Теоретический материал

Теоретический материал по файловому вводу-выводу

Сайт: Дистанционная подготовка
Курс: Д. П. Кириенко. Программирование на языке Python (школа 179 г. Москвы)
Book: Теоретический материал
Printed by: maung myo
Date: Воскресенье 4 Март 2018, 01:30

Table of Contents

Файловый ввод-вывод

Файловый ввод-вывод

Открытие файла

Для каждого файла, с которым необходимо производить операции ввода-вывода, нужно связать специальный объект - поток. Открытие файла осуществляется функцией open, которой нужно передать два параметра. Первый параметр (можно также использовать именованный параметр file) имеет значение типа str, в котором записано имя открываемого файла. Второй параметр (можно также использовать именованный параметр mode) —это значение типа str, которое равно "r", если файл открывается для чтения данных (read), "w", если на запись (write), при этом содержимое файла очищается, и "a" — для добавления данных в конец файла (append). Если второй параметр не задан, то считается, что файл открывается в режиме чтения.

Функция open возвращает ссылку на файловый объект, которую нужно записать в переменную, чтобы потом через данный объект использовать методы ввода-вывода. Например:

```
input = open('input.txt', 'r')
output = open('output.txt', 'w')
```

Чтение данных из файла

Для файла, открытого на чтение данных, можно вызывать следующие методы, позволяющие читать данные из файла.

Метод **readline()** считывает одну строку из файла (до символа конца строки '\n', возвращается считанная строка вместе с символом '\n'. Если считывание не было успешно (достигнут конец файла), то возвращается пустая строка. Для удаления символа '\n' из конца файла удобно использовать метод строки rstrip(). Например: s = s.rstrip().

Метод **readlines()** считывает все строки из файла и возвращает список из всех считанных строк (одна строка — один элемент списка). При этом символы '\n' остаются в концах строк.

Метод **read()** считывает все содержимое из файла и возвращает строку, которая может содержать символы '\n'. Если методу read передать целочисленный параметр, то будет считано не более заданного количества символов. Например, считывать файл побайтово можно при помощи метода read(1).

Вывод данных в файл

Данные выводятся в файл при помощи метода **write**, которому в качестве параметра передается одна строка. Этот метод не выводит символ конца строки '\n' (как это делает функция print при стандартном выводе), поэтому для перехода на новую строку в файле необходимо явно вывести символ '\n'.

Также можно выводить данные в файл при помощи функции print, если передать ей еще один именованный параметр file, равный ссылке на открытый файл. Например:

```
output = open('output.txt', 'w')
print(a, b, c, file=output)
```

Закрытие файла

После окончания работы с файлом необходимо закрыть его при помощи метода close().

Пример

Следующая программа считывает все содержимое файла input.txt, записывает его в переменную s, а затем выводит ее в файл output.txt.

```
input = open('input.txt', 'r')
output = open('output.txt', 'w')
s = input.read()
output.write(s)
input.close()
output.close()
```

А вот аналогичная программа, но читающая данные посимвольно:

```
input = open('input.txt', 'r')
output = open('output.txt', 'w')
c = input.read(1)
while len(c) > 0:
    output.write(c)
    c = input.read(1)
input.close()
output.close()
```