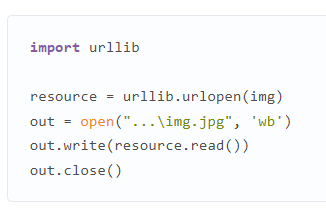
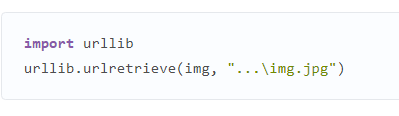
**Тема занятия №28: Сетевое программирование. Библиотеки requests и aiohttp**

**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:

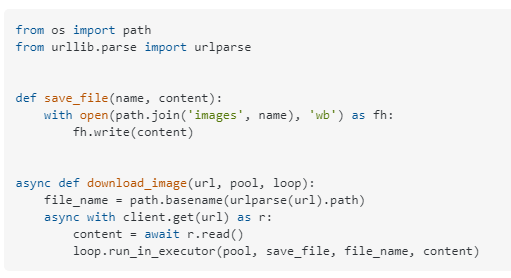


Четвертый метод по скорости кардинально отличается от предыдущих методов (на целый порядок). Основан на использовании модуля httplib2. Выглядит следующим образом:



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:



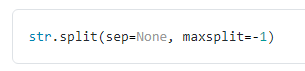
Можно с использованием других библиотек:



Как можно добраться до нужного текста?

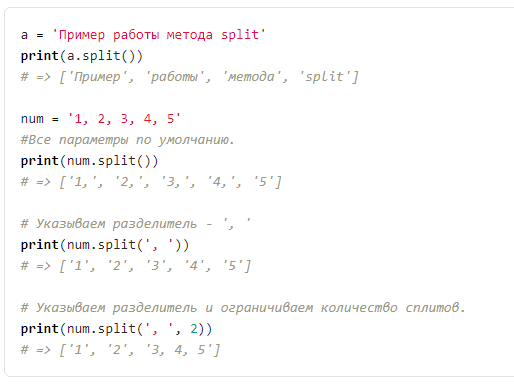
Метод split() в Python разделяет строку на список подстрок по разделителю.

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».