**Лабораторная работа № 5**

**РАБОТА С ФАЙЛАМИ. РАЗРАБОТКА СИНТАКСИЧЕСКОГО АНАЛИЗАТОРА. ВЫВОД ФОРМАТИРОВАННЫХ ДАННЫХ В ФОРМАТЕ JSON**

**Вариант 10.**

**Цель работы:** изучить работу с файлами с помощью функций из стандартной библиотеки; рассмотреть понятие синтаксического анализа текста и ознакомиться с его применением на языке Python; изучить возможности взаимодействия Python с форматом хранения данных JSON.

**Краткая теория**

Python поддерживает работу с множеством различных типов файлов. Условно их можно разделить на текстовые и бинарные. Текстовые файлы – это те файлы, которые хранят данные в текстовом виде, а бинарные – это, к примеру видео, изображение и прочее.

Python позволяет работать с файлами двумя способами:

− непосредственного открытия и закрытия файла с помощью встроенных функций open(path, mode) и close();

− с применением конструкции with.

Чтение файла и запись в файл осуществляются с помощью методов read() и write() соответственно. Python позволяет работать с файлом в определенных режимах, передаваемых функции open в качестве строкового аргумента для параметра mode:

1) «r» (Read) – файл открывается для чтения. Если файл не найден, то генерируется исключение FileNotFoundError;

2) «w» (Write) – файл открывается для записи. В любом случае создается новый. Следовательно, старые данные будут удалены;

3) «a» (Append) – файл открывается для дополнительной записи. Если файл отсутствует, то создастся новый. При существовании такого файла, данные запишутся в конец;

4) «b» (Binary) – используется для работы с бинарными файлами и применяется вместе с другими режимами («wb», «rb», «ab»);

5) «+» – файл открывается одновременно на чтение и запись. Применяется вместе с другими режимами («r+», «a+», «wb+» и т.д.).

Стоит выделить комбинации «r+» и «w+» при таких режимах, оба файла будут открыты на одновременное чтение и запись, но в первом случае при отсутствии файла возникнет исключение, а во втором создастся новый файл.

Как было отмечено ранее, для работы с файлами можно использовать встроенные функции open() и close() и конструкцию with. В первом случае стоит учитывать, что во время работы с файлами могут быть сгенерированы различные исключения, что может привести к не выполнению функции close() и соответственно не закрытию файла. Для предотвращения данной ситуации применяется конструкция try, где в блоке finally обязательно вызовется функция close(). Однако, чтобы избежать данных дополнительных действий существует конструкция with, которая внутри себя уже реализует обработку исключений и обязательного закрытия файла.

**Синтаксический анализ**

Часто приходится из текстового документа выделять конкретную информацию для её организации, упорядочивания и представления в удобной и читаемой форме, но при этом, не работая с ресурсом вручную. Как правило, это большие текстовые данные, которые тяжело и долго обрабатывать вручную, однако это может быть и обработка обновляемой информации (например, расписания занятий с сайта). Для этого применяют синтаксический анализ.

Синтаксический анализ (СА) – это процесс анализа информации и сопоставление последовательности токенов (лексем) естественного или формального языка с его формальной грамматикой. Как правило, СА работает вместе с лексическим анализом.

Лексический анализ (ЛА) – это процесс аналитического разбора входных символов и разбивка их на структурные единицы языка, называемые токенами или лексемами.

**Формат JSON**

Json (JavaScript Object Notation) – это простой формат обмена данными. Он прост для чтения и записи как человеком, так и программными средствами. Файлы, содержащие Json, имеют расширение .json. Json основан на двух структурах данных:

− коллекция пар ключ-значение. В Python такой структурой называется словарь;

− упорядоченный список значений. В Python – список. На сегодняшний день Json является самым популярным форматом передачи данных. Стандарт принят и поддерживается всеми крупным компаниями. Однако также используются и другие форматы передачи данных, такие как XML, YAML и другие. Но они сейчас не так популярны. В настоящий момент почти все приложения, которые передают данные через Интернет, делают это с помощью Json.

**Модуль Json**

− dump(obj, file) – метод предназначен для процесса сериализации (преобразование) объекта obj в формат Json и записи его в файл file;

− dumps(obj) – метод предназначен для процесса сериализации (преобразование) объекта obj в формат Json в виде строки;

− load(file) – метод предназначен для процесса десериализации (обратного преобразования) файла file, содержащего Json в объект Python;

− loads(str) – метод предназначен для процесса десериализации Json строки str в объект Python.

Таким образом, с помощью использования СА и Json можно обрабатывать различные веб-ресурсы и данные, а затем в удобной форме передавать их через Интернет. Такой работы придерживаются почти все современные приложения с многоуровневой архитектурой. Например, при разработке мобильных приложений, когда СА работает на стороне сервера, а мобильное приложение в ответ на запрос о получении данных получает их в формате Json, который поддерживается почти всеми современными языками программирования.

**Задание.** **Вариант 10.**

Написать функцию, которая принимает путь к HTML и путь к CSS файлам и возвращает словарь, в котором ключами выступают теги, идентификаторы или классы в файле CSS, а значениями список списков, где первым элементом внутреннего списка будет наименование тега, которые попадают под стили, указанные в файле CSS, а вторым – номер строк, в которых они находятся. Например, {'#inline-text': [[‘h1’, 29], [‘p’, 50]]}.

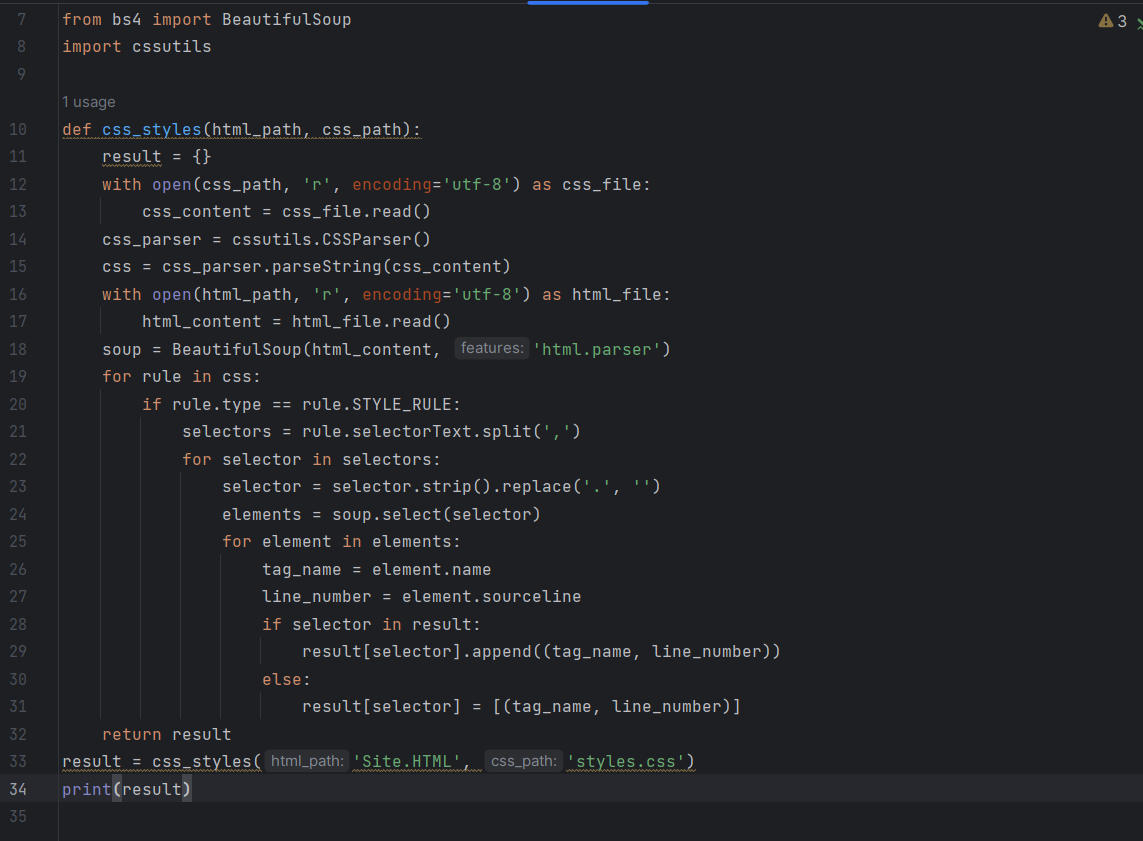


Рисунок 5.1 – код программы.

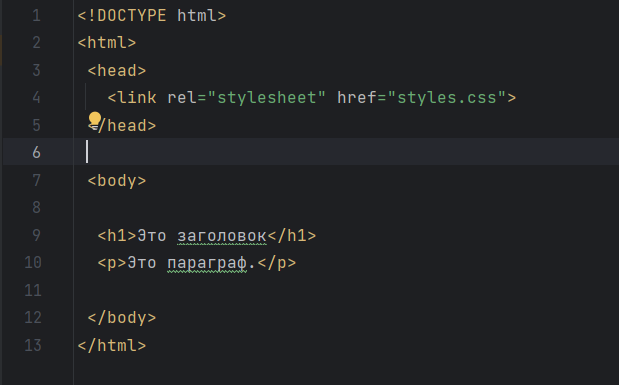


Рисунок 5.2 – код HTML файла.

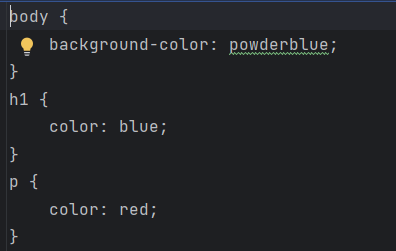


Рисунок 5.2 – код CSS файла.

Результат работы программы представлен на рисунке 5.4

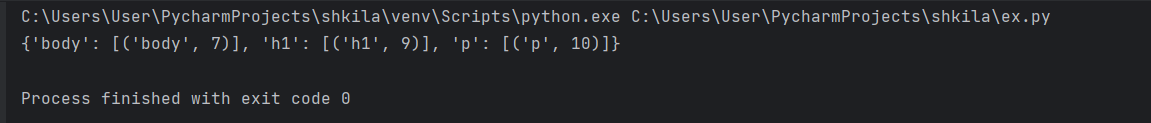


Рисунок 5.4 – результат работы программы

**Вывод**: в результате проделанной работы изучили работу с файлами с помощью функций из стандартной библиотеки; рассмотрели понятие синтаксического анализа текста и ознакомиться с его применением на языке Python; изучили возможности взаимодействия Python с форматом хранения данных JSON.

|  |  |
| --- | --- |
| Выполнили | Рогов М.А. |
| Проверил | Елкин Н.С. |