - 207 — 233 с.

Домашнее задание

1) Напишите программу, которая запишет в файл список целых чисел.
2) Напишите программу, которая вычитает текст из текстового файла и выведет на экран, сколько раз в тексте встречается буква «А».
3) Создайте консольный «текстовый редактор» с возможностью сохранения набранного текста в файл. (Не переусердствуйте. Имеется в виду, что вы сначала должны считать несколько строк с клавиатуры, а потом сохранить считанный текст в файл).
4) Напишите программу, которая найдет самое длинное слово в текстовом файле.

Дополнительное домашнее задание

1) Считайте из текстового файла текст на английском языке и выведите статистику по частоте использования букв в тексте (т. е. буква — количество использований). Причем первыми должны выводиться буквы используемые чаще всего.
2) Напишите программу, которая вычитает текст из двух текстовых файлов. Программа должна найти и записать в файл «result.txt» слова, которые есть и в первом, и во втором файле одновременно. Например, если в первом файле записано «Hello world», а во втором «Hello Java», то в результирующем файле должно быть слово «Hello» так, как только это слово есть и в первом и втором файле одновременно.