Создание инструкций для ParseMX

Основная инструкция для обработки сайта состоит из нескольких секций. На этапе текстов достаточно иметь только секцию [PRODUCT] - ведь если разбор страницы продукта налажен – 90% дела уже сделано!

Возьмём для примера простую строку:

name = tag_text “. Product h1” -> y_translate

Тут name – это название переменной. Можно использовать любые свои переменные. В инструкции надо заполнить нужные переменные, что бы данные попали в базу. name – это название товара.

tag_text и y_translate – это команды. y_translate делает перевод текста с помощью Яндекс-переводчика, а tag_text берёт из страницы текст нужного тэга.

“. Product h1” – это параметр команды tag_text. Если команда имеет несколько параметров, они пишутся через запятую, например replace “Model”, “Модель”. Как видно из примера, команды можно применять последовательно с использованием ->

CSS-селекторы

В предыдущей команде есть параметр “. Product h1”. Это CSS-селектор – очень удобный механизм для выбора на странице интересующих тэгов.

В тексте страницы есть теги с параметрами. Их можно увидеть, если нажать на чём-то правую кнопку и выбрать "Просмотр кода страницы" или "Просмотр/анализ (кода) элемента". Например:

Android phone

Тут <a> - это сам тэг, ItemTitle – его класс, title – его id, Android phone – его содержимое.

В селекторах теги пишутся без ничего ("a"), классы начинаются с точки (".ItemTitle") , а id с решетки ("#title").

Чтобы «забрать» какой-то тэг на странице, достаточно указать его **id** (если он есть, то он обычно уникальный) или класс (но надо следить, что бы он был только у нужных тэгов).

Можно "складировать" понятия, перечисляя их через пробел. Например, что бы взять тэг <a> внутри тэга с классом ItemTitle, надо написать: “.itemTitle a”. Если под этот селектор на странице подходит несколько тэгов, то вы получите их все. Кстати не обязательно перечислять полный путь, позиции можно пропускать – главное что бы оно всё еще брало только нужные вам тэги.

Команды

Команды могут применяться на чём-либо с помощью -> :

seo_url = name -> translit

Как уже говорилось, можно использовать -> несколько раз по цепочке. Если команда используется без ->, то она выполняется на материале полученной страницы.

Команды, которые возвращают массив, имеют также одинарную вариацию, которая даст вам только первый элемент из найденных. О работе с массивами будет рассказано чуть дальше.

Серым цветом показаны необязательные параметры.

tags_text ".attributes table tr rd"
Даёт текст всех тэгов, которые подходят под селектор (в виде массива). Одинарная вариация – tag_text

tags_href ".images a", "текст"
Даёт атрибут **href** (ссылки) тэгов, поэтому имеет смысл применять только к тэгам <a>. Если задан второй параметр "текст", то команда даст только те тэги, которые явно содержат внутри текст.

Применяет к результату команду urls, что бы ссылки были правильными. Одинарная вариация – tag_href

tags_attr "#gallery img", "src", "текст"
Даёт атрибут тэгов ("src"). Если задан "текст", то отберёт только тэги с таким текстом внутри. Одинарная вариация – tag_attr

urls
Обрабатывает ссылки, дополняя их, если они указаны на странице относительно (без домена). Во избежание проблем обрабатывайте urls все ссылки, кроме полученых через tags_href. Одинарная вариация - url

tag_html "#ProductDescription"
Даёт весь HTML, который находится внутри тэга. Часто используется для получения описания продукта с форматированием.

translit
Переводит текст в строчку из маленьких английских букв и дефисов. Чаще всего используется для seo_url

y_translate "en-ru"
Переводит текст с помощью Яндекс-Переводчика. Если параметр не задан, то переводит с английского на русский.

replace "найти", "заменить"
Находит все "найти" и заменяет на "заменить". Если не указано "заменить", то убирает все "найти". Поддерживает регулярные выражения.

inside "начало", "конец"
Даёт всё, что находится между "начало" и "конец" . Используется для получения того, что не достаётся с помощью селекторов.

replace_inside "найти", "заменить", "начало", "конец"
Производит замену между "начало" и "конец" .

Что бы в тексте в кавычках **упомянуть переменную**, используйте %:

description = "%name из лучших материалов. Модель %model особенно хороша!"

Работа с массивами

Что обратиться к какому-то элементу массива, используйте arr[15]. Что бы добавить в массив элемент, сделайте arr[] = newelement. Есть несколько команд, которые могут пригодиться при работе с массивами:

- title = **reset** arr – получить первый элемент
- link = **end** arr – получить последний элемент
- images[0] -> **unset** – удалить элемент
- **sort** arr – отсортировать
- arr3 = arr1 -> **array_merge** arr2 – объединить массивы

Кроме этих команд можно использовать все функции PHP. Также можно при необходимости переходить на PHP код – он будет корректно обработан. Посмотреть, что получилось в результате трансляции вашей инструкции в PHP можно в каталоге **ParseMX**.

Переменные продукта

Как уже говорилось, что бы данные попали в базу, в инструкции надо заполнить правильные переменные. В скобках подаются значения по умолчанию. Переменные с **[]** – массивы.

Основные характеристики

name - название товара. Если не заполнено, то товар пропускается.

model (значение **name**) - модель, если товар с такой моделью уже есть в базе, он будет обновлён

product_description - описание товара

price - цена, для перевода можно использовать

-> currency "EUR"
"EUR" - это валюта, с которой надо переводить в основную валюту магазина. Валюта с указанным кодом должна быть установлена в магазине.

category - название категории, куда добавить товар

categories[] - массив категорий. Используйте для добавления в несколько категорий или для отмены категорий, заданных в задаче

manufacturer (заданный в задаче)

main_image, **images**[]

options[][] – опции, масив массивов. Заполнять так: options[“Размер”] = tags_text “#size li”

attributes – аттрибуты, масив массивов. Заполнять так: attributes[“Характеристики”] = “.attrs td”

Массив атрибутов формируется так: первый элемент – это название, второй – значение, и т. д. Можно не выносить группы в отдельные подмассивы. Тогда началом новой группы будут считаться атрибуты без значения (с пустым нечётным элементом).

Есть функции специально для работы с атрибутами. Они работают после того, как массив attributes уже заполнен.

- **attribute** “Название” – даёт значение нужного атрибута
- **take_attribute** “Название” – даёт значение нужного атрибута и убирает его из атрибутов

SEO характеристики

seo_url (или keyword), h1, title, meta_keyword, meta_description, tags

Остальные

Складские параметры: sku, upc, ean, jan, isbn, mpn, location
tax_class_id
quantity (1000)
minimum (1)
subtract (1 - да)
stock_status_id (7 - в наличии)
shipping (1 - да)
date_available (сейчас)
length
width
height
length_class_id (1)
weight
weight_class_id (1)
status (1 - включено)
sort_order (1)
product_store[] (основной магазин)
points

(инструкция ещё не закончена)