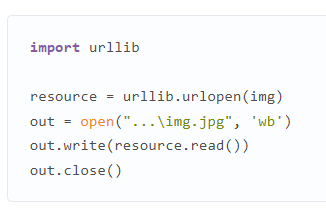
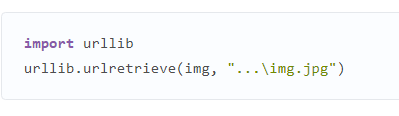
**Тема занятия №23: Сетевое программирование и веб-запросы. Парсинг данных**

**1. Загрузка картинок**

Первый метод использует модуль urllib (или же urllib2). Пусть имеется ссылка на некое изображение img. Метод выглядит следующим образом:



Второй метод использует тот же самый urllib. В дальнейшем будет показано, что этот метод чуть медленнее первого (отрицательный оттенок фактора скорости парсинга неоднозначен), но достоин внимания из-за своей краткости:



Третий метод использует модуль requests. Метод имеет одинаковый порядок скорости выгрузки картинок с первыми двумя методами:

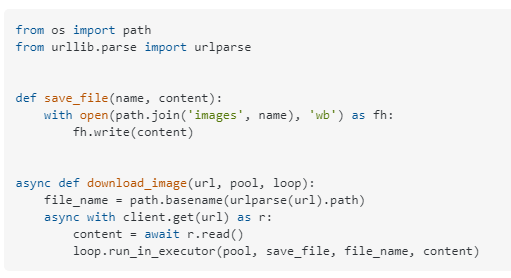


Четвертый метод по скорости кардинально отличается от предыдущих методов (на целый порядок). Основан на использовании модуля httplib2. Выглядит следующим образом:



Здесь явно используется кэширование. Без кэширования (h = httplib2.Http()) метод работает в 6-9 раза медленнее предыдущих аналогов.

Пятый это aiohttp



**2. Загрузка и парсинг html страниц**

Самый простой способ:



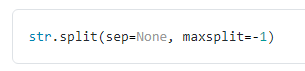
Можно с использованием других библиотек:



Как можно добраться до нужного текста?

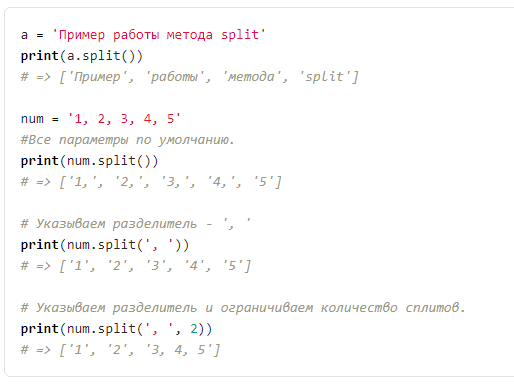
Метод split() в Python разделяет строку на список подстрок по разделителю.

Синтаксис:



sep - необязательный параметр, позволяющий задать разделитель вручную. По умолчанию любой пробел является разделителем.

maxsplit - необязательный параметр, указывающий максимальное количество разбиений, которое требуется выполнить. Если параметр указан,то выполняется не более maxsplit разбиений, то есть итоговый список будет содержать не более maxsplit+1 элементов. Если maxsplit не указан или равен -1, то ограничения на количество сплитов нет.



При этом html страница, в частности статическая, это набор текста, который можно «распилить».

**1. Прямая выгрузка данных с сайта**

Requests — это простая, но элегантная библиотека HTTP.

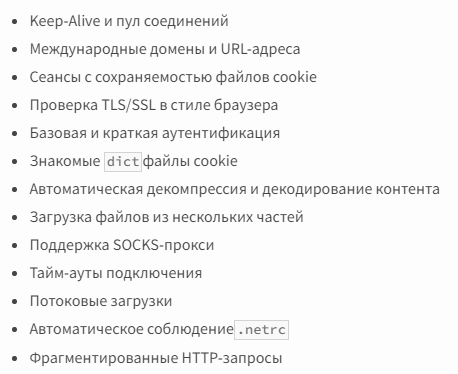


Запросы позволяют очень легко отправлять запросы HTTP/1.1. Нет необходимости вручную добавлять строки запроса к вашим URL-адресам или формировать кодирование ваших PUT& POST данных — но в настоящее время просто используйте этот jsonметод!

Requests — один из самых загружаемых пакетов Python на сегодняшний день. 30M downloads / weekСогласно GitHub, Requests в настоящее время зависит от репозиториев 1,000,000+.

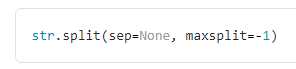
Поддерживаемые функции и лучшие практики

Запросы готовы к требованиям создания надежных и надежных приложений, использующих HTTP, для нужд сегодняшнего дня.



Метод split() в Python разделяет строку на список подстрок по разделителю.

Синтаксис:



Используя библиотеку, получение текста и метод split можно «выгрызать» из сайта необходимые данные. Но при условии что сайт не изменяет свои тэги / стили, иначе код нужно будет дорабатывать.

**2. Использование библиотеки bs4**

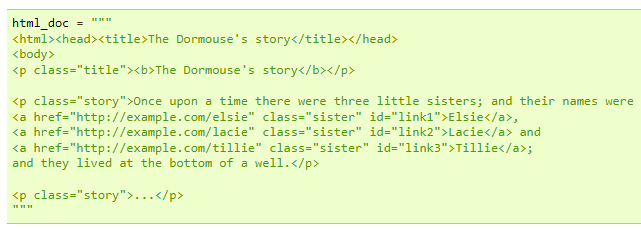
Beautiful Soup — это библиотека Python для извлечения данных из файлов HTML и XML. Она работает с вашим любимым парсером, чтобы дать вам естественные способы навигации, поиска и изменения дерева разбора. Она обычно экономит программистам часы и дни работы.

Эти инструкции иллюстрируют все основные функции Beautiful Soup 4 на примерах. Я покажу вам, для чего нужна библиотека, как она работает, как ее использовать, как заставить ее делать то, что вы хотите, и что нужно делать, когда она не оправдывает ваши ожидания.

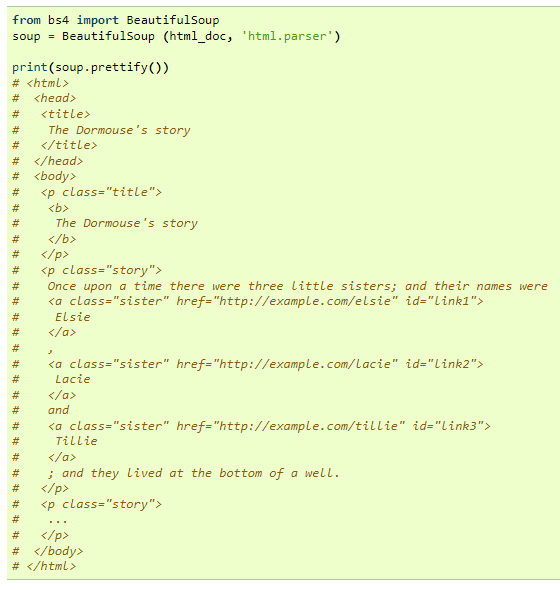
Примеры в этой документации работают одинаково на Python 2.7 и Python 3.2.

Быстрый старт

Вот HTML-документ, который я буду использовать в качестве примера в этой документации. Это фрагмент из «Алисы в стране чудес»:



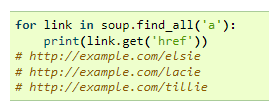
Прогон документа через Beautiful Soup дает нам объект BeautifulSoup, который представляет документ в виде вложенной структуры данных:



Вот несколько простых способов навигации по этой структуре данных:



Одна из распространенных задач — извлечь все URL-адреса, найденные на странице в тегах <a>:



Другая распространенная задача — извлечь весь текст со страницы:

