Файлы

Python позволяет оперировать разными типами документов: текстовыми в любом формате, графическими, медиа, табличными и json. Для этого имеются соответствующие модули и пакеты (так встроенные, так и те, которые необходимо дополнительно установить), а также ряд функций.

При работе с файлами в Python используется ряд функций и методов:

- функция **open()** - открывает файл для чтения, записи, добавления нового содержимого. Может принимать дополнительные параметры: для задания режима открытия, указания кодировки, вывода ошибок и др. Возвращает дескриптор файла, который обязательно нужно закрыть, иначе файл останется в памяти. Дескриптор в данном случае представляет собой путь к документу в виде строки;
- функция **close()** - закрывает файловый объект;
- инструкция **with** (позволяет автоматически закрывать файловый объект после работы с ним);
- метод **read()** - для чтения содержимого документа;
- метод **readlines()** - преобразует все строки файла в список;
- метод **readline()** - построчно выводит данные файла (удобно при работе с массивными документами);
- метод **write()** - записывает новую информацию в файл;
- функция **rename()** из модуля **os** - переименовывает документ и др.

При решении заданий потребуются следующие знания:

1. Способы открытия файлов в разных режимах;
2. Варианты задания кодировок;
3. Методы чтения содержимого документов;
4. Инструменты для записи файлов;
5. Популярные библиотеки для работы с файлами (**os**, **csv**, **json**, **Pillow** и др.).

Полезные материалы можно посмотреть по ссылкам:

Работа с файлами в python. Чтение и запись в файл ~ PythonRu
Основы работы с файлами в Python (tproger.ru)
Библиотека os в Python 3: описание модуля и методов — Примеры использования функций (all-python.ru)
Работа с CSV файлами в Python 3: чтение и запись (all-python.ru)
JSON в Python - Примеры работы модуля JSON (python-scripts.com)
Обработка изображений с помощью библиотеки Python Pillow / Хабр (habr.com)

Задача 1

Напишите функцию **read_last_lines(lines, file)**, которая будет открывать определенный файл **file** и выводить на печать построчно последние строки в количестве **lines** (на всякий случай проверим, что задано положительное целое число). Протестируем функцию на файле со следующим содержимым:

```
Листва зеленела
Вечерело
Тучи разошлись
Жужжали мухи
Светил фонарик
Кипела вода в чайнике
Венера зажглась на небе
Деревья шумели
```

Задача 2

Выберите любую папку на своем компьютере, имеющую вложенные директории. Выведите на печать в терминал ее содержимое, как и всех подкаталогов при помощи функции **print_content(directory)**.

Проход по все каталогам и файлам в определенной директории можно осуществить при помощи функции walk() модуля os.

Задача 3

Исходный файл тот же, что в задании 1.

Требуется реализовать функцию **most_long_words(file)**, которая выводит слово, имеющее максимальную длину (или список слов, если таковых несколько).

Задача 4*

Требуется создать csv-файл **file_row_100.csv** со следующими столбцами:

– **№** - номер по порядку (от **1** до **100**);

– **Секунда** – текущая секунда на вашем ПК;

– **Микросекунда** – текущая миллисекунда на часах.

На каждой итерации цикла искусственно приостанавливайте скрипт на **0,02** секунды.

Для работы с файлами подобного текстового формата потребуется встроенная в Python библиотека csv, а также библиотеки datetime и time.

Задача 5*

При помощи библиотеки **Pillow** в директории **Circles** (нужно ее создать во время выполнения функции) нарисуйте и сохраните **100** кругов радиусом **300** пикселей случайных цветов в формате **jpg** на белом фоне (каждый круг - отдельный файл). Для этого напишите функцию **figure_gen (num_circles=100)**.

Для выполнения задания потребуется установить модуль Pillow (PIL).

Далее потребуется выполнить импорт:

```
from PIL import Image, ImageDraw
```