

Тестовое задание

Задача

Вам предстоит собрать информацию с сайта <https://www.petsonic.com>
 Нас интересует та часть сайта, который занимается продажей товаров для животных - <https://www.petsonic.com/snacks-huesos-para-perros/>
 Нужно собрать **все товары** из категории и записать в **csv** файл.

По каждому товару нужно собрать три параметра

1. Полное название товара
2. Цену
3. Изображение

Важно! На случай проблем с доступностью категории, воспользуйтесь альтернативной ссылкой:
<https://www.petsonic.com/hobbit-half/>

Пример разбора

Вот пример страницы товара <https://www.petsonic.com/hobbit-alf-galletas-granja-para-perros.html>



Название

Весовка

Цена

Peso	Precio IVA INCLUIDO	Cantidad
<input checked="" type="radio"/> 200 Gr.	1,50 € 1.35 €	- 1 +
<input type="radio"/> 430 gr	2,60 € 2,34 €	
<input type="radio"/> 4 Kg.	18,70 € 16,83 €	

Precio total: **1.35 €**

AÑADIR AL CARRITO

Такую страницу мы называем страницей мультипродукта. Это означает, что на одной странице находится информация о нескольких разновидностях продукта. В данном случае разная весовка: 200 gr, 430 gr, 4 kg.

С этой страницы в выходной файл должно быть добавлено 3 записи:

Name	Price	Image
Galletas Granja para Perro - 200 gr	1.35	http://www.petsonic.com/5830-large_default/galletas-granja-para-perro.jpg
Galletas Granja para Perro - 430 gr	2.34	http://...
Galletas Granja para Perro - 4 kg	16.83	http://...

На что обратить внимание

1. Название сформировано, как общее название продукта *Galletas Granja para Perro* плюс весовка из таблицы *200 gr*
2. Если у продуктов нет вариаций, то и весовка будет указана только одна. В таком случае с одной страницы продукта получится одна запись в файле
3. Список продуктов категории состоит из нескольких страниц

Требования к оформлению кода

1. Должна **быть написана программа/скрипт на Ruby**. Программа получает на вход:
 - a. ссылка на страницу категории (может передаваться любая категория сайта)
 - b. имя файла в который будет записан результат
2. После отработки скрипта результаты записывает в выходной файл с заданным именем
3. В тестовом задании нужно использовать **xpath** для получения содержимого html элементов, таких как цена, название и т.п. Возможно, понадобится использовать **regex** для того чтобы достать отдельные части данных, если **xpath** недостаточно
4. Для скачивания страниц нужно использовать библиотеку **curb** (*могут быть проблемы под Windows, лучше использовать Linux*), для парсинга страниц **nokogiri**, для записи в csv файл – модуль **csv**
5. Вместе с решением нужно прислать пример запуска скрипта
6. Если в скрипте используются сторонние gem-ы, то приложить Gemfile
7. Скрипт должен работать достаточно быстро и при этом не привлекать к себе внимание администраторов сайта
8. Во время выполнения скрипта должно быть понятно, что он сейчас делает