Описание программы

Задача выполнена на Python 3.7.0 с использованием лишь стандартных его библиотек.

В ходе решения был определен класс Article с двумя полями: title и paragraphs. Для него определены следующие методы:

- AddTitle(self, title)
- AddParagraph(self, paragraph) - к существующему списку параграфов добавляется новый
- FindUrl(self, paragraph) - форматируются ссылки:
  1. Создание пустой строки (новый параграф)
  2. Поиск последовательности символов ‘<a ’
  3. Если такая последовательность существует (есть ссылка), то в новый параграф добавляются все символы из paragraph до символа, предшествующего ‘<’. Также добавляются элементы paragraph между первыми двумя символами ‘кавычки’(url) и сам текст текущего тега.
  4. Повтор шага 3, пока не закончатся все ссылки.
  5. В новый параграф добавляются оставшиеся после последней ссылки элементы paragraph. (В случае, если ссылки отсутствуют, добавляется весь paragraph)
- ParseArticle(self, html) - Записывает список параграфов статьи всё содержимое непустых тегов <p>

  /*(Множество сайтов новостей (например: lenta.ru, mail.ru) используют этот тег в оформлении основной статьи; без использования сторонних библиотек, таких как BeautifulSoup и lxml самый простой способ получить желаемый результат)*/
- WriteArticle(self, html) - Записывает все параграфы в файл ‘index.html’. Если в строке количество символов превышает 80, вместо пробела записывает переход на новую строку.

Программа получает в качестве параметра ссылку, по которой получает html код страницы. Определяет новый объект **article**, и вызывает его метод **WriteArticle**.

Результаты работы

url	filename
https://lenta.ru/news/2019/06/04/rubyprotection/	index1.txt
https://auto.mail.ru/article/73124-v_rabote_gibdd_snova_proizoshel_sboi_po_vsei_rossii/	index2.txt

Выводы

К сожалению, данный опыт показал, что ошибочно было выбрать направление, где не используются уже существующие библиотеки для обработки html.

При наличии должного количества времени, я бы попробовал построить дерево из тегов страницы, вес каждой ветви - количество символов в параметрах **text** тегов, вложенных в них, и пытаться проанализировать эти результаты.