УДК 004.422.833-047.44

Студ. К.А. Платонова
Науч. рук. преп.-стаж. А.В. Кизино
(кафедра информатики и веб-дизайна, БГТУ)

СПОСОБЫ АНАЛИЗА ДАННЫХ С ПОМОЩЬЮ МОДУЛЯ BEAUTIFUL SOUP

Модули в языке программирования Python предназначены для структурирования и упрощения кода. Одним из модулей, упрощающих извлечение и анализ данных является библиотека BeautifulSoup4 для извлечения данных из файлов HTML и XML.

В первую очередь работа библиотеки зависит от парсера – программы для сбора и систематизации данных. Встроенный в Python парсер html.parser – самый простой и быстрый, не требующий дополнительных установок, однако он может не справляться с некоторыми нестандартными HTML. Парсер html5lib, в свою очередь, обеспечивает полную совместимость с HTML5 и корректнее обрабатывает сложные структуры, но его производительность ниже. Парсер lxml быстро работает и имеет лёгкий синтаксис, что делает его мощным инструментом для обработки больших объёмов данных, но для его использования требуется установка дополнительных библиотек.

Для того, чтобы осуществлять работу с полученным деревом объектов, библиотека предоставляет четыре вида объекта: tag (соответствует тегу в исходном документе), NavigableString (извлекает текстовые данные между тегами), comment (извлекает закомментированный текст) и BeautifulSoup (конвертация документа). Каждый из этих объектов имеет свои параметры, расширяющие функционал.

Для поиска и навигации по элементам документа в BeautifulSoup можно использовать обращение по имени (.title, .section), а также метод find_all() для получения всех вхождений. При этом важно учитывать, что выбор метода поиска может существенно влиять на производительность: find_all() может быть более затратным по времени, особенно в больших документах, чем прямое обращение к элементу.

После нахождения элемента доступно множество методов для его модификации. Например, атрибуты можно изменять с помощью метода .attrs, а текст – через свойства .string или метод .insert(). В отличие от более сложных библиотек, таких как lxml, BeautifulSoup предлагает более интуитивно понятный интерфейс для таких операций. Удаление элементов с помощью метода .decompose() также демонстрирует простоту и эффективность работы с документом.