Алгоритм
Получаем страницу
Получаем элементы в которых содержится полезная информация
Обрабатываем информацию, приводим к удобному виду
Записываем в файл

Исходный код

```
# -*- coding: utf-8 -*-
import os
import sys
import requests
import re
from bs4 import BeautifulSoup
import textwrap


class Parser(object):
    """docstring for Parser"""

    def get_url(self):
        try:
            sys.argv[1]
        except Exception, e:
            raise e
        else:
            return str(sys.argv[1])

    def parse(self):
        """
        Метод для получения важной информации на странице
        Returns:
            List: Список найденных тегов в которых хранится информация
        """
        url = self.get_url()
        request = requests.get(url)
        soup = BeautifulSoup(request.content)
        # удаляем inline стили и скрипты
        for tag in soup.findAll(['script', 'style']):
            for code in tag:
                code.extract()
        # заменяем html пробелы
        soup.prettify(formatter=lambda s: s.replace(u'\xa0', ' '))
        tags = soup.findAll(["h1", "h2", "p"])
        self.write_to_file(tags)
        return tags

    def write_to_file(self, data):
        """
        Метод обработки и записи полученных данных в файл
        Args:
            data (int): список найденных тегов
        Returns:
            bool: True при успешной записи в файл
        """
        url = self.get_url()
        path_to_file = url[re.match(r'http(s?)\://', url).end():-1]
        parts = path_to_file.split('/')
        path_to_dir = ''.join([s + '/' for s in parts[:-1]])
        if not os.path.exists(path_to_dir):
            os.makedirs(path_to_dir)

        f = open(''.join([path_to_file,'.txt']), 'w')
        for i in data:
            for link in i.findAll('a'):
                if link:
                    href = link.get('href')
                    link.replaceWith(''.join([link.text, ' [', href, ']']))

            text = i.text.encode('utf8')
```

```
            text = textwrap.wrap(text, 140)

            for s in text:
                f.write(s)
                f.write('\n')
            f.write('\n\n')
        f.close()

        return True

if __name__ == '__main__':
    parser = Parser()
    parser.parse()
```