

## Управление настройками полей и значений

При выгрузке на любую торговую площадку в профиле настраиваются поля. Поле – это сущность выгрузки, которая принимает в себя одно или несколько значений и, преобразуя полученные данные, передаёт на выгрузку в виде одного поля данных в одном товаре. Таким образом, поле представляет собой контейнер значений, который выгружается как часть товара. Пример полей: идентификатор товара, название товара, изображения, цена, остаток.

Название	Тип	Значение	Доп.
<b>ОСНОВНЫЕ ДАННЫЕ О ТОВАРАХ</b>			
Идентификатор товара (артикул) * <small>offer_id</small>	Простое	Поле/свойство ID	... ⚙️ + ⚙️
Название товара * <small>name</small>	Простое	Поле/свойство Название	... ⚙️ + ⚙️
Изображения * <small>images</small>	Простое	Поле/свойство Детальная картинка	... ⚙️ + × ⚙️
		Поле/свойство Картинки галереи [13, MORE_PHOTO, F...	... ⚙️ + × ⚙️

Поле может быть одного из четырёх типов - простое, условное, мультиусловное, РНР-код. В большинстве случаев используется простое поле. Для проверки условий и вывода в зависимости от результата – условное и мультиусловное (последнее – это поле с произвольным количеством условий типа если1-то-если2-то-...-иначе). Поле с РНР-кодом используется для сложных случаев, когда необходимо выгружать собственные значения по произвольной логике, здесь обычно используют собственные функции и классы для расчёта сложных значений, например, при корректировках цен в зависимости от региона и в зависимости от торговой площадки.

Каждое поле имеет свои настройки – на скриншоте выше это кнопка-шестерёнка с самого правого края в каждом из трёх представленных полей. Но прежде чем говорить о настройках, необходимо выяснить почему в каждом поле есть две (или даже больше) кнопки настроек – одна слева от кнопки плюса, другая – справа.

## Значение и поле – в чём разница?

Поле – это контейнер значений, т.е. одно или несколько собранных воедино значений. Значение – это часть поля, которое можно заполнить лишь одним свойством товара или текстовой строкой. Собственно, это всё что требуется знать о структуре значений и полей. Остаётся только понять некоторые нюансы. Коротко их можно описать так: все значения собираются в поле, которое их обрабатывает и только потом отправляет на выгрузку как часть товара (при выгрузке одного товара выполняется поочерёдная выгрузка нескольких полей, в результате чего строится полный набор данных об одном товаре, который в дальнейшем выгружается).

Теперь опишем это на примере из вышеприведённого скриншота. Возьмём для примера поле «Изображения». Оно выгрузится в товаре как элемент **images** (на Ozon это массив изображений). В этом элементе мы хотим выгрузить все картинки, которые есть в товаре: детальная картинка и все дополнительные. Мы просто заполняем первое значение – выбираем «Детальная картинка», добавляем второе значение «Картинки галереи» (из свойства), и получаем что в одном поле – два значения. В результате при выгрузке поля **images** модуль находит картинки в товаре сразу из двух источников и собирает из них общий массив картинок, который и будет выгружен в **images**. Забегая

вперёд, отметим, что на этот процесс влияют настройки полей и значений, а некоторые из них – влияют особенно сильно, например, настройки множественности.

Название	Тип	Значение	Доп.
<b>ОСНОВНЫЕ ДАННЫЕ О ТОВАРАХ</b>			
Идентификатор товара (артикул) * <small>offer_id</small>	Простое	Поле/свойство ID	... ⚙️ + ⚙️
Название товара * <small>name</small>	Простое	Поле/свойство Название	... ⚙️ + ⚙️
Изображения * <small>images</small>	Простое	Поле/свойство Детальная картинка	... ⚙️ + × ⚙️
		Поле/свойство Картинки галереи [13, MORE_PHOTO, F	... ⚙️ + × ⚙️

Таким образом, разница между значением и полем очевидная: значение (на изображении их два, отмечены зелёной рамкой) – это составная часть поля (отмечено синей рамкой), причём значение не выгружается само по себе, оно передаётся в поле, которое и будет его выгружать, совместно с другими значениями.

Теперь необходимо разобраться с настройками значений и полей.

## Настройки значений и полей – какие использовать?

Мы уже описали, что при выгрузке каждого товара модуль последовательно обрабатывает каждое поле, а при выгрузке каждого поля последовательно обрабатывает входящие в него значения.

Таким образом, получается следующая цепочка: каждое значение обрабатывается отдельно, затем все вместе передаются в поле, которое полученный результат обрабатывает и выгружает.

Очевидно, что каждое значение при передаче в поле уже должно быть обработано в соответствии с собственными настройками. Также очевидно, что настройки самого поля могут применяться только после завершения этого этапа, причём применяются эти настройки к общему результату.

Собственно, поэтому в одном поле две (или больше) кнопок-шестерёнок с настройками – по сути, настройки есть у значений, и настройки есть у поля. Набор настроек в них очень похожий, различия небольшие (но принципиальные, они будут описаны ниже). Таким образом, если в поле всего одно значение, то визуально в строке две кнопки настроек – и это может запутать при первом знакомстве.

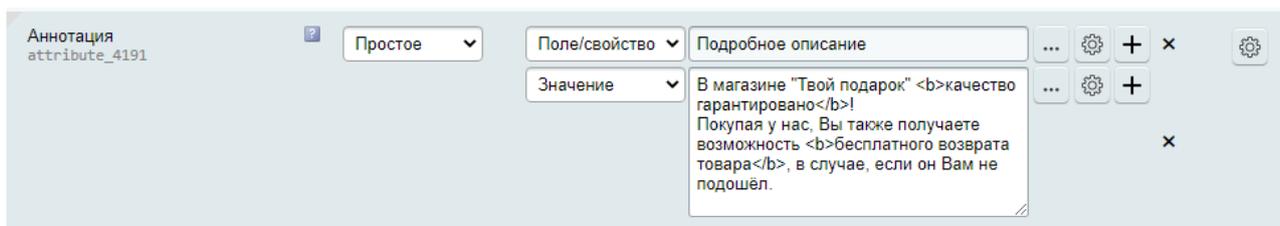
Остаётся понять, в каких случаях использовать настройки значений, а в каких – настройки полей. Главное в этом вопросе – понимать то, о чём было написано выше (сначала применяются настройки для всех значений поочерёдно, затем к общему результату – настройки полей), и понимать задачу, для которой требуются настройки.

Например, некая торговая площадка допускает передавать название не длиннее 50 символов. В настройках есть опция «Ограничение максимальной длины текста» - но где её задать – в значении или в поле? Однозначного ответа нет! Всё зависит от случая. Например, если в поле с названием задано только одно значение, получающее название товара, то нет разницы где задать ограничение текста – если задать в настройках значения, модуль сначала получит название, обрежет его до 50 символов, а потом передаст в поле, которое его так и выгрузит. А можно задать в настройках поля – тогда модуль сначала получит название товара, ничего с ним не сделает, и передаст это значение в поле, которое его обрежет до 50 символов. Результат будет одинаков. Даже если задать одновременно обрезку в обоих местах. Если же в одном поле выгружается два значения, то обрезка через настройки значений

получит в поле по 50 символов из каждого текста, а обрезка через настройки поля – только 50 символов из результата сложения этих строк.

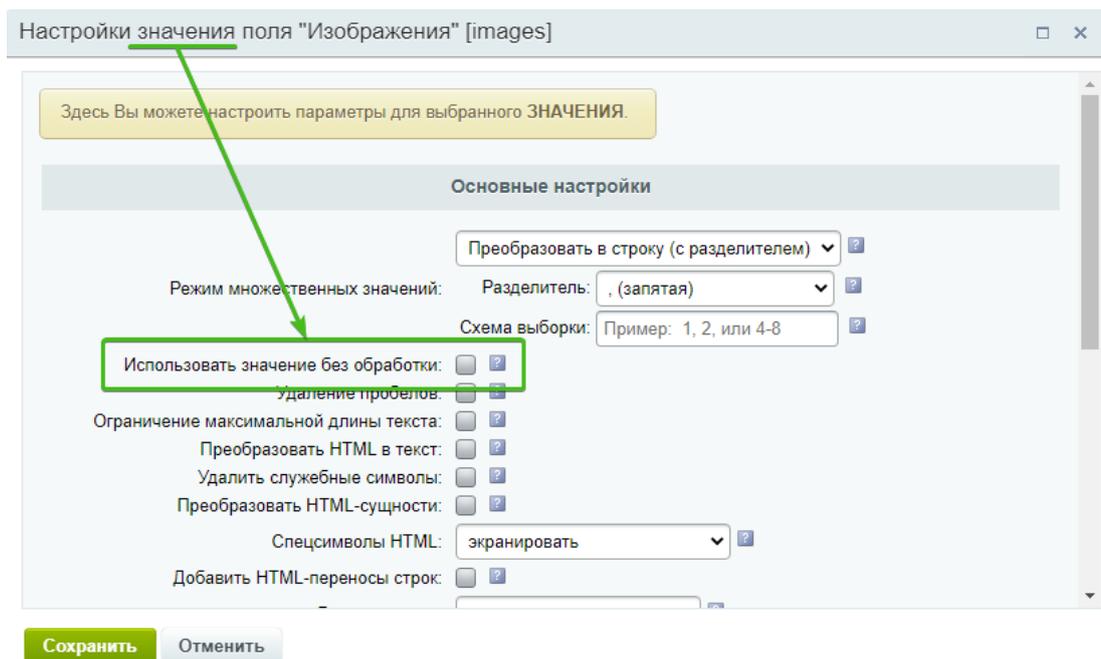
Но есть и другие ситуации (и их, на самом деле, много). Например, если на Вашем сайте из дополнительных картинок первую не нужно выгружать, то в данном случае выход только один – в настройках второго значения установить схему выборки (например, 2-10).

Другой пример: Вы выгружаете описание сразу из двух значений – поле с подробным описанием и простым текстовым значением: такое особенно часто делают при выгрузке, например, на Avito – чтобы дописать в конце объявления свой текст, постоянный для всех товаров.



При таком подходе часто возникает задача внести какие-то изменения в одно значение, не затрагивая другие. Например, если нужно выгрузить в поле аннотации подробное описание в обычном текстовом режиме (без HTML), а в дополнительном тексте HTML-теги оставить, то единственным возможным решением становится применение настройки преобразования HTML в текст только для первого значения.

Есть и особые случаи. Например, если в значении выбрано поле или свойство, которое для выгрузки обрабатывается – например, файл (преобразуется из числа в полный путь к файлу), или свойство типа «Список» (преобразуется из числа в текст), или свойство типа «Справочник» (преобразуется из символьного кода значения в текст) – то в настройках значения появляется галочка «Использовать значение без обработки», это позволяет не преобразовывать исходные значения в текстовые. Это может быть очень полезно при использовании дополнительного PHP-кода для реализации каких-то сложных сценариев (это будет описано ниже). В настройках поля такой опции быть не может, т.к. поле не получает данные из товаров, из которых можно взять чистое значение, оно получает уже обработанные данные из значений.



И следует сказать ещё об одном примере: часто в одно поле выгружается сразу несколько значений (иногда у товара некоторые из них могут быть даже не заполнены) – и если здесь нужно, например, удалить специальные символы, или произвести замены в тексте, то проще один раз указать это в настройках поля, чем отдельно указывать для каждого значения.

Отметим, что на практике в большинстве случаев оказывается, что указывать настройки можно либо в поле, либо в значении, т.к. большинство случаев – не сложные.

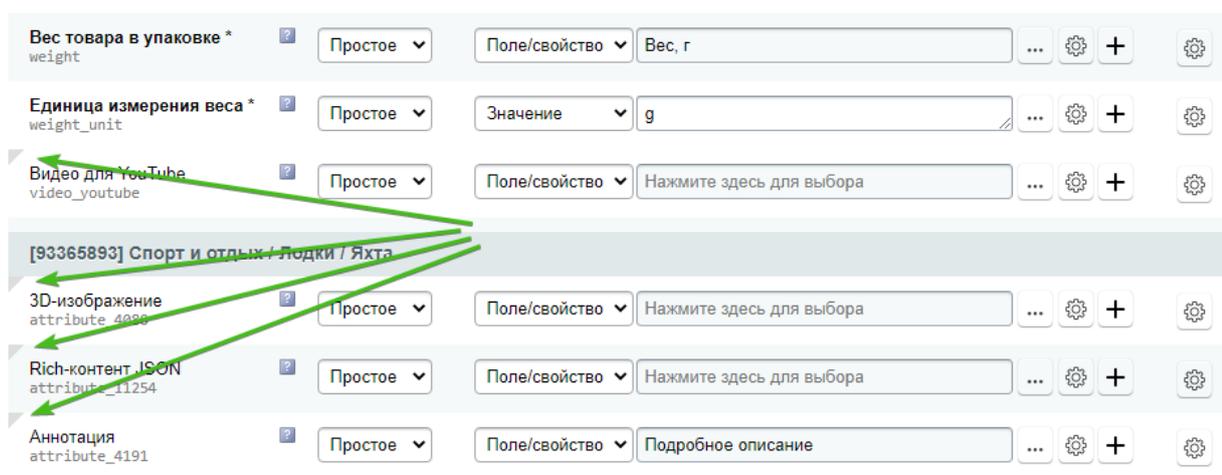
Подводя итог, повторим: настройки применяются сначала для каждого значения (настройки значения), а потом к их общему результату (настройки поля). Исходя из этого и нужно продумывать, где указывать настройки в каждом конкретном случае.

Теперь рассмотрим несколько примеров, основываясь на частых вопросах от пользователей.

## Описание настроек значений и полей

### Режим множественных значений

Одна из наиболее часто используемых настроек, и она имеет особое значение для выгрузки в целом. Прежде всего, необходимо отметить, что некоторые поля поддерживают выгрузку в множественном виде, такие поля отмечены серым значком треугольника в верхнем левом углу.



В некоторых форматах выгрузки (напр., в произвольном формате Excel или в произвольном формате CSV) множественных полей нет, т.к. фактически это ячейки таблицы в одном измерении, поэтому здесь если и можно говорить о множественности – только о перечислении значений (напр., через запятую, или каждое значение – на отдельной строке).

Ниже в таблице приведено сравнение простых и множественных значений для различных базовых форматов выгрузки.

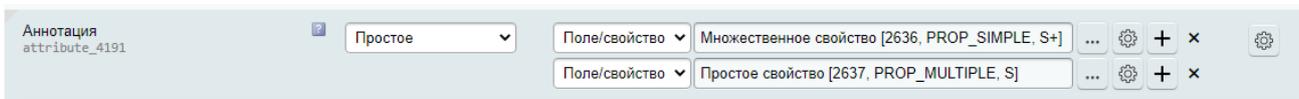
	Простое	Множественное
Формат XML	<phone>+79999999999</phone>	<phone>+79999999999</phone> <phone>+79998888888</phone> <phone>+79997777777</phone>
Формат JSON	{ "phone": "+79999999999" }	{ "phone": [ "+79999999999", "+79998888888", "+79997777777" ] }

Формат Excel	+79999999999	Не поддерживается.
Формат CSV	+79999999999	Значение не может быть множественным, но в одном значении могут быть несколько через запятую или любой другой разделитель.

Вариант «Оставить множественным» (как в значении, так и в поле) доступен только для множественных полей. Но, учитывая все возможные нюансы, необходимо объяснить, как это работает в целом.

Напомним, что в каждом поле сначала обрабатывается каждое из значений (со своими настройками), только затем применяются настройки поля. Каждое значение в составе поля может возвращать либо строку, либо массив (только для множественных). И само поле также может возвращать как строку, так и массив.

Чтобы не описывать этот простой вопрос страницами текста, приведём таблицу различных вариантов, чтобы Вы самостоятельно поняли разницу. В качестве примера возьмём такое поле:



В товаре для примера данные свойства имеют следующие значения:

**Значения свойств:**

Множественное свойство: 2636, PROP\_SIMPLE

A

B

Добавить

Простое свойство: 2637, PROP\_MULTIPLE

C

В таблице будем использовать сокращение «множ.» для варианта «Оставить множественным», и «прост.» для «Преобразовать в строку (с разделителем)». При этом, для наглядности, разделитель в значениях – запятая, в поле – точка с запятой. Колонка «Значение» - это вариант настройки сразу для двух значений, «Поле» - для поля.

Значение	Поле	Результат	Объяснение
прост.	прост.	<pre>"values": [   {     "value": "A,B;C"   } ]</pre>	Сначала во множ. значении происходит объединение через запятую, и потом полученный результат объединяется в поле с простым свойством через точку с запятой.
прост.	множ.	<pre>"values": [   {     "value": "A,B"   },   {     "value": "C"   } ]</pre>	Во множ. свойстве происходит объединение через запятую, а поле выгружает это как отдельные элементы.
множ.	прост.	<pre>"values": [   {     "value": "A;B;C"   } ]</pre>	Множ. свойство выгружает свойство как массив, поле получает этот массив, объединяет с простым, получает массив из трёх элементов, выгружает их через точку с запятой.

МНОЖ.	МНОЖ.	<pre>"values": [   {     "value": "A"   },   {     "value": "B"   },   {     "value": "C"   } ]</pre>	Множ. свойство выгружает свойство как массив, поле получает этот массив, объединяет с простым, получает массив из трёх элементов, и выгружает данные именно как массив трёх элементов.
-------	-------	---	--

Надеемся, внимательное изучение данной таблицы ответит на все возникающие вопросы. Если нет – пожалуйста, обращайтесь к нам в [техподдержку](#).

Также, обратим внимание на два дополнительных параметра в этой опции – «схема выборки» (мы об этом упоминали ранее), позволяющая из полученного массива значений выбрать значения под заданными номерами или интервалами (напр., если нужно выгрузить изображения только со 2го по 10е: «2-10»), и параметр «Разбивка», позволяющий разбивать текстовое значение во множественное (этот параметр доступен только для множественных полей профиля, помеченных серым треугольником в верхнем левом углу). Разбивка может быть очень полезна в случаях, когда, например, свойство в товаре имеет вид нескольких значений, перечисленных через запятую. Например, на сайтах иногда свойство «Материал» указано как «Хлопок, шерсть, полиэстер». Выборка позволяет это значение выгружать не через запятую, а отдельными значениями. Т.е. это действие, обратное действию «Преобразовать в строку (с разделителем)».

## Замена спецсимволов

Одна из самых часто используемых настроек значений и полей – «Спецсимволы HTML». Эта настройка позволяет заменить особые HTML-символы, которые используются повсеместно, но которые нельзя передавать в XML (т.к. XML – это строгий формат, и он при наличии таких символов гарантированно «сломается»). К таким символам относятся: символы кавычек (одинарные и двойные), стрелка влево – знак «меньше» (<), стрелка вправо – знак «больше» (>), амперсанд (&).

В связи с тем, что большинство форматов выгрузки имеет XML-тип, по умолчанию спецсимволы экранируются (выбрано значение «экранировать»):

The screenshot shows a settings panel with various options. The 'Special HTML characters' option is highlighted with a green box and is set to 'экранировать' (escape). Other options include 'Использовать значение без обработки', 'Удаление пробелов', 'Ограничение максимальной длины текста', 'Преобразовать HTML в текст', 'Удалить служебные символы', 'Преобразовать HTML-сущности', 'Добавить HTML-переносы строк', 'Регистр текста', 'Дополнительные настройки', 'URL-кодирование строки', 'Использовать как математическую формулу', 'Округление', 'Изменить формат даты', and 'Добавить ведущие нули'.

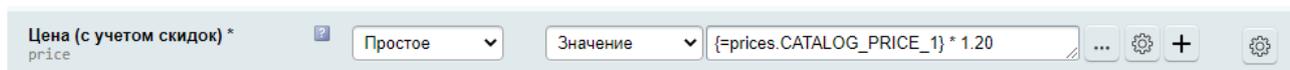
Это означает, что в тексте будут произведены замены:

- двойные кавычки ( " ) будут заменены на &quot;;
- одинарные кавычки ( ' ) будут заменены на &apos;;
- стрелка влево ( < ) будет заменена на &lt;;
- стрелка вправо ( > ) будет заменена на &gt;;
- амперсанд ( & ) будет заменён на &amp;;

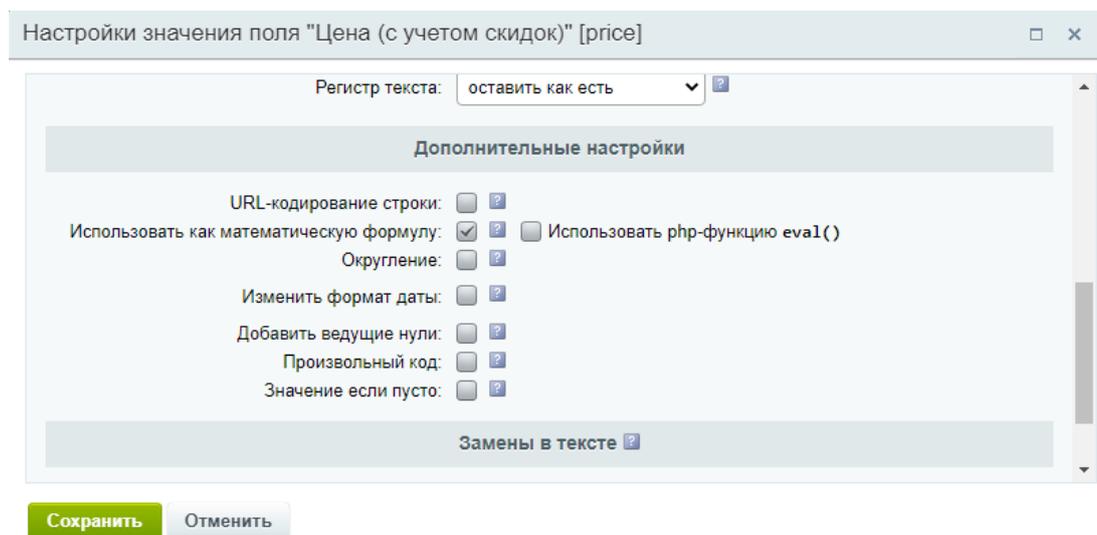
Но в некоторых случаях этого не требуется: и возникает вопрос, где же отключить этот параметр? Ответ такой: и там, и там. Т.е. и в настройках значения, и в настройках поля. Почему так? Да потому что, как мы описали выше, алгоритм работы такой, что сначала обрабатывается каждое значение, потом поле целиком. И если хотя бы в одном месте настройка экранирования остаётся – общий результат будет с заменёнными спецсимволами. Поэтому, **выключать замену спецсимволов через настройку «Спецсимволы HTML» нужно одновременно в настройках значения и в настройках поля.** Выключить экранирование – значит в данной настройке выбрать значение «Оставить как есть».

## Увеличить цену выгрузки на 20%

Для изменения числовых значений, например, увеличения или уменьшения цен, можно использовать опцию «Использовать как математическую формулу». Тип значения при этом должен быть «Значение», в нём следует выбрать необходимое свойство или цену (нажать кнопку с тремя точками, после выбора код обозначения будет добавлен в текстовое поле), и, используя арифметические знаки, построить формулу, например:

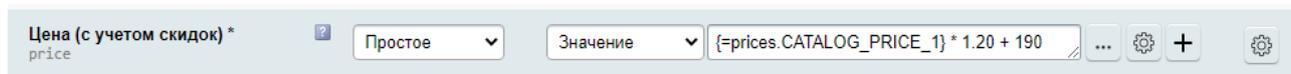


При этом в настройках следует отметить галочку «Использовать как математическую формулу»:



Если у Вас возник вопрос «а где указывать – в настройках значения или поля», перечитайте информацию выше, т.к. однозначного ответа нет, всё зависит от конкретного случая: если у Вас здесь в поле выгружается только одно поле, то можно указывать либо там, либо там.

Этот пример умножает цену на 1.2, т.е. фактически увеличивает цену на 20 процентов. Аналогично можно использовать и другие действия, например, умножить на 1.2 и прибавить 190:



Цена (с учетом скидок) \* price

Простое

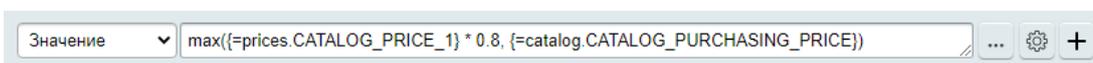
Значение

{=prices.CATALOG\_PRICE\_1} \* 1.20 + 190

Также, можно делать и так: первое значение в поле – выбранное поле, например, цена, второе значение – арифметическое действие, третье – второй операнд, например, текст 1.2. В итоге в поле эта конструкция также поступит в виде формулы.

В формулах можно использовать и более сложные действия, например доступны математические функции: abs [значение по модулю], sqrt [квадратный корень], pow [возведение в степень], round [математическое округление], floor [округление вниз], ceil [округление вверх], max [максимальное из заданных значений], min [минимальное из заданных значение]. Забегая вперед, скажется, что при использовании галочки «Использовать рНР-функцию eval()» нам становится доступен весь арсенал функций РНР, но обычно это не требуется.

В качестве примера можно привести такую формулу:



Значение

max({=prices.CATALOG\_PRICE\_1} \* 0.8, {=catalog.CATALOG\_PURCHASING\_PRICE})

Эта формула позволяет сделать скидку на товары в размере 20% (умножение на 0.8), но только если полученная цена не будет меньше закупочной цены: если полученная цена больше закупочной, то функция max выберет её, иначе будет выбрана закупочная цена. Это позволяет ограничить нижнюю планку цены, т.е. можно уменьшить цену, но не ниже закупочной.

Обратим внимание, что при включении опции «Использовать как математическую формулу» появляется галочка «Использовать рНР-функцию eval()». Многие не понимают, нужно ли её отмечать или нет. В данном случае совет простой – если не знаете для чего это нужно, не включайте. Опишем кратко: без галочки функционал математических преобразований выполняется специальной рНР-библиотекой, которая сама анализирует и считает формулу, библиотека имеет ограниченный набор поддерживаемых действий, сложные операции она может не поддерживать. А при включённой галочке «Использовать рНР-функцию eval» эти действия выполняет сам язык программирования РНР, поэтому никаких ограничений нет – всё что умеет язык программирования РНР, доступно в формулах. Для большинства случаев не будет никакой разницы между включенной и отключенной галочкой. Но т.к. eval – это, фактически, обычное выполнение кода языком программирования, возможности здесь гораздо шире. Это позволяет выполнять не только базовые арифметические операции, но и сложные операции, и даже собственные функции! Например, можно форматировать валюты и числа, получать случайные числа и строки, склонять (при наличии отдельных инструментов) и видоизменять слова, преобразовывать числа в текст, выполнять замены в тексте, и т.д.

При этом, документации по данному вопросу быть не может, т.к. каждый случай уникален, если Вы – программист, Вы и без документации знаете, как использовать eval, а если Вы не программист – то и документация не поможет, в таком случае вопросы нужно решать через нашу [техподдержку](#), мы обязательно поможем.

## Преобразование HTML в текст

Данная опция пригодится в случаях, когда подробное описание задано в формате HTML, а торговая площадка не допускает описание в формате HTML. Необходимо преобразовать форматированный

HTML-в текст обычный текст, при этом хотя бы немного сохраняя его структуру: в частности деление на абзацы, списки и т.п.

Опция имеет несколько режимов:

- Стандартный режим
- Битрикс-режим
- Html2text
- Собственный обработчик

К сожалению, идеального среди них нет, и мы не знаем более подходящих средств для этого, т.к. преобразование HTML в текст – это, с учётом всех нюансов разметки HTML, очень сложная обработка, и часто имеются какие-то особые случаи, с которыми один режим справляется лучше или хуже других.

**Простой режим** – это фактически, только вырезание HTML-тегов PHP-функцией `strip_tags`, с парой незначительных улучшений читабельности.

**Битрикс режим** – использование встроенной в Битрикс функции [HTMLToTxt](#).

**Html2text** – использование сторонней PHP-библиотеки [Html2text](#) для преобразования HTML в текст.

**Собственный обработчик** – вариант для программирования собственного обработчика преобразования, только для программистов.

Повторим, что идеального среди них нет, поэтому рекомендуем протестировать их все три, и выбрать наиболее подходящий для каждого конкретного случая.

## Удаление служебных символов

Данная опция предназначена для случаев, когда при выгрузке в XML открытие полученного файла сообщает об ошибке из-за наличия неподдерживаемых символов. Дело в том, что в тексте (например, в подробном описании), могут быть служебные (управляющие) символы, многие из которых недопустимы при выгрузке в XML. Наиболее известные из таких символов – символ новой строки, символ возврата каретки, символ табуляции – как раз обычно не требуют удаления (но можно их удалить, для каких-то особых случаев), а вот почти все остальные обязательны к удалению при выгрузке в XML (при выгрузке в другие форматы требования могут быть индивидуальны).

При этом важно не путать служебные символы, удаляемые данной опцией со специальными символами («»>()[]!~?+\_\*& и т.д.) и с HTML-сущностями (к примеру, `&nbsp;` и `&quot;`). Для справки, приведём [ссылку на таблицу символов](#) (первые 32 символа из таблицы ASCII).

Опция также имеет параметры для удаления и пробела, но необходимо понимать, что это удалит все пробелы в строке, это не подходит для названий и описаний, но может подойти для некоторых других выгружаемых данных. В общем наборе опций также есть и другая – «Удаление пробелов», которая позволяет выполнять удаление незначащих пробелов в конце и/или начале значений (на языке программистов это называется `trim`).

## Преобразование HTML-сущностей

Не путать с опциями «Удалить служебные символы» и «Спецсимволы HTML»! Данная опция преобразует HTML-сущности в соответствующие им привычные для нас символы, например:

- &quot; - в двойную кавычку "
- &nbsp; - в неразрывный пробел
- &laquo; - в символ левой кавычки-стрелки «
- &raquo; - в символ правой кавычки-стрелки »
- &copy; - в символ копирайта ©
- и т.д.

Полный список HTML-сущностей можно посмотреть по [ссылке](#).

При использовании опции также появляется вложенная опция «Также удалить все символы &» - это добавляет возможность удаления всех лишних (но иногда некоторые такие элементы не лишние!) символов амперсанда, **обычно это не требуется**, но может помочь в некоторых случаях.

## Добавление HTML-переносы строк

Данная опция решает простую, но важную задачу выгрузки текстовых описаний. Она необходима при выгрузке простых текстовых описаний в формат HTML – без этой опции текст в несколько строк будет выгружен, с точки зрения HTML, одной сплошной строкой, т.к. обычные переносы строк для HTML ничего не значат. Опция заменяет простые переносы на HTML-переносы (<br/>), благодаря чему текст будет выгружаться с разделением на строки.

## URL-кодирование строки

Данная опция позволяет выгружать ссылки не в прямом виде, а в кодированном, т.е. с заменой символов, недопустимых для URL. Это сейчас может потребоваться только в очень редких случаях, например, при URL-выгрузке в качестве переменных в адресе.

Например, если на сайте используется ссылка на товар <https://site.ru/catalog/одежда/футболка-поло-123/>, её кодирование даст следующий результат:

```
https%3A%2F%2Fsite.ru%2Fcatalog%2F%D0%BE%D0%B4%D0%B5%D0%B6%D0%B4%D0%B0%2F%D1%84%D1%83%D1%82%D0%B1%D0%BE%D0%BB%D0%BA%D0%B0-%D0%BF%D0%BE%D0%BB%D0%BE-123%2F
```

Повторим, что в последнее время опция практически не находит своего применения.

## Округление чисел

Округление – важная опция, используется довольно часто при выгрузке цен и габаритов. Однако имейте в виду, что для некоторых торговых площадок может быть важно соответствие цен в выгрузке и на сайте с точностью до копеек, т.е. не во всех выгрузках можно округлять цены.

Доступны 4 варианта округления:

1. математически,
2. в меньшую сторону,
3. в большую сторону,
4. по правилам округления.

Последний вариант доступен только в настройках значения (недоступен в настройках поля), и только тогда, когда в значении выбрана цена – тогда будут применяться настройки округления, заданные для этой цены (Магазин – Настройки – Цены – Округление).

Описывать варианты и точность округления не имеет смысла, т.к. здесь никаких особенностей нет, чистая математика. Например, при математическом округлении до десятков мы получим цены кратные десяти, например, 150, 160, 170 и т.д.

## Добавление ведущих нулей

Добавление ведущих нулей – опция, которая позволит выгружать числа с заданной длиной, добавляя в начале незначимые нули. Это может быть удобно (в некоторых случаях) с артикулами выгрузки, построенными на основании ID товара. Например, если ID товаров на сайте небольшие (123 или 4578), можно использовать данную опцию, указав длину строки (число знаков) – например 8. Тогда данные числа будут выгружаться как 00000123 и 00004578.

Для других случаев найти применение данной опции сложно.

## Произвольный код

Возможность использования произвольного кода почти везде, где только можно подумать – часть того уникального, что отличает наш модуль от конкурентов: максимальная гибкость даже в сложных случаях. В нашем модуле практически всегда можно настроить то, что в аналогичных модулях совершенно недоступно. Благодаря этому, модуль позволяет настраивать выгрузки именно так, как требуется Вам, а не только так, как предусмотрели разработчики. Полная свобода действий!

Опция «Произвольный код» в настройках значений и полей позволяет для полученного значения выполнить произвольную обработку PHP-кодом. В некоторых случаях это используется как замена каким-то недостающим опциям. Также может использоваться (наряду с полем типа «PHP-код») для выгрузки сложных значений. Например, если на сайте сложная система расчёта цен по формулам в зависимости от региона и от торговой площадки (хотя это возможно выполнить и несколькими другими способами).

Другой пример использования – вырезание бренда в названиях выгружаемых товаров. С помощью других средств (за исключением поля «PHP-код» и обработчиков) этого сделать нельзя, т.к. бренд для названия каждого товара нужно вырезать именно тот, который указан в свойстве самого товара.

Описывать подробно применение PHP-кода не будем, т.к. это документация не для разработчиков. В подсказке к опции описаны доступные переменные, работая с которыми можно получить не только текущее значение, но и всю необходимую информацию для конкретного товара, инфоблока, профиля, поля.

Отметим, что в связке с данной опцией часто используется опция «Использовать значение без обработки». Опция доступна только для настроек значений и актуальна для тех данных из товара, которые преобразуются в ходе экспорта – напр., числовые ID файлов преобразуются в полные пути к файлам, а числовые привязки к элементам и спискам преобразуются в соответствующие тексты. Использование данной опции вместе с произвольным кодом обеспечивает доступ к «чистому» значению, т.е. обычно к идентификатору. Это требуется, например, если в коде нужно работать именно с ID файла, а не с его адресом.

## Замена пустого значения на собственный текст

Опция «Значение если пусто» позволяет выполнить одну простую задачу: если модуль видит, что для поля ничего не выгружается, он может в поле подставить заданную строку (иначе поле не будет выгружено).

Это может работать, например, со свойством, в котором проставлены только значения «Да», а если не проставлены – чтобы модуль не выгружал пустые значения, можно указать, чтобы выгружал «Нет». Или установка каких-то других дефолтных значений.

## Замены в тексте

Замены в тексте – одна из самых часто используемых опций, которая позволяет в тексте заменять одно на другое, или удалять часть входящих текста. Например, это пригодится, если необходимо заменить название магазина, прописанное в описаниях вручную. Но чаще данный функционал используют для удаления лишней информации из текста. Например, какие-то ненужные символы, или слова. Иногда кавычки-ёлочки заменяют на обычные кавычки.

Также возможно удаление текста с использованием регулярных выражений. Это поможет, например, удалить в тексте все ссылки, отбросить копейки в ценах внутри описаний, или выполнить любые другие сложные текстовые замены. Для построения регулярных выражений, особенно сложных, требуется большой опыт работы с ними, либо просто обратиться к нам в [техподдержку](#).

Другой вариант использования – подобие справочника замен. Например, если некоторые бренды на сайте прописаны не так, как требует выгружаемая площадка. В таком случае приходится добавлять замены для каждого бренда, например, если на сайте задан бренд «Компания Microsoft», а площадка требует просто «Microsoft», можно добавить замену «Компания Microsoft» на «Microsoft».

Также, замена может быть использована для удаления единиц измерения – например, если свойство товара «Вес» (не из каталога, а из инфоблока) выгружается как 2.5 кг, а нужно просто 2.5, это также может пригодиться совместно с использованием формул (использование единиц измерения в формулах недопустим).

Есть также множество других ситуаций, где без замен в тексте практически не обойтись, главное понимать, как это работает, и тогда применение этой возможности обязательно найдётся.

## Различия в наборе опция между различными редакциями модулей

Модуль экспорта на торговые порталы имеет несколько редакций, в которых отличается набор доступных опций, а в некоторых случаях отличаются дополнительные возможности в опциях. Подробная информация о различиях между редакциями, в т.ч. о различиях в опциях, приведена на [нашем сайте](#).