

diafan.CMS version 4.4, <http://cms.diafan.ru/>



# **Полное руководство для разработчика**

# Содержание

## Введение

### I. Шаблоны и шаблонные функции

- [Шаблон сайта](#)
- [Шаблон модуля](#)
- [Общие шаблонные функции](#)
- [Шаблонные функции модулей](#)

### II. Модули

- [Страницы сайта](#)
- [Меню на сайте](#)
- [Новости](#)
- [Магазин](#)
- [Фотогалерея](#)
- [Статьи](#)
- [Файловый архив](#)
- [Теги](#)
- [Вопрос-Ответ](#)
- [Обратная связь](#)
- [Комментарии](#)
- [Форум](#)
- [Опросы](#)
- [Рейтинг](#)
- [Рассылки](#)
- [Пользователи сайта / Регистрация](#)
- [Автообновление](#)
- [Параметры сайта](#)
- [Администраторы](#)
- [Языки сайта](#)
- [Корзина](#)
- [Карта сайта](#)
- [Поиск по сайту](#)
- [Редактирование контента из пользовательской части](#)
- [Прикрепленные изображения](#)
- [Прикрепленные файлы](#)
- [Постраничная навигация](#)
- [Защитный код](#)

### III. Подключения

- [Базы данных](#)
- [Валидация данных](#)
- [Изображения](#)
- [Конфигурация сайта](#)
- [Кэширование](#)
- [Маршрутизатор](#)
- [Общие функции](#)

- [Подключение шаблонов](#)
- [Пользователи](#)
- [Почта](#)
- [Режим разработки](#)
- [Сессии](#)
- [Сжатие страниц](#)

#### [IV. Плагины](#)

#### [V. Архитектура diafan.CMS](#)

- [Файловая структура](#)
- [Пользовательская часть сайта](#)
  - [Инициализация системы](#)
  - [Структура модуля](#)
    - [Контроллер](#)
    - [Модель](#)
    - [Шаблон](#)
    - [Обработка Ajax-запросов](#)
    - [Подключение](#)
    - [Инсталляция](#)
    - [Прямое обращение к файлам модуля](#)
    - [Взаимодействие с другими модулями](#)
  - [Как добавить шаблонный тег](#)
  - [Подключение модуля](#)
- [Административная часть сайта](#)
  - [Инициализация системы](#)
  - [Действия](#)
  - [Структура модуля](#)
    - [Файл администрирования](#)
    - [Настройки модуля](#)
    - [Подключение модуля](#)

#### [VI. Справочник тегов](#)

# Введение

Система управления сайтами diafan.CMS предназначена для создания и управления интернет-проектами.

diafan.CMS позволяет легко и быстро подключать любой дизайн, удобно создавать структуру сайта неограниченной вложенности, наполнять сайт различной информацией любому пользователю, обладающему элементарными навыками по работе с офисными приложениями.

## Терминология

**Пользовательская часть сайта** – общедоступная для всех пользователей часть сайта, доступна по адресу site.ru.

**Административная часть сайта** – часть сайта, предназначенная для редактирования информации пользовательской части. Доступна только для администратора, после ввода логина и пароля. Адрес административной части задается администратором при установке diafan.CMS. Например: <http://www.site.ru/admin/> или <http://www.site.ru/adminka/>.

**Шаблонный тег** – команда для вывода информации в пользовательской части сайта.

**Шаблон** – файл оформления пользовательской части сайта. Это текстовый файл HTML-кода со вставками шаблонных тегов.

**Шаблонная функция** – функция на PHP, выводящая контент в пользовательской части сайта. Запускается с помощью шаблонного тега.

**Модуль** – часть сайта, работающая по определенному алгоритму (Например, модуль «Новости» - вывод группы новостей с датой, анонсом и заголовком). Модуль прикрепляется к одной или нескольким страницам сайта в административной части сайта и выполняет некоторые действия по формированию страницы в пользовательской части сайта.

**Подключение** – набор дополнительных PHP-функций, подключаемых для осуществления определенных целей (работа с файлами, работа с базой данных, работа с изображениями).

**Плагины** – функционал сторонних разработчиков, используемый для выполнения определенных задач.

**Страница сайта** – в пользовательской части часть сайта с уникальным адресом [site.ru/stranica/](http://site.ru/stranica/), оформленная шаблоном дизайна, с заголовком и текстовой информацией. В административной части страницы сайта редактируются в основном модуле «Страницы сайта», к странице может быть подключен один модуль, несколько подключений и шаблонных функций.

# I. Шаблоны и шаблонные функции

## Шаблоны

**Шаблоны** – это файлы оформления сайта.

[Шаблон сайта](#) – это оформление страницы в пользовательской части сайта, дизайн сайта. Шаблоны сайта хранятся в папке *themes*.

[Шаблон модуля](#) — это файл *modules/модуль/модуль.view.php*. Он отвечает за оформление вывода модульной информации.

## Шаблонные функции

**Шаблонные функции** – это функции, вызываемые в шаблонах сайта с помощью [шаблонных тегов](#), для вывода всевозможного контента в пользовательской части сайта. Шаблонные функции делятся на:

- [общие шаблонные функции](#) – это функции из файла *includes/function.php* для общего назначения;
- [шаблонные функции модулей](#) – это функции, хранящиеся в шаблоне модуля *modules/модуль/модуль.view.php*.

# Шаблоны сайта

**Шаблон сайта** – это оформление страницы в пользовательской части сайта, дизайн сайта.

Основной шаблон сайта - *themes/site.php*. По умолчанию все страницы сайта используют этот шаблон. Шаблон представляет собой файл с обычным [HTML](#)-кодом со вставками [шаблонных тегов](#) diafan.CMS.

Совет: при изменении дизайна сайта проще всего взять исходный файл, входящий в пакет diafan.CMS по умолчанию *themes/site.php* и заменить HTML-код, находящийся между тегами: `<body></body>`

## Мультидизайн

diafan.CMS поддерживает мультидизайн сайта. Это означает, что разные страницы пользовательской части сайта могут иметь разный внешний вид.

Чтобы какой-либо странице сайта установить другой дизайн, необходимо в папку *themes* загрузить шаблон с расширением *.php* (например, *site\_shop.php* или *n\_site\_news.php*). Имя шаблона не может содержать никаких символов кроме букв латинского алфавита, точки, тире и нижнего подчеркивания.

Далее в административной части сайта (модуль «Страницы сайта») открыть страницу, для которой необходимо подключить другой дизайн и во вкладке «Дополнительно» в выпадающем списке свойства «Дизайн страницы» выбрать загруженный файл шаблона. «Дизайн страницы» наследуется страницами нижнего уровня по умолчанию.

## Шаблонные теги

Для вывода контента в шаблонах используются шаблонные теги. Шаблонный тег представляет собой тег вида:

```
<insert [name="" ] [module="" ] [value="" ] [attribute_1="" ]  
[attribute2="" ]>
```

Тег включает ряд необязательных атрибутов, делящихся на определенные и неопределенные атрибуты.

Определенные атрибуты:

- **name** – задает вызываемую функцию или выводимую переменную;
- **value, value2, value3, ...** – значение. Номер соответствует номеру языка сайта (если сайт мультиязычный);
- **module** – задает название модуля, к которому относится вызываемая функция.

Неопределенные атрибуты – это все остальные атрибуты. Они передаются функции в качестве массива в формате:

```
array(название => значение, ...);
```

Значения атрибутов должны быть заключены в кавычки и не должны содержать символы `>`, `<`, `"`, `'`. Эти символы должны быть заменены HTML-мнемониками или специальными символами:

```
< – &lt; или [  
> – &gt; или ]
```

& – `&amp;`;  
" – `&quot;`; или `

Шаблонный тег выводит следующий контент:

1. значение объявленных переменных:

— адрес сайта;

Пример: `<insert name="path">`

— адрес сайта, с учетом языка сайта;

Пример: `<insert name="path_url">`

— идентификатор языка сайта;

Пример: `<insert name="language">`

— название сайта на текущем языке;

Пример: `<insert name="title">`

— название модуля, прикрепленного к текущей странице;

Пример: `<insert name="module">`

2. переводимый текст (`value="значение"`, `value2="значение2"`, ...);

Пример: `<insert value="Меню" value2="Menu">`

3. контент, генерируемый [общими шаблонными функциями](#);

Пример: `<insert name="show_year" year="2003">`

4. контент генерируемый [шаблонными функциями модулей](#).

Пример: `<insert name="show_block" module="menu"  
tag_start_1="[span class=`menutd`]" tag_end_1="[/span][br]"  
tag_start_2="[span class=`menutd_level`]" tag_end_2="[/span][br]">`

## Шаблоны модулей

Оформление модуля вынесено в отдельный файл - *modules/модуль/модуль.view.php*.  
Шаблон модуля – это PHP-класс `Модуль_view`. Он включает следующие функции:

- `show_module()` – выводит в пользовательской части контент модуля (если модуль содержит одну логическую часть);
- `form()` – выводит в пользовательской части форму (если модуль интерактивный). Например, в модуле «Вопрос-ответ» форму отправки вопроса;
- `list_()` – выводит в пользовательской части список элементов модуля. Например, в модуле «Новости» - список новостей;
- `first_page()` – выводит главную страницу модуля;
- `id()` – выводит в пользовательской части отдельный элемент. Например, новость в модуле «Новости»;
- `show_block()` – выводит шаблонную функцию модуля. Например, для модуля «Новости», вывод блока новостей для шаблонного тега `show_block`.

В шаблоне могут быть другие функции, отражающие индивидуальные особенности модуля.

## Общие шаблонные функции

**Общие шаблонные функции** – функции из файла *includes/function.php* для общего назначения.

В diafan.CMS определены следующие шаблонные функции:

**show\_body** – вывод основного контента страницы: заголовка (если не запрещен его вывод в настройке странице «Не показывать заголовок»), текста страницы и прикрепленного модуля.

Пример: `<insert name="show_body">`

**show\_previous\_next** – выводит ссылки на предыдущую и последующую страницы.

Пример: `<insert name="show_previous_next">`

CSS-форматирование:

`.previous_next_links{clear: both; padding-top: 10px; height: 40px;}` – ссылки на предыдущую и следующую страницы

`.previous_link{width: 40%; float: left;}` – ссылка на предыдущую страницу

`.next_link{width: 40%; float: right; text-align: right;}` – ссылка на следующую страницу

**show\_links** – вывод ссылок на страницы нижнего уровня, принадлежащие текущей странице.

Пример: `<insert name="show_links">`

CSS-форматирование:

`.show_links{}`

`.show_links li{}`

**show\_language** – выводит ссылки на альтернативные языковые версии сайта (если используется мультиязычность);

Пример: `<insert name="show_language">`

выведет список ссылок на языковые версии сайта, «ru/eng»

CSS-форматирование: `.language{}`

**show\_path** – вывод навигации по сайту «Хлебные крошки». Необязательный атрибут **separator** – разделитель ссылок в навигации.

Пример: `<insert name="show_path">`

выведет ссылки навигации по сайту, «Главная – О компании – Услуги»

CSS-форматирование: `.path{}`

**show\_year** – выводит период функционирования сайта в годах. Начало отсчета от атрибута **year** (по умолчанию текущий год).

Пример: `<insert name="show_year">`

в 2009 году выведет «2009 г.», в 2010 году выведет «2010 г.»

`<insert name="show_year" year="2010">`

в 2010 году выведет «2010 г.», в 2011 году выведет «2010-2011 гг.»

**show\_block** – выводит содержимое страницы сайта, номер которой передан в виде атрибута **id**. Страница должна быть помечена опцией «Блок на сайте» и быть активной.

Пример: `<insert name="show_block" id="7">`

выведет в ячейке таблицы содержимое страницы с id=7, редактируемой в модуле «Страницы сайта».

**show\_title** – выводит заголовок. Используется между тегами `<title></title>` в шапке сайта.

Пример: `<title><insert name="show_title"></title>`

выведет заголовок текущей страницы

**show\_description** – выводит мета-тег description страницы.

Пример: `<description><insert name="show_description"></description>`

выведет описание текущей страницы

**show\_keywords** – выводит ключевые слова страницы. Используется для мета-тега keywords.

Пример: `<keywords><insert name="show_keywords"></keywords>`

выведет ключевые слова текущей страницы

**show\_protect** – выводит JS код запрещающий копирование контента на сайте.

Пример: `<body><insert name="show_protect"></body>`

**show\_js** – подключает JS файлы, обеспечивающие работу diafan.CMS.

Пример: `<insert name="show_js">`

**show\_images** – выводит изображения, прикрепленные к странице сайта (если в конфигурации модуля «Страницы сайта» включен параметр «Использовать изображения»).

Пример: `<insert name="show_images">`

выведет в ячейке таблицы группу фотографий, прикрепленных к текущей странице

**show\_votes** – выводит голосования. Если не задан номер вопроса, то выводится

один или все вопросы (в зависимости от конфигурацией модуля), соответствующие открытой странице, в случайном порядке.

Атрибуты:

— **id** – задает номер вопроса.

Пример: `<insert name="show_votes">`

выведет вопрос, варианты ответа и кнопки «Ответить» и «Результаты», согласно настройкам и заполнению модуля «Опросы» в административной части сайта

**show\_users** – выводит статистику пользователей на сайте.

Пример: `<insert name="show_users">`

выведет «Сейчас на сайте: \*\* гостей, \*\* пользователей.»

CSS-форматирование: `.show_users{}`

**show\_comments** – выводит комментарии, прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» подключены комментарии.

Пример: `<insert name="show_comments">`

выведет форму для отправки комментариев и добавленные комментарии посетителей

**show\_tags** – выводит теги (слова-якори), прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» подключены теги.

Пример: `<insert name="show_tags">`

выведет список ссылок на теги, прикрепленные к текущей странице

**show\_href** – выводит ссылку на страницу сайта. Если текущая страница соответствует адресу, на которую ведет ссылка, то ссылка становится неактивной. Шаблонная функция аналогична стандартной HTML-конструкции `<a href=""></a>`, но ликвидирует ссылки, которые никуда не ведут.

Атрибуты:

- **rewrite** — псевдоссылка страницы, на которую ведет ссылка, например: *news*;
- **img** — адрес изображения, используемого в качестве ссылки, например: *img/logo.gif*;
- **img\_act** — адрес изображения на текущей странице, например: *img/home\_act.gif*;
- **width** — ширина изображения;
- **height** — высота изображения;
- **alt, alt1, ...** — альтернативный тег для изображения или текст ссылки. Номер необходимо заменить номером языка, если на сайте используется мультязычность.

**show\_social\_links** – выводит ссылки на социальные сети.

# Шаблонные функции модулей

## Поиск по сайту

**show\_search** – выводит форму поиска по сайту. Атрибуты: **button**, **button2**, ... – переводимое значение кнопки «Найти».

```
Пример: <insert name="show_search" module="search"
button="найти" button2="search">
```

выведет форму поиска по сайту

Шаблон: функция *show\_search* в файле *modules/search/search.view.php*.

CSS-форматирование:

`.search` – форма поиска

`.search .input_search` – поле для ввода выражения для поиска

`.search .submit_search` – кнопка «Найти»

## Меню на сайте

**show\_block** – вывод меню.

Без атрибутов выводит просто ссылки вида `<a href="...">имя пункта меню</a>`, одну за одной.

Атрибуты:

- **id** – идентификатор категории меню;
- **template** – шаблон тега (функция в файле *modules/menu/menu.view.php* должна называться *show\_block\_template*; по умолчанию значение атрибута *default* шаблон *show\_block*). Если атрибут не задан, меню оформляется с помощью атрибутов оформления и выводится с помощью функции в шаблоне *show\_menu*.

Атрибуты оформления:

- **tag\_start\_номер** – текст, выводимый перед каждой ссылкой меню (может содержать слово *Increment* – при выводе автоматически заменяемое на арифметическую прогрессию, 1, 2, 3, ... и *Level* – заменяемое на номер уровня меню);
- **tag\_end\_номер** – текст, выводимый после каждой ссылке пункта меню (может содержать слово *Increment* - заменяемое на увеличивающийся номер и *Level* – заменяемое на номер уровня меню);
- **tag\_level\_start\_номер** – текст, выводимый перед уровнем (номер) меню;
- **tag\_level\_end\_номер** – текст, выводимый после уровня (номер) меню;
- **tag\_active\_start\_номер** – текст, выводимый перед активным пунктом меню уровня (может содержать слово *Increment* - заменяемое на увеличивающийся номер и *Level* – заменяемое на номер уровня меню);
- **tag\_active\_end\_номер** – текст, выводимый после активного пункта меню уровня (может содержать слово *Increment* - заменяемое на увеличивающийся номер и *Level* – заменяемое на номер уровня меню);
- **tag\_active\_child\_start\_номер** – текст, выводимый перед пунктом меню уровня (номер) с активным дочерним пунктом (может содержать слово *Increment* -

заменяемое на увеличивающийся номер и Level – заменяемое на номер уровня меню);

- **tag\_active\_child\_end\_номер** – текст, выводимый после пункта меню уровня (номер) с активным дочерним пунктом (может содержать слово Increment - заменяемое на увеличивающийся номер и Level – заменяемое на номер уровня меню);
- **separator\_номер** – текст, разделяющий пункты меню.

Номер в названии этих атрибутов обозначает номер уровня. Если атрибуты не указаны для текущего уровня, то они наследуются у верхнего уровня.

Пример: `<insert name="show_block" module="menu">`

выведет просто список ссылок основного меню

```
<table>
  <insert name="show_block" module="menu" id="2"
tag_start_1="[tr][td]" tag_end_1="[/td][/tr]">
</table>
```

сформирует таблицу, в отдельной ячейке которой будут элементы списка меню из второй категории

```
<insert name="show_block" module="menu"
tag_start_1="[div class=`menu`]" tag_end_1="[/div]"
tag_start_2="[div class=`submenu`]" tag_end_2="[/div]">
```

Выведет:

```
<div class="menu"><a href="...">0 нас</a></div>
  <div class="submenu"><a href="...">Животные</a></div>
  <div class="submenu"><a href="...">Миссия</a></div>
  <div class="submenu"><a href="...">История</a></div>
<div class="menu"><a href="...">Фотогалерея</a></div>
```

```
<insert name="show_block" module="menu"
tag_start_1="[div class=`menuLevel`]" tag_end_1="[/div]">
```

Выведет:

```
<div class="menu1"><a href="...">0 нас</a></div>
  <div class="menu2"><a href="...">Животные</a></div>
  <div class="menu2"><a href="...">Миссия</a></div>
  <div class="menu2"><a href="...">История</a></div>
<div class="menu1"><a href="...">Фотогалерея</a></div>
```

Шаблон: функция *show\_block*, *show\_level* (второй уровень) – при оформлении шаблоном, *show\_menu* – при оформлении тегами в файле *modules/menu/menu.view.php*.

CSS-форматирование:

`.menu_block` – блок меню, если используется шаблон по-умолчанию

`.menu_block h2` – заголовок блока меню

`.menu_item` – пункт меню

`.menu_active` – текущий пункт меню

`.menu_active_child` – пункт меню с текущим дочерним пунктом

`.menu_level_HOMEP` – уровень меню HOME

`.menu_item_HOMEP` – пункт меню уровня HOME

## Новости

**show\_block** – выводит последние новости на всех страницах, кроме страницы новостей, когда выводится список тех же новостей, что и в функции.

Атрибуты:

- **count** – количество выводимых новостей (по умолчанию 3);
- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** – категория новостей (id категории, по умолчанию выводятся все новости без учета категории), если в конфигурации модуля отмечено «Использовать категории»;
- **images** – количество фотографий, прикрепленных к новости;
- **template** – шаблон тега (функция в файле *modules/news/news.view.php* должна называться *show\_block\_template*; по умолчанию шаблон *show\_block*).

Формат даты задается в конфигурации модуля.

Пример:

```
<insert name="show_block" module="news">
```

выведет 3 последние новости сайта

```
<insert name="show_block" module="news" count="5">
```

выведет 5 последних новостей сайта

```
<insert name="show_block" module="news" count="1" site_id="12">
```

выведет последнюю новость, прикрепленную к странице 12

Шаблон: функция *show\_block* в файле *modules/news/news.view.php*.

CSS-форматирование:

`.news_block` – шаблонная функция *show\_block*

`.news_block h2` – заголовок

`.news_block .news` – новость в блоке

`.news_block .news_name{}` – название новости  
`.news_block .news_img{}` – изображение новости  
`.news_block .news_date{}` – дата новости  
`.news_block .news_anons{}` – анонс  
`.news_block .show_all{}` – ссылка на все новости

**show\_calendar** – выводит календарь со ссылками на новости за период. Периоды отображаются в виде ссылок на месяцы, только если имеются новости, соответствующие этим периодам.

Атрибуты:

- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** – категория новостей (id категории, по умолчанию учитываются все новости), если в конфигурации модуля отмечено «Использовать категории»;
- **only\_news** – выводит блок только на странице «Новости»;
- **month\_hidden** – если необязательный атрибут равен 1, то выводятся только годы без месяцев;
- **template** – шаблон тега (функция в файле `modules/news/news.view.php` должна называться `show_calendar_template`; по умолчанию шаблон `show_calendar`).

Пример: `<insert name="show_calendar" module="news">`  
выведет календарь новостей

Шаблон: функция `show_calendar` в файле `modules/news/news.view.php`.

CSS-форматирование:

`.news_calendar{}` – шаблонная функция `show_calendar`  
`.news_calendar h2{}` – заголовок  
`.news_calendar .news_year{}` – год  
`.news_calendar .news_year_current{}` – текущий (открытый) год  
`.news_calendar .news_month{}` – месяц  
`.news_calendar .news_month_current{}` – текущий (открытый) месяц  
`.news_calendar .show_all{}` – ссылка на все новости

## Магазин

**show\_block** – выводит несколько товаров из каталога.

Атрибуты:

- **count** – количество выводимых товаров (по умолчанию 3);
- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** – категория товаров (id категории, по умолчанию выводятся все товары без учета категории), если в конфигурации модуля отмечено «Использовать категории»;

- **images** – количество фотографий, прикрепленных к товарам;
- **sort** – сортировка товаров (**rand** – в случайном порядке, **date** – по дате, **price** – по цене). По умолчанию товары сортируются по дате;
- **param** – значения дополнительных характеристик;

Пример:

Товары обладают следующими характеристиками:

- цвет – выпадающий список, номер 3;
- высота – текстовое поле, номер 10;
- наличие аналогов – галочка, номер 16.

Значит значение атрибута `param3=5&param3=6&param10=12м&param16=0` расшифровывается как товары красного и синего цвета (5 и 6 номер), высотой 12м, не имеющие аналогов.

— **template** – шаблон тега (функция в файле `modules/shop/shop.view.php` должна называться `show_block_template`; по умолчанию шаблон `show_block`).

Пример: `<insert name="show_block" module="shop">`

выведет 3 последних товара из магазина

`<insert name="show_block" module="shop" count="5" sort="rand">`

выведет 5 случайных товаров из магазина

`<insert name="show_block" module="shop" sort="price" count="4" cat_id="12">`

выведет 4 самых дешевых товаров из рубрики №12 магазина

Шаблон: функция `show_block` в файле `modules/shop/shop.view.php`.

CSS-форматирование:

`.shop_block{}` – шаблонная функция `show_block`

`.shop_block h2{}` – заголовок

`.shop_block .shop{}` – товар в блоке

`.shop_block .shop_name{}` – название товара

`.shop_block .shop_img{}` – изображение товара

`.shop_block .shop_price{}` – цена товара

`.shop_block .shop_old_price{}` – цена товара без скидки

`.shop_block .shop_discount{}` – скидка товара

`.shop_block .shop_anons{}` – краткое описание товара

`.shop_block .shop_text{}` – полное описание товара

`.shop_block .shop_form` — форма покупки товара

`.shop_block .show_all` — ссылка на все товары

**show\_search** — выводит форму поиска товаров. Если для категорий прикреплены дополнительные характеристики, то поиск по ним производится только на странице категории.

Атрибуты:

- **only\_shop** — выводить форму поиска только на странице магазина;
- **site\_id** — страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** — категория товаров (*id* категории, по умолчанию поиск осуществляется по всем категориям магазина), если в конфигурации модуля отмечено «Использовать категории»; можно указать значение **current**, тогда поиск будет осуществляться по текущей (открытой) категории магазина или по всем категориям, если ни одна категория не открыта;
- **template** — шаблон тега (функция в файле *modules/shop/shop.view.php* должна называться *show\_search\_template*; по умолчанию шаблон *show\_search*).

Пример: `<insert name="show_search" module="shop">`

выведет форму поиска по каталогу товаров

Шаблон: функция *show\_search* в файле *modules/search/search.view.php*.

CSS-форматирование:

`.search` — форма поиска

`.search .input_search` — поле для ввода выражения для поиска

`.search .submit_search` — кнопка «Найти»

**show\_basket** выводит информацию о заказанных товарах, т. н. корзину.

Пример: `<insert name="show_basket" module="basket">`

выведет информацию о корзине

Шаблон: функция *show\_basket* в файле *modules/basket/basket.view.php*.

CSS-форматирование:

`.basket_block` — шаблонная функция *show\_basket*

`.basket_block .basket_count` — количество купленных товаров

`.basket_block .basket_summ` — стоимость товаров

## Фотогалерея

**show\_block** — выводит несколько фотографий.

Атрибуты:

- **count** — количество выводимых фотографий (по умолчанию 3);
- **site\_id** — страница, к которой прикреплен модуль, по умолчанию выбирается

одна страница;

— **cat\_id** – альбом фотографий (id альбома, по умолчанию выводятся все фотографии без учета альбома), если в конфигурации модуля отмечено «Использовать альбомы»;

— **sort** — сортировка фотографий (**rand** — в случайном порядке, **date** — по дате). По умолчанию фотографии сортируются по дате;

— **template** – шаблон тега (функция в файле *modules/photo/photo.view.php* должна называться *show\_block\_template*; по умолчанию шаблон *show\_block*).

Пример: `<insert name="show_block" module="photo">`

выведет 3 последние фотографии из фотогалереи

Шаблон: функция *show\_block* в файле *modules/photo/photo.view.php*.

CSS-форматирование:

`.photo_block{}` – шаблонная функция *show\_block*

`.photo_block h2{}` – заголовок

`.photo_block .photo{}` – фотография в блоке

`.photo_block .photo_name{}` – название фотографии в блоке

`.photo_block .photo_img{}` – изображение фотографии в блоке

`.photo_block .photo_size{}` – размер фотографии в блоке

`.photo_block .photo_anons{}` – краткое описание фотографии в блоке

`.photo_block .show_all{}` – ссылка на все фотографии

### Файловый архив

**show\_block** – выводит несколько файлов из файлового архива.

Атрибуты:

— **count** – количество выводимых файлов (по умолчанию 3);

— **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;

— **cat\_id** – категория файлов (id категории, по умолчанию выводятся все файлы без учета категории), если в конфигурации модуля отмечено «Использовать категории»;

— **images** – количество фотографий, прикрепленных к файлам;

— **sort** – сортировка файлов (**rand** – в случайном порядке, **date** – по дате). По умолчанию файлы сортируются по дате;

— **template** – шаблон тега (функция в файле *modules/files/files.view.php* должна называться *show\_block\_template*; по умолчанию шаблон *show\_block*).

Пример: `<insert name="show_block" module="files">`

выведет 3 последние файла из файлового архива

Шаблон: функция *show\_block* в файле *modules/files/files.view.php*.

CSS-форматирование:

`.files_block{}` – шаблонная функция `show_block`

`.files_block h2{}` – заголовок

`.files_block .files{}` – файл в блоке

`.files_block .files_img{}` – изображение файла

`.files_block .files_name{}` – название файла

`.files_block .files_anons{}` – краткое описание файла

`.files_block .show_all{}` – ссылка на все файлы

## Теги

**show\_block** – выводит список тегов.

```
Пример: <insert name="show_block" module="tags">
```

выведет облако тегов

Шаблон: функция `show_block` в файле `modules/tags/tags.view.php`.

CSS-форматирование:

`.tags_block{}` – список всех тегов

`.tags_block h2{}` – заголовок «Теги»

`.tags_block a{}` – ссылки на теги

`.tags_block span{}` – текущий/активный тег

## Статьи

**show\_block** – выводит последние статьи на всех страницах, кроме страницы статей, когда выводится список тех же статей, что и в функции.

Атрибуты:

— **count** – количество выводимых статей (по умолчанию 3);

— **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;

— **cat\_id** – категория статей (id категории, по умолчанию выводятся все статьи без учета категории), если в конфигурации модуля отмечено «Использовать категории»;

— **images** – количество фотографий, прикрепленных к статьям;

— **sort** – сортировка статей (**rand** – в случайном порядке, **date** – по дате). По умолчанию статьи сортируются по дате;

— **template** – шаблон тега (функция в файле `modules/clauses/clauses.view.php` должна называться `show_block_template`; по умолчанию шаблон `show_block`).

Формат даты задается в конфигурации модуля.

Пример:

```
<insert name="show_block" module="clauses">
```

выведет 3 последние статьи сайта

```
<insert name="show_block" module="clauses" count="5">
```

выведет 5 последних статей сайта

```
<insert name="show_block" module="clauses" count="1" site_id="35">
```

выведет одну последнюю статью сайта, из раздела статей с id=35

Шаблон: функция *show\_block* в файле *modules/clauses/clauses.view.php*.

CSS-форматирование:

`.clauses_block` – блок статей

`.clauses_block h2` – заголовок

`.clauses_block .clauses` – статья в блоке

`.clauses_block .clauses_name` – название статьи

`.clauses_block .clauses_img` – изображение статьи

`.clauses_block .clauses_date` – дата статьи

`.clauses_block .clauses_anons` – анонс статьи

`.clauses_block .show_all` – ссылка на все статьи

### Вопрос-ответ

**show\_block** – выводит последние вопросы. На странице вопросов, когда выводится список аналогичных вопросов, блок не выводится.

Атрибуты:

- **count** – количество выводимых вопросов (по умолчанию 3);
- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** – категория вопросов (id категории, по умолчанию выводятся все вопросы без учета категории), если в конфигурации модуля отмечено «Использовать категории»;
- **often** – часто задаваемые вопросы (если often=1, выводятся вопросы с пометкой «Часто задаваемый вопрос»);
- **template** – шаблон тега (функция в файле *modules/faq/faq.view.php* должна называться *show\_block\_template*; по умолчанию шаблон *show\_block*).

Формат даты задается в конфигурации модуля.

```
Пример: <insert name="show_block" module="faq">
```

выведет 3 последних активных вопроса сайта с ответами

```
<insert name="show_block" module="faq" count="5">
```

выведет в ячейке таблицы 5 последних активных вопросов сайта с ответами

```
<insert name="show_block" module="faq" often="1">
```

выведет часто задаваемые вопросы

Шаблон: функция *show\_block* в файле *modules/faq/faq.view.php*.

CSS-форматирование:

*.faq\_block*{ } – шаблонная функция *show\_block*

*.faq\_block h2*{ } – заголовок

*.faq\_block .faq*{ } – вопрос в блоке

*.faq\_block .faq\_date*{ } – дата

*.faq\_block .faq\_question*{ } – вопрос

*.faq\_block .faq\_answer*{ } – ответ

*.faq\_block .show\_all*{ } – ссылка на все вопросы

**show\_form** – выводит форму добавления вопроса. Для правильной работы тега должна существовать и быть активной страница, к которой прикреплен модуль Вопрос-Ответ.

Атрибуты:

— **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;

— **cat\_id** – категория вопросов (id категории, по умолчанию выбирается одна категория), если в конфигурации модуля отмечено «Использовать категории»;

— **template** – шаблон тега (функция в файле *modules/feedback/feedback.view.php* должна называться *form\_template*; по умолчанию шаблон *form*).

```
Пример: <insert name="show_form" module="faq">
```

выведет форму добавления вопроса

Шаблон: функция *form* в файле *modules/faq/faq.view.php*.

CSS-форматирование:

*.faq\_form*{ } – форма «Задать вопрос»

*.faq\_form .inptext*{ } – поля «Ваше имя», «Ваш вопрос», «E-mail для ответа»

*.faq\_form .inpfile*{ } – поле «Прикрепляемые файлы»

*.faq\_form .infofield*{ } – подпись к полю в форме

*.faq\_form .button*{ } – кнопка «Отправить» в форме

[Обратная связь](#)

**show\_form** – выводит форму добавления сообщения. Для правильной работы тега должна существовать и быть активной страница, к которой прикреплен модуль Обратная связь.

Атрибуты:

- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **template** – шаблон тега (функция в файле *modules/feedback/feedback.view.php* должна называться *form\_template*; по умолчанию шаблон *form*).

Пример: `<insert name="show_form" module="feedback">`

выведет форму добавления сообщения

Шаблон: функция *form* в файле *modules/feedback/feedback.view.php*.

CSS-форматирование:

`.feedback_form{}` – форма

`.feedback_form .infofiled{}` – подпись к полю в форме

`.feedback_form .inptext{}` – текстовые поля

`.feedback_form .inpnum{}` – цифровые поля

`.feedback_form .inpselect{}` – поля «Список»

`.feedback_form .inpcheckbox{}` – поля «Чекбокс»

`.feedback_form .inpfile{}` – поле «Файл»

`.feedback_form .button{}` – кнопка «Оправить»

`.feedback_form_param_text{}` – описание поля

`.feedback_form_paramNUM{}` – поле с номером NUM

`.feedback_form_paramNUM .infofield{}` – название параметра с номером NUM

`.feedback_form_paramNUM .inptext{}` – текстовое поле для ввода текстового значения параметра с номером NUM

`.feedback_form_paramNUM .inpnum{}` – текстовое поле для ввода цифрового значения параметра с номером NUM

`.feedback_form_paramNUM .inpselect{}` – список для выбора значения параметра с номером NUM

`.feedback_form_paramNUM .inpcheckbox{}` – чекбокс для выбора значения параметра с номером NUM

## **Пользователи**

**show\_login** – выводит форму авторизации или приветствие и ссылки на редактирование данных и выход, если пользователь авторизован.

Пример: `<insert name="show_login" module="registration">`

выведет форму авторизации

Шаблон: функция *show\_login* в файле *modules/registration/registration.view.php*.

CSS-форматирование:

`.login`{} – шаблонная функция `show_login` – авторизация

`.login .login_avatar`{} – аватар пользователя

`.login .infofield`{} – подпись к полю в форме

`.login .inpbutton`{} – слой с кнопкой «ОК»

`.login .button`{} – кнопка «ОК»

`.login .inptext`{} – текстовые поля для ввода логина и пароля

`.login_reminding`{} – слой с ссылкой «Забыли пароль?»

`.login_registration`{} – слой с ссылкой «Регистрация»

## II. Модули

### Страницы сайта

- [Настройки](#)
- [Файлы](#)

Модуль «Страницы сайта» – основной механизм управления страницами сайта.

Модуль относится к части CMS «Контент» и является обязательным для установки.

Если перед названием страницы имеется знак +, то этой странице принадлежат вложенные страницы нижнего уровня. Добавить страницу нижнего уровня можно, нажав на знак  рядом с названием текущей страницы в списке всех страниц. Кнопка  служит для удаления текущей страницы. Если странице принадлежат страницы нижнего уровня, то при удалении они будут также удалены. Кнопка  служит для быстрой публикации страницы (или, наоборот,  блокирования). Порядок отображения страниц можно редактировать с помощью кнопок  .

Если страница помечена опцией «Блок на сайте», то перед ее названием будет выведена метка «(блок)».

Если к странице подключен какой-либо модуль, то нажатие на название страницы переводит к редактированию этого модуля. А рядом с его названием появится ссылка «изменить», по которой можно изменить саму страницу, к которой прикреплен модуль.

Например, если в структуре сайта есть страница «Наши новости», то нажатие на её название приведет к редактированию этой страницы. А если к странице «Наши новости» прикрепить модуль «Новости», то нажатие приведет к списку новостей из модуля «Новости». Отредактировать же страницу «Наши новости» можно, нажав на «(изменить)» рядом с заголовком, перед списком новостей.

Страница сайта имеет следующие характеристики.

Основные:

- **Название** – используется как заголовок страницы а так же выводится автоматически в виде заголовка *h1* в поле для контента (если не отмечен параметр «Не копировать название в заголовок H1»);
- **Показывать на сайте** – публикация на сайте, активность;
- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Контент страницы** – текстовое поле. В текстовое поле можно вставлять шаблонные теги.

Дополнительные:

- **Изображения** – возможность загрузки нескольких изображений, если в конфигурации модуля включен параметр «Использовать изображения». Изображение загружаются в двух вариантах: маленьком (предпросмотр) и большом (увеличенная версия). Также для отражения в административной части сайта генерируются дополнительные изображения (50x50, качество 70). Качество и размер загружаемых изображений для пользовательской части

сайта указываются в настройках модуля. Отображение фотографий осуществляется с помощью шаблонной функции [show\\_photo](#);

- **Номер** – номер страницы, используется для шаблонных тегов;
- **Блок на сайте** – страница используется для шаблонного тега [show\\_block](#), на сайте не отображается;
- **Время последнего изменения** – выводится в серверных заголовках *Last Modify*;
- **Заголовок окна в браузере, тэг Title** – Если не заполнен, тег title будет автоматически сформирован из Названия страницы - Названия раздела - Названия сайта;
- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «*keywords*», используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «*description*», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу будет генерироваться следующим образом: <http://www.site.ru/псевдоссылка/>. Если при сохранении страницы псевдоссылку не указать, она автоматически сгенерируется из поля «Название в меню» транслитом и псевдоссылки страницы верхнего уровня;
- **Не копировать название в заголовок H1** – позволяет запретить автоматический вывод заголовка *title* в заголовке *h1*, перед основным контентом (при использовании [show\\_body](#));
- **Не показывать на карте сайта** – скрывает отображение ссылки на категорию в файле `sitemap.xml` и [модуле «Карта сайта»](#).
- **Прикрепить модуль** – позволяет прикрепить модуль к странице сайта;
- **Вложенность: принадлежит** – страница верхнего уровня, которой принадлежит страница;
- **Сортировка: установить перед** – редактирование порядка отображения пункта;
- **Дизайн страницы** – возможность подключить к странице шаблон отличный от основного (*site.php*). Все шаблоны для сайта должны храниться в папке `themes` и содержать слово *site* (например, *siteother.php*). Подробнее в [разделе «Шаблоны сайта»](#);
- **Доступ только для зарегистрированных** – страницу увидят только авторизованные пользователи.

Если в конфигурации модуля включен параметр «Показывать комментарии к страницам сайта», то при редактировании страницы будет отображена активная ссылка на комментарии или надпись «Комментариев нет» при их отсутствии.

Шаблонные функции: [show\\_photo](#), [show\\_links](#), [show\\_previous\\_next](#), [show\\_block](#).

## Настройки

**Использовать изображения** – параметр позволяет включить/отключить загрузку и отображение изображений, прикрепленных к страницам.

**Размер большого изображения** – размер изображения, отображаемый в

пользовательской части сайта при увеличении изображения предпросмотра.

**Размер маленького изображения** – размер изображения предпросмотра.

**Использовать анимацию при увеличении изображений** – параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений.

**Максимальный размер загружаемых файлов** – параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.

**Накладывать водяной знак** – параметр позволяет включить/отключить наложение водяного знака на изображение.

**Расположение водяного знака** – расположение водяного знака в пикселях от левого края или от центра и от нижнего края или от центра.

**Использовать водяной знак только для большого изображения** – отключает наложение водяного знака на изображение предпросмотра.

**Применить настройки ко всем ранее загруженным изображениям** – позволяет перекодировать размер уже загруженных изображений. Размер вновь созданных изображений не может быть больше размера большой копии уже загруженного изображения.

**Защита от копирования** – параметр добавляет JavaScript код, запрещающий копирование контента. См. шаблонную функцию [show\\_protect](#).

**Прикрепить комментарии к страницам сайта** – подключение модуля «Комментарии». Комментарии для страницы сайта выводятся с помощью шаблонной функции [show\\_comments](#). Комментарии выводятся только на страницах, к которым не подключен иной модуль со своими комментариями. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в [разделе «Модуль Комментарии»](#).

**Подключить теги** — подключение модуля «Теги». Теги для страницы сайта выводятся с помощью шаблонной функции [show\\_tags](#). Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [«Модуль Теги»](#).

## Файлы

1. *modules/site/admin/site.admin.php* – редактирование страниц сайта;
2. *modules/site/admin/site.admin.config.php* – конфигурация модуля;
3. *includes/function.php* – функции `show_photo`, `show_previous_next`, `show_block` и `show_links`.

## Меню на сайте

- [Категории](#)
- [Пункты меню](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль «Меню на сайте» формирует меню любой сложности и в любом количестве.

Модуль относится к части CMS «Контент» и является обязательным для установки.

Модуль состоит из 2х частей: категории меню и пункты меню.

### Категории

Категории меню – это различные меню на сайте. Категорий меню может быть неограниченное множество. Выводятся они на сайте с помощью шаблонного тега [show\\_block](#).

Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот,  блокирования). При удалении категории удаляются все вложенные пункты меню.

Категории меню имеют следующие характеристики:

- **Название;**
- **Номер** – номер категории;
- **Показывать на сайте** – публикация на сайте, активность;
- **Показывать заголовок** – на сайте отображается название меню
- **Раскрывать все пункты меню** – на сайте отображаются сразу все пункты меню, не только вложенные активного пункта;
- **Не отображать ссылку на элемент, если он имеет дочерние пункты.**

### Пункты меню

Пункты меню – ссылки, формирующие меню. Пункты меню могут ссылаться на страницы сайта, категории модулей и элементы модулей, а также на любую внешнюю ссылку.

Если перед названием пункта меню имеется знак +, то этому пункту принадлежат вложенные пункты нижнего уровня. Добавить пункт нижнего уровня можно, нажав на знак  рядом с названием текущего пункта в общем списке.

Кнопка  служит для удаления текущего пункта меню. Если пункту меню принадлежат пункты нижнего уровня, то при удалении они будут также удалены.

Кнопка  служит для быстрой публикации пункта меню (или, наоборот,  блокирования). Порядок отображения пунктов меню можно редактировать с помощью кнопок  .

Пункты меню имеют следующие характеристики:

- **Название** – название пункта меню;

- **Показывать на сайте** – публикация на сайте, активность;
- **Элемент** – страница сайта, категория или элемент модуля, на который ссылается пункт меню;
- **Вложенность: принадлежит** – пункт меню верхнего уровня;
- **Категория** – категория меню;
- **Внешняя ссылка (в формате `http://site.com/`)** – если в меню необходимо вывести ссылку на другой сайт, то заполняется данное поле (например, на форум на поддомене: `http://forum.site.ru/`). Если «Внешняя ссылка» заполнена, остальные характеристики игнорируются.

При блокировании элемента, на который ссылается пункт меню, соответствующий пункт меню тоже блокируется. Также при удалении элемента, соответствующий пункт меню удаляется.

Структуру меню можно формировать непосредственно из модуля «Меню на сайте» или при заполнении сайта. При этом структура сайта повторяется в меню.

Например, отмечаем опцию «Показывать в меню» для новости. Если категория, которой принадлежит новость также имеет ссылку в меню, то новый пункт меню добавится как вложенный в пункт категории меню. Иначе добавится пункт меню первого уровня.

При формировании списка элементов, которые можно выбрать в качестве ссылок для пунктов меню, используются элементы других модулей. Ссылки на эти элементы формируются с помощью файла `modules/модуль/admin/модуль.admin.menu.php`.

## CSS-форматирование

`.menu_block{}` – блок меню, если используется шаблон по-умолчанию

`.menu_block h2{}` – заголовок блока меню

`.menu_item{}` – пункт меню

`.menu_active{}` – текущий пункт меню

`.menu_active_child{}` – пункт меню с текущим дочерним пунктом

`.menu_level_НОМЕР{}` – уровень меню НОМЕР

`.menu_item_НОМЕР{}` – пункт меню уровня НОМЕР

## Файлы

1. `modules/menu/menu.php` – контроллер модуля;
2. `modules/menu/menu.model.php` – формирование данных для меню;
3. `modules/menu/menu.view.php` – шаблон модуля;
4. `modules/menu/admin/menu.admin.php` – редактирование пунктов меню;
5. `modules/menu/admin/menu.admin.category.php` – редактирование категорий меню;
6. `modules/menu/admin/menu.admin.act.php` – активирование / блокирование пунктов меню при активировании / блокировании элементов модулей;
7. `modules/menu/admin/menu.admin.edit.php` – редактирование меню в административной части модулей;

8. *modules/menu/admin/menu.admin.del.php* – удаление пунктов меню при удалении элементов модулей;
9. *modules/menu/admin/menu.admin.save.php* – сохранение меню в административной части модулей;
10. *modules/menu/admin/menu.admin.ajax.php* – обработка Ajax-запросов в административной части модуля;
11. *modules/menu/admin/menu.admin.js* – редактирование пунктов меню, JS-сценарий;
12. *modules/модуль/admin/модуль.admin.menu.php* – формирование списка элементов модуля, доступные для добавления в меню.

# Новости

- [Категории](#)
- [Новости](#)
- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль «Новости» отображает тексто-графические материалы с сортировкой по датам.

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль «Новости» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка новостей, категорий и настроек) по принадлежности к странице.

Модуль состоит из 2х частей: категории новостей и новости.

## Категории

Категории новостей могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в конфигурации модуля.

Если перед названием категории имеется знак +, то этой категории принадлежат подкатегории. Добавить подкатеорию к категории можно, нажав на знак  рядом с названием категории.

Порядок отображения категорий можно редактировать с помощью кнопок  .

Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот,  блокирования). При удалении категории удаляются все вложенные подкатегории и новости.

Категории имеют следующие характеристики.

Основные:

- **Название;**
- **Показывать на сайте;**
- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Анонс** – краткое описание категории, для публикации в списке;
- **Описание** – описание категории.

Дополнительные:

- **Номер** – номер категории;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет автоматически сформирован из Названия категории - Названия страницы

сайта - Названия сайта;

- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords», используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу категории будет генерироваться следующим образом: *http://www.site.ru/псевдоссылка/*. Если псевдоссылка не указана адрес страницы будет: *http://www.site.ru/псевдоссылка\_страницы/catНомер\_категории/*. При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы. При редактировании можно это исправить;
- **Вложенность: принадлежит** – категория верхнего уровня;
- **Раздел сайта** – принадлежность к странице сайта;
- **Сортировка: установить перед** – редактирование порядка отображения пункта. Поле доступно для редактирования только для элементов, отображаемых на сайте;
- **Не показывать на карте сайта** – скрывает отображение ссылки на категорию в файле sitemap.xml и [модуле «Карта сайта»](#).

Если в конфигурации модуля включен параметр «Показывать комментарии к категориям», то при редактировании категории будет отображена активная ссылка на комментарии к категории или надпись «Комментариев нет» при их отсутствии.

## Новости

В списке новости выводятся с приоритетом важности и новизны.

Кнопка  служит для удаления новости. Кнопка  служит для быстрой публикации новости (или, наоборот,  блокирования).

Новости имеют следующие характеристики.

Основные:

- **Название;**
- **Дата** – новости, старше текущей даты начнут отображаться на сайте, начиная с указанной даты;
- **Изображения** – возможность загрузки нескольких изображений, если в конфигурации модуля включен параметр «Использовать изображения». Изображение загружаются в двух вариантах: маленьком (предпросмотр) и большом (увеличенная версия). Также для отражения в административной части сайта генерируются дополнительные изображения (50x50, качество 70). Качество и размер загружаемых изображений для пользовательской части сайта указываются в настройках модуля;
- **Показывать на сайте** – публикация на сайте, активность;
- **Категория** – помимо основной можно указать дополнительные категории;
- **Анонс** – краткое содержание новости, для публикации в списке;
- **Описание** – содержание новости.

Дополнительные:

- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Номер** – номер новости;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет автоматически сформирован из Названия новости - Названия страницы сайта - Названия сайта;
- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords», используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу с новостью будет генерироваться следующим образом: <http://www.site.ru/псевдоссылка/>. Если псевдоссылка не указана адрес страницы будет: [http://www.site.ru/псевдоссылка\\_страницы/catНомер\\_категории/showНомер\\_новости/](http://www.site.ru/псевдоссылка_страницы/catНомер_категории/showНомер_новости/). При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы и категории новости. При редактировании можно это исправить;
- **Раздел сайта** – принадлежность к странице сайта;
- **Важно (всегда сверху)** – новость будет всегда сверху в списке всех новостей, независимо от даты;
- **Не показывать на карте сайта** – скрывает отображение ссылки на новость в файле *sitemap.xml*.

Если в конфигурации модуля включен параметр «Показывать комментарии к новостям», то при редактировании новости будет отображена активная ссылка на комментарии к новости или надпись «Комментариев нет» при их отсутствии.

Если в конфигурации модуля включен параметр «Показывать рейтинг новостей», то при редактировании новости будет отображена активная ссылка на рейтинг новости с указанием среднего балла и количества голосов или надпись «Рейтинг: нет голосов», если голосов за новость нет.

Если в конфигурации модуля включен параметр «Подключить теги», то при редактировании новости появится возможность редактирования тегов, прикрепленных к новости.

Шаблонные функции: [show\\_block](#), [show\\_calendar](#).

В модуль «Новости» встроена автоматическая публикация новостей в формате RSS.

Адрес rss-ленты: <http://site.ru/news/rss/>.

У англоязычной версии сайта адрес RSS-ленты новостей будет иметь вид: <http://site.ru/eng/news/rss/>

## Настройки

Можно сохранять разные конфигурации модуля для разных страниц, к которым прикреплен модуль.

**Использовать категории** – параметр позволяет включить/отключить категории новостей.

**Количество новостей на странице** – количество одновременно выводимых новостей в списке новостей, после чего начинается строка навигации.

**Количество новостей в списке категорий** – количество новостей, выводимых в списке категорий на главной странице модуля.

**Показывать новости подкатегорий** – этот параметр позволяет показывать в категории верхнего уровня не только новости, принадлежащие этой категории, но и новости подкатегорий, принадлежащих данной категории.

**Использовать изображения** – параметр позволяет включить/отключить загрузку изображений к новости.

**Размер большого изображения** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер маленького изображения** – размер изображения предпросмотра.

**Отображение изображений в списке новостей** – параметр принимает значения:

- нет (отключает отображение изображений в списке);
- показывать одно изображение;
- показывать все изображения.

**Использовать изображения для категории** – параметр позволяет включить/отключить загрузку изображений для категорий.

**Размер большого изображения для категории** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер маленького изображения для категории** – размер изображения предпросмотра.

**Отображение изображений категорий в списке категорий** – параметр принимает значения:

- нет (отключает отображение изображений в списке);
- показывать одно изображение;
- показывать все изображения.

**Использовать анимацию при увеличении изображений** – параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений.

**Максимальный размер загружаемых файлов** – параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.

**Накладывать водяной знак** – параметр позволяет включить/отключить наложение водяного знака на изображение.

**Расположение водяного знака** – расположение водяного знака в пикселях от левого края или от центра и от нижнего края или от центра.

**Использовать водяной знак только для большого изображения** – отключает наложение водяного знака на изображение предпросмотра.

**Применить настройки ко всем ранее загруженным изображениям** – позволяет перекодировать размер уже загруженных изображений. Размер вновь созданных изображений не может быть больше размера большой копии уже загруженного

изображения.

**Формат даты** – позволяет настроить отображение даты в модуле.

**Показывать комментарии к новостям** – подключение модуля «Комментарии». Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в [разделе «Модуль Комментарии»](#).

**Показывать комментарии к категориям** – подключение модуля «Комментарии» к категориям новостей. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в [разделе «Модуль Комментарии»](#).

**Подключить теги** – подключение модуля «Теги» к модулю «Новости». Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [«Модуль Теги»](#).

**Показывать рейтинг новостей** – подключение модуля «Рейтинг». Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [«Модуль Рейтинг»](#).

## CSS-форматирование

Общие стили:

`.news_name{}` – заголовок новости

`.news_date{}` – дата новости

`.news_anons{}` – анонс новости

`.news_text{}` – текст новости

`.news_img{}` – изображение, прикрепленное к новости

`.news_all_img{}` – все изображения, прикрепленные к новости

Список новостей:

`.news_list{}` – список новостей

`.news_list .news{}` – новость в списке

`.news_list .news_date{}` – дата новости в списке

`.news_list .news_name{}` – название новости в списке

`.news_list .news_anons{}` – анонс новости в списке

`.news_list .news_img{}` – изображение, прикрепленное к новости в списке

`.news_list .news_cat_anons{}` – краткое описание категории новости

`.news_list .news_cat_text{}` – полное описание категории новости

`.news_list .news_cat_img{}` – изображение, прикрепленное к категории новости

`.news_list .news_cat_all_img{}` – все изображения, прикрепленные к категории новости

`.news_list .news_cat_link{}` – ссылка на вложенную категорию новостей

`.news_list .show_all{}` – ссылка на все новости в категории

`.news_list .previous_next_links{}` – ссылки на предыдущую, следующую категории

`.news_list .previous_link{}` – ссылка на предыдущую категорию

`.news_list .next_link{}` – ссылка на следующую категорию

Страница новости:

`.news_id` – страница новости  
`.news_id .news_date` – дата новости  
`.news_id .news_anons` – анонс новости  
`.news_id .news_text` – текст новости  
`.news_id .news_all_img` – все изображения, прикрепленные к новости  
`.news_id .show_all` – ссылка на все новости  
`.news_id .previous_next_links` – ссылки на предыдущую, следующую новости  
`.news_id .previous_link` – ссылка на предыдущую новость  
`.news_id .next_link` – ссылка на следующую новость

Шаблонная функция [show\\_block](#):

`.news_block` – шаблонная функция `show_block`  
`.news_block h2` – заголовок  
`.news_block .news` – новость в блоке  
`.news_block .news_name` – название новости  
`.news_block .news_img` – изображение новости  
`.news_block .news_date` – дата новости  
`.news_block .news_anons` – анонс  
`.news_block .show_all` – ссылка на все новости

Шаблонная функция [show\\_calendar](#):

`.news_calendar` – шаблонная функция `show_calendar`  
`.news_calendar h2` – заголовок  
`.news_calendar .news_year` – год  
`.news_calendar .news_year_current` – текущий (открытый) год  
`.news_calendar .news_month` – месяц  
`.news_calendar .news_month_current` – текущий (открытый) месяц  
`.news_calendar .show_all` – ссылка на все новости

## Файлы

1. `modules/news/news.php` – контроллер модуля;
2. `modules/news/news.model.php` – формирование данных для страницы модуля;
3. `modules/news/news.view.php` – шаблон модуля;
4. `modules/news/news.install.php` – установка модуля;
5. `modules/news/news.sitemap.php` – карта модуля для файла `sitemap.xml`;
6. `modules/news/news.rss.php` – RSS-лента;
7. `modules/news/admin/news.admin.php` – редактирование новостей;
8. `modules/news/admin/news.admin.category.php` – редактирование категорий новостей;

9. *modules/news/admin/news.admin.config.php* – конфигурация модуля;
10. *modules/news/admin/news.admin.menu.php* – формирование списка новостей и категорий новостей в административной части модуля «Меню».

## Магазин

- [Категории](#)
- [Товары](#)
- [Характеристики](#)
- [Заказы](#)
- [Отчет о продажах](#)
- [Оформление заказа](#)
- [Скидки](#)
- [Импорт](#)
- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль «Магазин» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка товаров, категорий и настроек) по принадлежности к странице.

Модуль состоит из 8 частей: товары, категории товаров, характеристики, заказы, отчет о продажах, конструктор оформления заказа, скидки, импорт.

## Категории

Категории товаров могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в конфигурации модуля.

Если перед названием категории имеется знак +, то этой категории принадлежат подкатегории. Добавить подкатеорию к категории можно, нажав на знак  рядом с названием категории.

Порядок отображения категорий можно редактировать с помощью кнопок  .

Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот,  блокирования). При удалении категории удаляются все вложенные подкатегории и товары.

Категории имеют следующие характеристики.

Основные:

- **Название;**
- **Показывать на сайте;**
- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Анонс** – краткое описание категории, для публикации в списке;

- **Описание** – описание категории.

Дополнительные:

- **Номер** – номер категории;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет автоматически сформирован из Названия категории - Названия страницы сайта - Названия сайта;
- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords», используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу категории будет генерироваться следующим образом: *http://www.site.ru/псевдоссылка/*. Если псевдоссылка не указана адрес страницы будет: *http://www.site.ru/псевдоссылка\_страницы/catНомер\_категории/*. При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы. При редактировании можно это исправить;
- **Вложенность: принадлежит** – категория верхнего уровня;
- **Раздел сайта** – принадлежность к странице сайта;
- **Сортировка: установить перед** – редактирование порядка отображения пункта. Поле доступно для редактирования только для элементов, отображаемых на сайте;
- **Выгружать в Яндекс Маркет** – параметр появляется, если в конфигурации модуля параметр «Выгружать категории в Яндекс.Маркет» определен как «только помеченные»;
- **Не показывать на карте сайта** – скрывает отображение ссылки на категорию в файле *sitemap.xml* и [модуле «Карта сайта»](#).

Если в конфигурации модуля включен параметр «Показывать комментарии к категориям», то при редактировании категории будет отображена активная ссылка на комментарии к категории или надпись «Комментариев нет» при их отсутствии.

## Товары

Порядок отображения товаров можно редактировать с помощью кнопок  . Если подключены категории товаров, то сортировать товары можно только в пределах категории (в фильтре «Категория» нужно выбрать категорию товаров). Кнопка  служит для удаления товаров. Кнопка  служит для быстрой публикации товаров (или, наоборот,  блокирования).

Товары имеют следующие характеристики.

Основные:

- **Название**;
- **Артикул**;
- **Показывать на сайте** – публикация на сайте, активность;

- **Цена, количество** – цифровое поле, если используется зависимая цена, то это поле скрывается. Поле «Количество» отображается, если в настройках модуля подключена опция «Учитывать остатки товаров на складе». Если поле «Количество» пустое, то товар нельзя купить. Можно задать несколько вариантов цены и количества для товара. Для этого следует создать дополнительную характеристику с типом «список с выбором нескольких значений» и при редактировании товара отметить возле характеристики опцию «Влияет на цену»;
- **Товар временно отсутствует** – скрывает кнопку «Купить» у товара и выводит надпись «Товар временно отсутствует»;
- **Изображения** – возможность загрузки нескольких изображений, если в конфигурации модуля включен параметр «Использовать изображения». Изображение загружаются в двух вариантах: маленьком (предпросмотр) и большом (увеличенная версия). Также для отражения в административной части сайта генерируются дополнительные изображения (50x50, качество 70). Качество и размер загружаемых изображений для пользовательской части сайта указываются в настройках модуля;
- **Дополнительные характеристики** – группа полей, определенных в части «Характеристики». Для характеристики с типом «список с выбором нескольких значений» можно задать несколько цен для товара. Для одного товара можно выбрать только одну характеристику, влияющую на цену;
- **Похожие товары** – возможность задать похожие товары для вывода соответствующего блока на странице товары. Связи между товарами являются двусторонними;
- **Категория** – помимо основной можно указать дополнительные категории;
- **Анонс** – краткое описание товара;
- **Описание** – полное описание товара.

Дополнительные:

- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Номер** – номер товара;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет автоматически сформирован из Названия товара - Названия страницы сайта - Названия сайта;
- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords», используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу статьи будет генерироваться следующим образом: <http://www.site.ru/псевдоссылка/>. Если псевдоссылка не указана адрес страницы будет: [http://www.site.ru/псевдоссылка\\_страницы/catНомер\\_категорию/showНомер\\_товара/](http://www.site.ru/псевдоссылка_страницы/catНомер_категорию/showНомер_товара/). При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы и категории товара. При редактировании можно это

исправить;

- **Раздел сайта** – принадлежность к странице сайта;
- **Сортировка: установить перед** – редактирование порядка отображения пункта. Поле доступно для редактирования только для элементов, отображаемых на сайте;
- **Выгружать в Яндекс Маркет** – параметр появляется, если в конфигурации модуля параметр «Выгружать товары в Яндекс.Маркет» определен как «только помеченные»;
- **Поля для Яндекс Маркет** – параметры, необходимые для формирования информации для системы «Яндекс Маркет» файлом `/modules/shop/shop.yandex.php` (если в конфигурации модуля включен параметр «Подключить Яндекс Маркет»).

Если в конфигурации модуля включен параметр «Показывать комментарии к товарам», то при редактировании товара будет отображена активная ссылка на комментарии к товарам или надпись «Комментариев нет» при их отсутствии.

Если в конфигурации модуля включен параметр «Показывать рейтинг товаров», то при редактировании товара будет отображена активная ссылка на рейтинг товара с указанием среднего балла и количества голосов или надпись «Рейтинг: нет голосов», если голосов за товар нет.

Если в конфигурации модуля включен параметр «Подключить теги», то при редактировании товара появится возможность редактирования тегов, прикрепленных к товару.

## Характеристики

Порядок отображения характеристик товаров можно редактировать с помощью кнопок  . Кнопка  служит для удаления характеристики.

Характеристики – это дополнительные параметры, характеризующие товар. Можно добавлять характеристики общие (для всех товаров в пределах одного раздела сайта) или присваивать характеристику одной или нескольким категориям товаров.

Характеристики имеют следующие параметры.

Основные:

- **Название;**
- **Тип** – строка, число, дата, текстовое поле, галочка, выпадающий список, список с выбором нескольких значений, поле с визуальным редактором, заголовок группы характеристик;
- **Использовать в форме поиска;**
- **Показывать в списке;**
- **Показывать в блоке товаров;**
- **Раздел сайта** – принадлежность к странице сайта;
- **Категория** – категории товаров, для которых действует данный параметр. Если не указана ни одна категория, то характеристика считается общей;
- **Доступен к выбору при заказе** – поле с типом «список с выбором нескольких значений» можно выводить в виде выпадающего списка рядом с кнопкой

«Купить» в пользовательской части, если нужно задавать значение характеристики при покупке товара;

- **Отдельная страница для значений** – поле появляется для характеристик с типом «выпадающий список» и «список с выбором нескольких значений». Если опция отмечена, то на сайте появляется страница с товарами, объединенными одной характеристикой. Например, цвет: красный. Странице можно задать ЧПУ, для этого рядом с полем «Значение» выводится дополнительное поле «Псевдоссылка»;
- **Значения** – поле появляется для характеристик с типом «галочка», «выпадающий список» и «список с выбором нескольких значений»;
- **Описание характеристики** – поле используется для комментирования характеристик в списке. В пользовательской части поле не используется.

Если выбран тип «галочка», появится 2 дополнительных поля: «да» и «нет». Они позволяют задать отображение параметра в пользовательской части сайта. Если значения пустые, то при показе товара будет, отображается только название параметра (при активной галочке) и не отображается ничего (при неактивной галочке).

Пример: *Есть на складе*

Если значения заданы, то отображаются эти значения.

Пример:

*На складе: есть*

*На складе: отсутствует.*

Если выбран тип «выпадающий список» или «список с выбором нескольких значений», то появятся дополнительные поля со значениями. Добавить значение можно, нажав  внизу последнего поля, сортировать значения можно с помощью кнопок  . Кнопка  служит для удаления значений.

Если выбран тип «поле с визуальным редактором», то при редактировании поля будет отображаться визуальный редактор, что позволяет редактировать стиль текста.

## Заказы

Отображается таблица со всеми заказами, поступающими из пользовательской части сайта. Таблица содержит:

- **Дата** – дата создания заказа;
- **Номер заказа** (в виде ссылки на подробную информацию);
- **Статус** (новый – красный, в обработке – синий, отменен – серый, выполнен – серый). Изменить список статусов можно в файле `modules/shop/shop.admin.order.php` массивы `$this->select_arr["status"]` и `$this->select_arr["color"]`;
- **Пользователь, добавивший заказ** (в виде ссылки на подробную информацию о пользователе или надпись «без регистрации», если пользователь предпочел не регистрироваться);

- **Дополнительные поля** – группа полей, определенных в части «Оформление заказа».

Кнопка  служит для удаления заказа.

Заказы имеют следующие параметры.

Основные:

- **Номер заказа;**
- **Пользователь** – если заказал зарегистрированный пользователь;
- **Дополнительные поля** – группа полей, определенных в части «Оформление заказа»;
- **Дата** – дата создания заказа;
- **Статус заказа** (новый, в обработке, отменен, выполнен) – если в настройках модуля «Магазин» отмечена опция «Учитывать остатки товаров на складе», то при смене статуса на «в обработке», количество товара уменьшится и появится соответствующая запись в отчете о продажах;
- **Способ оплаты** – список подключенных методов оплаты;
- **Язык интерфейса** – фиксируется язык интерфейса, который был при оформлении заказа;
- **Таблица заказанных товаров.**

**Корзина** – это отдельный модуль в пользовательской части сайта (basket). Он устанавливается вместе с модулем «Магазин» и необходим для просмотра корзины. Сохранение изменений в корзине осуществляются с помощью технологии Ajax, то есть без перезагрузки всей страницы.

**Оформление заказа** – это отдельный модуль в пользовательской части сайта (order). Он устанавливается вместе с модулем «Магазин» и необходим для оформления заказа. Оформление заказа осуществляется с помощью технологии Ajax, то есть без перезагрузки всей страницы.

При оформлении заказа администратор получает уведомление о поступившем заказе на электронную почту, пользователь получает уведомление об оформлении заказа на электронную почту и заказ добавляется в базу данных.

Шаблонные функции: [show\\_catalog](#), [show\\_block](#), [show\\_search](#), [show\\_basket](#).

## Отчет о продажах

Отчет о продажах — таблица с перечнем проданных товаров в хронологическом порядке с выводом общей суммы за период.

Отчет формируется, если в настройках модуля включена опция «Учитывать остатки товаров на складе». При этом записи в таблицу делаются при смене статуса на заказа на «В обработке»: вручную из панели администрирования или автоматически при выборе системы оплаты Webmoney или Roboxchange.

## Оформление заказа

Форму «Оформление заказа» можно дополнить своими полями с помощью конструктора формы.

Порядок отображения полей можно редактировать с помощью кнопок  . Кнопка .

служит для удаления полей.

Дополнительные поля имеют следующие параметры.

Основные:

- **Название;**
- **Тип** – строка, число, дата, текстовое поле, галочка, выпадающий список, список с выбором нескольких значений, электронный ящик;
- **Обязательно для заполнения;**
- **Использовать в форме** – возможные значения:
  - нет;
  - во всех формах;
  - только при регистрации;
  - только при редактировании данных.
- **Значения** – появляется для полей с типом «галочка», «выпадающий список» и «список с выбором нескольких значений».

Дополнительные:

- **Сортировка: установить перед** – редактирование порядка отображения поля.

Если выбран тип «выпадающий список» или «список с выбором нескольких значений», то появятся дополнительные поля со значениями. Добавить значение можно, нажав  внизу последнего поля, сортировать значения можно с помощью кнопок  . Кнопка  служит для удаления значений.

## Скидки

Скидки можно установить на весь магазин, на несколько категорий и на несколько товаров.

Скидки имеют следующие характеристики:

- **Скидка** – процент скидки;
- **Показывать на сайте** – публикация на сайте, активность;
- **Период действия** – даты начала и окончания действия скидки, обе даты являются необязательными параметрами;
- **Категория** – категории, на которые распространяется скидка;
- **Товар** – товары, на которые распространяется скидка.

Количество предлагаемых скидок не ограничено. Из нескольких скидок, предлагаемых для одного товара выбирается одна, которая имеет меньшую область применения.

## Импорт

Импорт товаров и категорий возможен из файлов в формате CSV.

Прежде чем начать импорт, нужно описать содержание файлов.

Во-первых нужно добавить новый файл, нажав на ссылку «Настроить новый»

импорт».

Файл импорта имеет следующие характеристики:

- **Название** – краткое описание файла импорта (например, «Импорт товаров», «Импорт цен» и т. д.);
- **Тип** – товар или категория.
- **Удалить неописанные в файле импорта записи** – если список товаров или категорий в файле является исчерпывающим, то следует отметить эту опцию. Тогда при импорте удалятся ранее занесенные товары или категории, кроме тех, что будут обновлены (определяется по идентификатору);
- **Раздел сайта** – страница сайта с прикрепленным модулем «Магазин», для которой будет производится импорт.

Во-вторых, нужно описать информацию, содержащуюся в файле. Для этого нажимаем на название импорта в списке файлов импорта или нажимаем кнопку «Сохранить и выйти» при добавлении и редактировании файла импорта.

Поля файла имеют следующие характеристики:

- **Название** – краткое описание поля;
- **Тип** – обновляемая информация:
  - **Идентификатор (уникальный код)** – уникальный набор символов. По этому полю при повторном импорте определяется товар или категория для обновления. Также это поле используется для определения родителя у категорий и поля категория у товаров;
  - **Название** – поле «Название»;
  - **Артикул** – поле «Артикул», используется у товаров;
  - **Анонс** – поле «Анонс»;
  - **Текст** – поле «Описание»;
  - **Цена** – поле «Цена», используется у товаров;
  - **Количество** – поле «Количество», используется у товаров;
  - **Категория** – идентификатор категории из файла импорта категорий, используется у товаров. Чтобы категория для товара определилась правильно, нужно сначала импортировать категории;
  - **Пропуск** – неиспользуемая информация;
  - **Родитель** – идентификатор родителя (должен соответствовать данным из поля первого типа), используется у категорий;
  - **Дополнительная характеристика** – характеристика товара из конструктора характеристик, используется у товаров;
- **Дополнительная характеристика** – список характеристик для поля с типом «Дополнительная характеристика»;
- **Категория** – файл импорта;
- **Сортировка: установить перед** – редактирование порядка следования поля.

В-третьих, нужно загрузить файл импорта.

[Примеры импорта товаров и категорий](#)

## Настройки

Можно сохранять разные конфигурации модуля для разных страниц, к которым прикреплен модуль.

### Основные настройки

**Формат цены** — возможность указать формат отображения цены (количество знаков после запятой, разделитель целых, разделитель десятков).

**Искать по цене** – параметр позволяет выводить в модуле поиска по товарам поиск по цене.

**Искать по описанию** – параметр позволяет выводить в модуле поиска по товарам поиск по описанию.

**Искать по названию** – параметр позволяет выводить в модуле поиска по товарам поиск по названию.

**Искать по артикулу** – параметр позволяет выводить в модуле поиска по товарам поиск по артикулу.

**Использовать категории** – параметр позволяет включить/отключить категории товаров.

**Количество товаров в списке категорий** – количество товаров, выводимых в списке категорий на главной странице модуля.

**Показывать товары подкатегорий** – позволяет показывать в категории верхнего уровня не только товары, принадлежащие текущей категории, но и товары подкатегорий, принадлежащих данной категории.

**Количество выводимых на странице похожих товаров** – количество похожих товаров, выводимых на странице товара.

**Количество товаров на странице** – количество товаров, показываемых на одной странице при открытии раздела каталога или при просмотре всех товаров (если группы не используются), прежде чем появится строка навигации.

**Показывать комментарии к товарам** – подключение модуля «Комментарии» к товарам. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [«Модуль Комментарии»](#).

**Показывать комментарии к категориям товаров** – подключение модуля «Комментарии» к группам товаров. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [«Модуль Комментарии»](#).

**Подключить теги** – подключение модуля «Теги» к товарам. Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [«Модуль Теги»](#).

**Показывать рейтинг товаров** – подключение модуля «Рейтинг» к товарам. Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [«Модуль Рейтинг»](#).

**Валюта** – сокращенное наименование валюты, используемой при указании цены товара.

**Покупать могут только зарегистрированные** – запрещает покупать товары неавторизованным пользователям, убирает для них все кнопки «Купить» и корзину.

**Учитывать остатки товаров на складе** – товару можно задать свойство «Количество», которое будет уменьшаться при покупке товара. Товар с нулевым количеством нельзя купить.

**Не заполнены обязательные поля** – сообщение об ошибке заполнения формы «Оформление заказа».

## **Изображения**

**Использовать изображения** – позволяет включить/отключить загрузку изображений к товару.

**Размер большого изображения для категории** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер маленького изображения для категории** – размер изображения предпросмотра.

**Отображение изображений в списке товаров** – параметр принимает значения:

- нет (отключает отображение фотографии в списке);
- показывать одну фотографию;
- показывать все фотографии.

**Использовать изображения для категории** – параметр позволяет включить/отключить загрузку изображений для категорий.

**Размер большого изображения для категории** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер маленького изображения для категории** – размер изображения предпросмотра.

**Отображение изображений категорий в списке категорий** – параметр принимает значения:

- нет (отключает отображение изображений в списке);
- показывать одно изображение;
- показывать все изображения.

**Использовать анимацию при увеличении изображений** – параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений.

**Максимальный объем загружаемых файлов** – параметр показывает максимально допустимый объем загружаемых файлов, установленный в настройках хостинга.

**Накладывать водяной знак** – параметр позволяет включить/отключить наложение водяного знака на изображение.

**Расположение водяного знака** – расположение водяного знака в пикселях от левого края или от центра и от нижнего края или от центра.

**Использовать водяной знак только для большого изображения** – отключает наложение водяного знака на изображение предпросмотра.

**Применить настройки ко всем ранее загруженным изображениям** – позволяет перекодировать размер уже загруженных изображений. Размер вновь созданных изображений не может быть больше размера большой копии уже загруженного изображения.

## Яндекс Маркет

**Подключить Яндекс Маркет** – позволяет импортировать информацию о товарах в систему «Яндекс Маркет». Информация выдается в формате YML по адресу:

**<http://www.site.ru/modules/shop/shop.yandex.php>**

Все подробности и требования к магазинам смотрите на сайте «Яндекс Маркет» (<http://partner.market.yandex.ru/legal/it/>).

**Короткое название магазина** – Название магазина для системы Яндекс Маркет. Не должно содержать более 20 символов. Нельзя использовать слова, не имеющие отношения к наименованию магазина («лучший», «дешевый»), указывать номер телефона и т. п. Название магазина, должно совпадать с фактическим названием магазина, которое публикуется на сайте).

**Валюта** – валюта для системы «Яндекс Маркет».

**Курс к рублю** – если выбрана валюта, отличная от RUR (рубли), то необходимо выбрать курс: свой курс (ниже появится поле, в котором необходимо указать свой курс), курс ЦБ РФ или курс ЦБ РФ + % (ниже появится поле, в котором необходимо указать величину процента). См. инструкцию «Яндекс Маркет».

**Выгружать категории в Яндекс.Маркет** – позволяет выбрать какие категории выгружать в Яндекс.Маркет: все или только помеченные (появляется галочка при редактировании категории).

**Выгружать товары в Яндекс.Маркет** – позволяет выбрать какие товары выгружать в Яндекс.Маркет: все или только помеченные (появляется галочка при редактировании товара).

**Основная ставка** – см. инструкцию «Яндекс Маркет».

**Ставка для карточек** – см. инструкцию «Яндекс Маркет».

## Методы оплаты

**Подключить платежные системы к заказу** – если параметр выбран, то при оформлении заказа пользователю будет предложено выбрать способ оплаты.

**Описание платежа** – Описание платежа для платежных систем.

Пример: *Оплата заказа %order на сайте site.ru.*

Где %order – номер заказа.

Методы:

**Наличными курьеру:** описание – сообщение, которые будет выводиться пользователю, если он выберет данный способ оплаты.

**Webmoney:** кошелек – выданный в системе Webmoney, секретный ключ – указанный в настройках системы Webmoney, описание – сообщение, которые будет выводиться пользователю, если он выберет данный способ оплаты.

**Roboxchange:** логин, пароль 1, пароль 2 – указанные в настройках системы Roboxchange, описание – сообщение, которые будет выводиться пользователю, если он выберет данный способ оплаты.

**Банковские платежи:** реквизиты организации – Наименование организации, КПП, ИНН, Сокр. наим. налогового органа, Код ОКАТО, ОГРН, Расч. счет, Банк, БИК, Кор. счет, КБК, Адрес, Руководитель предприятия, Главный бухгалтер, описание –

сообщение, которые будет выводиться пользователю, если он выберет данный способ оплаты.

**Платеж успешно принят** – сообщение, которое выводится пользователю при успешной оплате.

**Платеж не принят** – сообщение, которое выводится пользователю при неудачной оплате.

**Параметры настройки WebMoney:**

ResultURL:

site.ru/order/payment/wm/result/

SuccessURL:

site.ru/order/payment/wm/success/

FailURL:

site.ru/order/payment/wm/fail/

**Параметры настройки Roboxchange:**

ResultURL:

site.ru/order/payment/robox/result/

SuccessURL:

site.ru/order/payment/robox/success/

FailURL:

site.ru/order/payment/robox/fail/

**Почтовые сообщения**

**Спасибо за Ваш заказ! В ближайшее время мы с Вами свяжемся для подтверждения заказа** – сообщение, получаемое пользователем по окончании оформления заказа.

**Оформление платежа Шаг 2** – сообщение, которое выводится на втором шаге оформления заказа.

**Тема письма пользователю о новом заказе**

Пример: *Вы оформили заказ на сайте %title (%url)*

Где:

– %title – название сайта;

– %url – адрес сайта.

**Сообщение пользователю о новом заказе**

Пример:

*Здравствуйтe!<br>*

*Вы оформили заказ на сайте %title (%url):<br><br>*

*Номер заказа: %id<br>*

*%order<br>*

<br>

Спасибо за Ваш заказ! В ближайшее время мы с Вами свяжемся для подтверждения заказа.

Где:

- %title – название сайта;
- %url – адрес сайта;
- %order – таблица заказа;
- %id – номер заказа.

**Е-mail, указываемый в обратном адресе пользователю** – возможные значения: e-mail, указанный в параметрах сайта; другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).

**Тема письма администратору о новом заказе**

Пример: %title (%url). Новый заказ

Где:

- %title – название сайта;
- %url – адрес сайта.

**Текст письма администратору о новом заказе**

Пример:

Здравствуйте, администратор сайта %title (%url)!<br>На сайте появился новый заказ номер %id: %order<br><br>%message

Где:

- %title – название сайта;
- %url – адрес сайта;
- %message – поля формы «Оформление заказа»;
- %order – таблица заказа;
- %id – номер заказа.

**Е-mail для уведомлений администратора** – возможные значения: e-mail, указанный в параметрах сайта; другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).

## CSS-форматирование

Общие стили:

.shop\_name{} – название товара

.shop\_img{} – изображение товара

.shop\_price{} – цена товара

.shop\_price .shop\_price\_value{} – значение цены товара

.shop\_price .shop\_price\_currency{} – валюта цены товара

`.shop_old_price{}` – цена товара без скидки  
`.shop_old_price .shop_price_value{}` – значение цены товара без скидки  
`.shop_old_price .shop_price_currency{}` – валюта цены товара без скидки  
`.shop_discount{}` – скидка на товар  
`.shop_discount .shop_discount_value{}` – значение скидки на товар  
`.shop_article{}` – артикул товара  
`.shop_article .shop_article_value{}` – значение артикула товара  
`.shop_param{}` – дополнительная характеристика товара  
`.shop_param .shop_param_value{}` – значение дополнительной характеристики товара  
`.shop_anons{}` – краткое описание товара  
`.shop_text{}` – полное описание товара

Форма покупки товара:

`.shop_form{}` – форма покупки товара  
`.shop_form .shop_depend_param{}` – название и значение параметра влияющего на цену  
`.shop_form .shop_price{}` – цена товара для параметра, влияющего на цену  
`.shop_form .shop_price .shop_price_value{}` – значение цены товара для параметра, влияющего на цену  
`.shop_form .shop_price .shop_price_currency{}` – валюта цены товара для параметра, влияющего на цену  
`.shop_form .shop_old_price{}` – цена товара без скидки для параметра, влияющего на цену  
`.shop_form .shop_old_price .shop_price_value{}` – значение цены товара без скидки для параметра, влияющего на цену  
`.shop_form .shop_old_price .shop_price_currency{}` – валюта цены товара без скидки для параметра, влияющего на цену  
`.shop_form .shop_form_param{}` – параметр, доступный к выбору при заказе  
`.shop_form .inpselect{}` – список значений параметра, доступного к выбору при покупке  
`.shop_form .inpnum{}` – поле для ввода количества покупаемого товара  
`.shop_form .shop_no_buy{}` – надпись «Товар временно отсутствует»  
`.shop_form .button{}` – кнопка «Купить»

Список товаров:

`.shop_list{}` – список товаров  
`.shop_list .shop{}` – товар в списке  
`.shop_list .shop_name{}` – название товара в списке  
`.shop_list .shop_img{}` – изображение товара в списке  
`.shop_list .shop_price{}` – цена товара в списке  
`.shop_list .shop_old_price{}` – цена товара без скидки в списке

.shop\_list .shop\_discount{} – скидка товара в списке  
.shop\_list .shop\_article{} – артикул товара в списке  
.shop\_list .shop\_param{} – дополнительная характеристика товара в списке  
.shop\_list .shop\_anons{} – краткое описание товара в списке  
.shop\_list .shop\_form{} – форма покупки товара в списке товаров  
.shop\_list .shop\_cat\_anons{} – краткое описание категории товаров  
.shop\_list .shop\_cat\_text{} – полное описание категории товаров  
.shop\_list .shop\_cat\_img{} – изображение, прикрепленное к категории товаров  
.shop\_list .shop\_cat\_all\_img{} – все изображения, прикрепленные к категории товаров  
.shop\_list .shop\_cat\_link{} – ссылка на вложенную категорию товаров  
.shop\_list .show\_all{} – ссылка на все элементы в категории  
.shop\_order{} – форма сортировки товаров  
.shop\_order a{} – ссылки в форме сортировки товаров  
.shop\_order .active{} – текущее/активное направление сортировки  
.shop\_list .previous\_next\_links{} – ссылки на предыдущую, следующую категории  
.shop\_list .previous\_link{} – ссылка на предыдущую категорию  
.shop\_list .next\_link{} – ссылка на следующую категорию

Страница товара:

.shop\_id{} – страница товара  
.shop\_id .shop\_price{} – цена товара  
.shop\_id .shop\_old\_price{} – цена товара без скидки  
.shop\_id .shop\_discount{} – скидка на товар  
.shop\_id .shop\_article{} – артикул товара  
.shop\_id .shop\_param{} – дополнительная характеристика  
.shop\_id .shop\_anons{} – краткое описание товара  
.shop\_id .shop\_text{} – полное описание товара  
.shop\_id .shop\_form{} – форма покупки товара  
.shop\_id .shop\_all\_img{} – все изображения товара  
.shop\_rel{} – блок похожих товаров  
.shop\_rel .shop\_img{} – изображение похожего товара  
.shop\_rel .shop\_price{} – цена похожего товара  
.shop\_rel .shop\_old\_price{} – цена похожего товара без скидки  
.shop\_rel .shop\_discount{} – скидка на похожий товар  
.shop\_rel .shop\_article{} – артикул похожего товара  
.shop\_rel .shop\_param{} – дополнительная характеристика похожего товара  
.shop\_rel .shop\_anons{} – краткое описание похожего товара  
.shop\_rel .shop\_form{} – форма покупки похожего товара

`.shop_id .previous_next_links` – ссылки на предыдущий, следующий товары

`.shop_id .previous_link` – ссылка на предыдущий товар

`.shop_id .next_link` – ссылка на следующий товар

Шаблонная функция `show_block`:

`.shop_block` – шаблонная функция `show_block`

`.shop_block h2` – заголовок

`.shop_block .shop` – товар в блоке

`.shop_block .shop_name` – название товара

`.shop_block .shop_img` – изображение товара

`.shop_block .shop_price` – цена товара

`.shop_block .shop_old_price` – цена товара без скидки

`.shop_block .shop_discount` – скидка товара

`.shop_block .shop_anons` – краткое описание товара

`.shop_block .shop_text` – полное описание товара

`.shop_block .shop_form` – форма покупки товара

`.shop_block .show_all` – ссылка на все товары

Шаблонная функция `show_search`:

`.shop_search` – шаблонная функция `show_search`

`.shop_search .infoform` – заголовок группы полей

`.shop_search .infofield` – подпись к полю в форме

`.shop_search .inptext` – текстовые поля

`.shop_search .inpshort` – цифровые поля

`.shop_search .inpcheckbox` – поля «Чекбокс»

`.shop_search .button` – кнопка «Оправить»

`.shop_search_name` – поиск по названию

`.shop_search_name .infofield` – заголовок «Название:»

`.shop_search_name .inptext` – текстовое поле для ввода названия товара

`.shop_search_description` – поиск по описанию

`.shop_search_description .infofield` – заголовок «Описание:»

`.shop_search_description .inptext` – текстовое поле для ввода описания товара

`.shop_search_price` – поиск по цене

`.shop_search_price .infofield` – заголовок «Цена:»

`.shop_search_price .inpnum` – текстовое поле для ввода цены товара

`.shop_search_paramNUM` – поиск по параметру с номером NUM

`.shop_search_paramNUM .infofield` – название параметра с номером NUM

`.shop_search_paramNUM .infoform` – заголовок – параметр с номером NUM

`.shop_search_paramNUM .inptext` – текстовое поле для ввода значения параметра с

номером NUM

`.shop_search_paramNUM .inpnum` – текстовое поле для ввода цифрового значения параметра с номером NUM

`.shop_search_paramNUM .inpcheckbox` – чекбокс для выбора значения параметра с номером NUM

### Корзина

`table.basket` – таблица «Корзина»

`table.basket th` – шапка таблицы «Корзина»

`table.basket th.basket_first_th` – первая ячейка в шапке таблицы

`table.basket th.basket_last_th` – последняя ячейка в шапке таблицы

`table.basket td` – ячейки в таблице «Корзина»

`table.basket td.basket_img` – изображение товара

`table.basket td.basket_name` – название товара

`table.basket td.basket_count` – количество товара

`table.basket td.basket_price` – цена товара

`table.basket td.basket_summ` – стоимость товаров

`table.basket td.basket_delete` – чекбокс «Удалить»

`table.basket tr.basket_last_tr` – последняя итоговая строка корзины

`table.basket td.basket_total` – надпись «Итого»

`table.basket td.basket_last_td` – последняя ячейка в итоговой строке корзины

`.to_order` – ссылка «Оформление заказа»

Шаблонная функция [show\\_basket](#):

`.basket_block` – шаблонная функция `show_basket`

`.basket_block .basket_count` – количество купленных товаров

`.basket_block .basket_summ` – стоимость товаров

### Оформление заказа

`.order_form` – оформление заказа

`.order_form .infofield` – подпись к полю в форме

`.order_form .inptext` – текстовые поля

`.order_form .inpselect` – выпадающий список

`.order_form .inpnum` – цифровые поля

`.order_form .button` – кнопка «Оформить»

`.order_form_param_text` – описание поля

`.order_form_paramNUM` – поле с номером NUM

`.order_form_paramNUM .infofield` – название параметра с номером NUM

`.order_form_paramNUM .inptext` – текстовое поле для ввода текстового значения параметра с номером NUM

`.order_form_paramNUM .inpnum` – текстовое поле для ввода цифрового значения

параметра с номером NUM

`.order_form_paramNUM.inpselect{}` – список для выбора значения параметра с номером NUM

`.order_form_paramNUM.inpcheckbox{}` – чекбокс для выбора значения параметра с номером NUM

## Файлы

1. `modules/shop/shop.php` – контроллер;
2. `modules/shop/shop.model.php` – вывод товаров и разделов каталога в пользовательской части сайта;
3. `modules/shop/shop.view.php` – шаблон вывода в пользовательской части сайта;
4. `modules/shop/shop.ajax.php` – добавление товаров в корзину;
5. `modules/shop/shop.sitemap.php` – карта сайта модуля для файла `sitemap.xml`;
6. `modules/shop/shop.install.php` – установка модуля;
7. `modules/shop/shop.yandex.php` – файл экспорта данных в систему «Яндекс.Маркет».
8. `modules/basket/basket.php` – контроллер модуля «Корзина»;
9. `modules/basket/basket.model.php` – формирование данных для вывода модуля «Корзина»;
10. `modules/basket/basket.ajax.php` – изменения данных корзины;
11. `modules/basket/basket.view.php` – шаблон вывода в пользовательской части модуля «Корзина», шаблон функций `show_basket`;
12. `modules/basket/basket.js` – JS-сценарий для пересчета корзины;
13. `modules/order/order.php` – контроллер модуля «Оформление заказа»;
14. `modules/order/order.model.php` – формирование данных для вывода модуля «Оформление заказа»;
15. `modules/order/order.view.php` – шаблон вывода в пользовательской части модуля;
16. `modules/order/order.ajax.php` – обработка данных, получаемых от пользователя при оформлении заказа;
17. `modules/order/order.payment.php` – вспомогательные страницы, обеспечивающие работу разных методов оплаты заказа;
18. `modules/order/webmoney.php` – обработка запросов платежной системы «WebMoney»;
19. `modules/order/roboboxchange.php` – обработка запросов платежной системы «Roboboxchange»;
20. `modules/order/non_cash.php` – генерирование счета при безналичной системе оплаты;
21. `modules/order/num2str.php` – функции для перевода чисел в слова;
22. `modules/shop/admin/shop.admin.php` – редактирование товаров;
23. `modules/shop/admin/shop.admin.js` – редактирование товаров, JS-сценарий;

24. *modules/shop/admin/shop.admin.view.php* – шаблон вывода магазина в административной части;
25. *modules/shop/admin/shop.admin.ajax.php* – обработка Ajax-запросов в административной части модуля;
26. *modules/shop/admin/shop.admin.menu.php* – формирование списка товаров и категорий товаров в административной части модуля «Меню».
27. *modules/shop/admin/shop.admin.category.php* – редактирование категорий товаров;
28. *modules/shop/admin/shop.admin.param.php* – редактирование характеристик товаров;
29. *modules/shop/admin/shop.admin.param.js* – редактирование характеристик товаров JS-сценарий;
30. *modules/shop/admin/shop.admin.discount.php* – редактирование скидок;
31. *modules/shop/admin/shop.admin.discount.js* – редактирование скидок, JS-сценарий;
32. *modules/shop/admin/shop.admin.order.php* – редактирование заказов;
33. *modules/shop/admin/shop.admin.orderparam.php* – редактирование полей формы «Оформление заказа»;
34. *modules/shop/admin/shop.admin.orderparam.js* – редактирование полей формы «Оформление заказа», JS-сценарий;
35. *modules/shop/admin/shop.admin.ordercount.php* – отчеты о продажах;
36. *modules/shop/admin/shop.admin.import.php* – администрирование импорта;
37. *modules/shop/admin/shop.admin.import.element.php* – редактирование структуры файла импорта, импорт;
38. *modules/shop/admin/shop.admin.import.element.js* – редактирование структуры файла импорта, импорт, JS-сценарий;
39. *modules/shop/admin/shop.admin.import.category.php* – редактирование списка файлов импорта;
40. *modules/shop/admin/shop.admin.config.php* – конфигурация модуля.

# Фотогалерея

- [Альбомы](#)
- [Фотографии](#)
- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль состоит из 2х частей: альбомы и фотографии.

Модуль «Фотогалерея» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка фотографий, категорий и настроек) по принадлежности к странице.

## Альбомы

Альбомы фотографий могут иметь неограниченное число вложенных альбомов. При необходимости альбомы можно не использовать. Для этого нужно отключить параметр «Использовать альбомы» в настройках модуля.

Если перед названием альбома имеется знак +, то этому альбому принадлежат вложенные альбомы. Добавить вложенные альбомы можно, нажав на знак  рядом с названием альбома.

Порядок отображения альбомов можно редактировать с помощью кнопок  .

Кнопка  служит для удаления альбома. Кнопка  служит для быстрой публикации альбома (или, наоборот,  блокирования). При удалении альбома удаляются все вложенные альбомы и фотографии.

Альбом имеют следующие характеристики.

Основные:

- **Название;**
- **Показывать на сайте;**
- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Анонс** – краткое описание категории, для публикации в списке;
- **Описание.**

Дополнительные:

- **Номер** – номер альбома;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет автоматически сформирован из Названия альбома - Названия страницы сайта - Названия сайта;
- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords»,

- используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
  - **Псевдоссылка** – ссылка на страницу альбома будет генерироваться следующим образом: <http://www.site.ru/псевдоссылка/>. Если псевдоссылка не указана адрес страницы будет: [http://www.site.ru/псевдоссылка\\_страницы/catНомер\\_альбома/](http://www.site.ru/псевдоссылка_страницы/catНомер_альбома/). При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы. При редактировании можно это исправить;
  - **Вложенность: принадлежит** – альбом верхнего уровня;
  - **Раздел сайта** – принадлежность к странице сайта;
  - **Сортировка: установить перед** – редактирование порядка отображения пункта. Поле доступно для редактирования только для элементов, отображаемых на сайте;
  - **Не показывать на карте сайта** – скрывает отображение ссылки на альбом в файле sitemap.xml и [модуле «Карта сайта»](#).

Если в конфигурации модуля включен параметр «Показывать комментарии к альбомам», то при редактировании альбома будет отображена активная ссылка на комментарии или надпись «Комментариев нет» при их отсутствии.

## Фотографии

Порядок отображения фотографий можно редактировать с помощью кнопок  . Если подключены альбомы фотографий, то сортировать фотографии можно только в пределах альбома (в фильтре «Категория» нужно выбрать альбом). Кнопка  служит для удаления фотографии. Кнопка  служит для быстрой публикации фотографии (или, наоборот,  блокирования).

Фотографии имеют следующие характеристики.

Основные:

- **Название;**
- **Показывать на сайте** – публикация на сайте, активность;
- **Фотография** – загрузка одной фотографии. Фотографии загружаются в двух вариантах: маленьком (предпросмотр) и большом (увеличенная версия). Также для отражения в административной части сайта генерируются дополнительные изображения (50x50, качество 70). Качество и размер загружаемых фотографий для пользовательской части сайта настраиваются в конфигурации модуля;
- **Анонс** – краткое описание фотографии;
- **Альбом** – помимо основной можно указать дополнительные категории;
- **Описание** – полное описание фотографии.

Дополнительные:

- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);

- **Номер** – номер фотографии;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет автоматически сформирован из Названия фотографии - Названия страницы сайта - Названия сайта;
- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords», используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу фотографии будет генерироваться следующим образом: *http://www.site.ru/псевдоссылка/*. Если псевдоссылка не указана адрес страницы будет: *http://www.site.ru/псевдоссылка\_страницы/catНомер\_категории/showНомер\_фотографии/*. При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы и категории фотографии. При редактировании можно это исправить;
- **Раздел сайта** – принадлежность к странице сайта;
- **Сортировка: установить перед** – редактирование порядка отображения пункта. Поле доступно для редактирования только для элементов, отображаемых на сайте;
- **Не показывать на карте сайта** – скрывает отображение ссылки на фотографию в файле *sitemap.xml*.

Если в конфигурации модуля включен параметр «Показывать комментарии к фотографиям», то при редактировании фотографии будет отображена активная ссылка на комментарии к фотографии или надпись «Комментариев нет» при их отсутствии.

Если в конфигурации модуля включен параметр «Показывать рейтинг фотографий», то при редактировании фотографии будет отображена активная ссылка на рейтинг фотографии с указанием среднего балла и количества голосов или надпись «Рейтинг: нет голосов», если голосов за фотографию нет.

Если в конфигурации модуля включен параметр «Подключить теги», то при редактировании фотографии появится возможность редактирования тегов, прикрепленных к фотографии.

Шаблонная функция [show\\_block](#).

## Настройки

**Использовать альбомы** – параметр позволяет включить/отключить категории фотографии.

**Размер большого изображения** – размер фотографии, отображаемый в пользовательской части сайта при увеличении фотографии предпросмотра.

**Размер маленького изображения** – размер маленькой фотографии предпросмотра, отображаемой в пользовательской части сайта.

**Открывать фотографию на отдельной странице** – подключает отдельную страницу для каждой фотографии.

**Использовать анимацию при увеличении изображений** – параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений.

**Количество фотографий в списке альбомов** – количество фотографий, выводимых в списке альбомов на главной странице модуля.

**Количество фотографий на странице** – количество фотографий, показываемых на одной странице при открытии альбома или при просмотре всех фотографий (если альбомы не используются).

**Показывать фотографии вложенных альбомов** – позволяет показывать в альбоме верхнего уровня не только фотографии, принадлежащие текущему альбому, но и фотографии вложенных альбомов.

**Показывать комментарии к фотографиям** – подключение модуля «Комментарии» к фотографиям. Если параметр активен, то в пользовательской части сайта появляется дополнительная страница при клике на фотографию, где можно увеличить фотографию и посмотреть комментарии к ней. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [«Модуль Комментарии»](#).

**Показывать комментарии к альбомам** – подключение модуля «Комментарии» к альбомам фотографий. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [«Модуль Комментарии»](#).

**Подключить теги** – подключение модуля «Теги» к фотографиям. Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [«Модуль Теги»](#).

**Показывать рейтинг фотографий** – подключение модуля «Рейтинг» к фотографиям. Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [«Модуль Рейтинг»](#).

## CSS-форматирование

Общие стили:

`.photo_name{}` – название фотографии

`.photo_img{}` – изображение

`.photo_size{}` – размер фотографии

`.photo_anons{}` – краткое описание фотографии

`.photo_text{}` – полное описание фотографии

Список фотографий:

`.photo_list{}` – список фотографий

`.photo_list .photo{}` – фотография в списке

`.photo_list .photo_name{}` – название фотографии в списке

`.photo_list .photo_img{}` – изображение фотографии в списке

`.photo_list .photo_size{}` – размер фотографии в списке

`.photo_list .photo_anons{}` – краткое описание фотографии в списке

`.photo_list .photo_cat_anons{}` – краткое описание альбома фотографий

`.photo_list .photo_cat_text{}` – полное описание альбома фотографий

`.photo_list .photo_cat_link{}` – ссылка на вложенные альбомы фотографий  
`.photo_list .show_all{}` – ссылка на все фотографии в альбоме  
`.photo_list .previous_next_links{}` – ссылки на предыдущий, следующий альбомы  
`.photo_list .previous_link{}` – ссылка на предыдущий альбом  
`.photo_list .next_link{}` – ссылка на следующий альбом

Страница фотографии:

`.photo_id{}` – страница фотографии  
`.photo_id .photo_img{}` – фотография  
`.photo_id .photo_size{}` – размер  
`.photo_id .photo_anons{}` – краткое описание фотографии  
`.photo_id .photo_text{}` – полное описание фотографии  
`.photo_id .previous_next_links{}` – ссылки на предыдущую, следующую фотографии  
`.photo_id .previous_link{}` – ссылка на предыдущую фотографию  
`.photo_id .next_link{}` – ссылка на следующую фотографию

Список фотографий, прикрепленных к тегу:

`.tags_list .photo{}` – фотография в списке  
`.tags_list .photo_name{}` – название фотографии в списке  
`.tags_list .photo_img{}` – изображение фотографии в списке  
`.tags_list .photo_size{}` – размер фотографии в списке  
`.tags_list .photo_anons{}` – краткое описание фотографии в списке

Шаблонная функция `show_block`:

`.photo_block{}` – шаблонная функция `show_block`  
`.photo_block h2{}` – заголовок  
`.photo_block .photo{}` – фотография в блоке  
`.photo_block .photo_name{}` – название фотографии в блоке  
`.photo_block .photo_img{}` – изображение фотографии в блоке  
`.photo_block .photo_size{}` – размер фотографии в блоке  
`.photo_block .photo_anons{}` – краткое описание фотографии в блоке  
`.photo_block .show_all{}` – ссылка на все фотографии

## Файлы

1. `modules/photo/photo.php` – контроллер модуля;
2. `modules/photo/photo.model.php` – вывод фотографий и альбомов фотографий в пользовательской части сайта;
3. `modules/photo/photo.view.php` – шаблон вывода в пользовательской части сайта;
4. `modules/photo/photo.sitemap.php` – карта сайта модуля для файла `sitemap.xml`;
5. `modules/photo/photo.install.php` – установка модуля;

6. *modules/photo/photo.js* – JS-сценарий листания фотографий;
7. *modules/photo/admin/photo.admin.php* – редактирование фотографий;
8. *modules/photo/admin/photo.admin.category.php* – редактирование альбомов фотографий;
9. *modules/photo/admin/photo.admin.menu.php* – формирование списка фотографий и альбомов фотографий в административной части модуля «Меню»;
10. *modules/photo/admin/photo.admin.config.php* – конфигурация модуля.

## Статьи

- [Категории](#)
- [Статьи](#)
- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль «Статьи» отображает статьи.

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль «Статьи» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка статей, категорий и настроек) по принадлежности к странице.

Модуль состоит из 2х частей: категории статей и статьи.

## Категории

Категории статей могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в конфигурации модуля.

Если перед названием категории имеется знак +, то этой категории принадлежат подкатегории. Добавить подкатеорию к категории можно, нажав на знак  рядом с названием категории.

Порядок отображения категорий можно редактировать с помощью кнопок  .

Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот,  блокирования). При удалении категории удаляются все вложенные подкатегории и статьи.

Категории имеют следующие характеристики.

Основные:

- **Название;**
- **Показывать на сайте;**
- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Анонс** – краткое описание категории, для публикации в списке;
- **Описание** – описание категории.

Дополнительные:

- **Номер** – номер категории;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет автоматически сформирован из Названия категории - Названия раздела - Названия сайта;

- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords», используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу категории будет генерироваться следующим образом: *http://www.site.ru/псевдоссылка/*. Если псевдоссылка не указана адрес страницы будет: *http://www.site.ru/псевдоссылка\_страницы/catНомер\_категории/*. При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы. При редактировании можно это исправить;
- **Вложенность: принадлежит** – категория верхнего уровня;
- **Раздел сайта** – принадлежность к странице сайта;
- **Сортировка: установить перед** – редактирование порядка отображения пункта. Поле доступно для редактирования только для элементов, отображаемых на сайте;
- **Не показывать на карте сайта** – скрывает отображение ссылки на категорию в файле sitemap.xml и [модуле «Карта сайта»](#).

Если в конфигурации модуля включен параметр «Показывать комментарии к категориям», то при редактировании категории будет отображена активная ссылка на комментарии к категории или надпись «Комментариев нет» при их отсутствии.

## Статьи

В списке статьи выводятся с сортировкой по важности и новизне.

Кнопка  служит для удаления статьи. Кнопка  служит для быстрой публикации статьи (или, наоборот,  блокирования).

Статьи имеют следующие характеристики.

Основные:

- **Название;**
- **Дата** – статьи, старше текущей даты начнут отображаться на сайте, начиная с указанной даты;
- **Изображения** – возможность загрузки нескольких изображений, если в конфигурации модуля включен параметр «Использовать изображения». Изображение загружаются в двух вариантах: маленьком (предпросмотр) и большом (увеличенная версия). Также для отражения в административной части сайта генерируются дополнительные изображения (50x50, качество 70). Качество и размер загружаемых изображений для пользовательской части сайта указываются в настройках модуля;
- **Показывать на сайте** – публикация на сайте, активность;
- **Категория** – помимо основной можно указать дополнительные категории;
- **Анонс** – краткое содержание статьи, для публикации в списке;
- **Описание** – содержание статьи.

Дополнительные:

- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Номер** – номер статьи;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет автоматически сформирован из Названия страницы - Названия раздела - Названия сайта;
- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords», используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу статьи будет генерироваться следующим образом: *http://www.site.ru/псевдоссылка/*. Если псевдоссылка не указана адрес страницы будет: *http://www.site.ru/псевдоссылка\_страницы/catНомер\_категории/showНомер\_с\_таблицы/*. При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы и категории статьи. При редактировании можно это исправить;
- **Раздел сайта** – принадлежность к странице сайта;
- **Важно (всегда сверху)** – статья будет всегда сверху в списке всех статей, независимо от даты;
- **Не показывать на карте сайта** – скрывает отображение ссылки на статью в файле *sitemap.xml*.

Если в конфигурации модуля включен параметр «Показывать комментарии к статьям», то при редактировании статьи будет отображена активная ссылка на комментарии к статье или надпись «Комментариев нет» при их отсутствии.

Если в конфигурации модуля включен параметр «Показывать рейтинг статей», то при редактировании статьи будет отображена активная ссылка на рейтинг статей с указанием среднего балла и количества голосов или надпись «Рейтинг: нет голосов», если голосов за статью нет.

Если в конфигурации модуля включен параметр «Подключить теги», то при редактировании статьи появится возможность редактирования тегов, прикрепленных к статье.

Шаблонная функция: [show\\_block](#).

## Настройки

Можно сохранять разные конфигурации модуля для разных страниц, к которым прикреплен модуль.

**Использовать категории** – параметр позволяет включить/отключить категории статей.

**Количество статей на странице** – количество одновременно выводимых статей в списке статей, после чего начинается строка навигации.

**Количество статей в списке категорий** – количество статей, выводимых в списке категорий на главной странице модуля.

**Показывать статьи подкатегорий** – этот параметр позволяет показывать в категории верхнего уровня не только статьи, принадлежащие текущей категории, но и статьи подкатегорий, принадлежащих данной категории.

**Использовать изображения** – параметр позволяет включить/отключить загрузку изображений к статье.

**Размер большого изображения** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер маленького изображения** – размер изображения предпросмотра.

**Отображение изображений в списке статей** – параметр принимает значения:

- нет (отключает отображение изображений в списке);
- показывать одно изображение;
- показывать все изображения.

**Использовать изображения для категории** – параметр позволяет включить/отключить загрузку изображений для категорий.

**Размер большого изображения для категории** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер маленького изображения для категории** – размер изображения предпросмотра.

**Отображение изображений категорий в списке категорий** – параметр принимает значения:

- нет (отключает отображение изображений в списке);
- показывать одно изображение;
- показывать все изображения.

**Использовать анимацию при увеличении изображений** – параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений.

**Максимальный размер загружаемых файлов** – параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.

**Накладывать водяной знак** – параметр позволяет включить/отключить наложение водяного знака на изображение.

**Расположение водяного знака** – расположение водяного знака в пикселях от левого края или от центра и от нижнего края или от центра.

**Использовать водяной знак только для большого изображения** – отключает наложение водяного знака на изображение предпросмотра.

**Применить настройки ко всем ранее загруженным изображениям** – позволяет переконвертировать размер уже загруженных изображений. Размер вновь созданных изображений не может быть больше размера большой копии уже загруженного изображения.

**Формат даты** – позволяет настроить отображение даты в модуле.

**Показывать комментарии к статьям** – подключение модуля «Комментарии». Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в [разделе «Модуль Комментарии»](#).

**Показывать комментарии к категориям** – подключение модуля «Комментарии» к

категориям статей. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в [разделе «Модуль Комментарии»](#).

**Подключить теги** – подключение модуля «Теги» к модулю «Статьи». Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [«Модуль Теги»](#).

**Показывать рейтинг статей** – подключение модуля «Рейтинг». Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в [разделе «Модуль Рейтинг»](#).

## CSS-форматирование

Общие стили:

`.clauses_name{}` – название статьи

`.clauses_date{}` – дата статьи

`.clauses_anons{}` – анонс статьи

`.clauses_text{}` – текст статьи

`.clauses_img{}` – изображение, прикрепленное к статье

`.clauses_all_img{}` – все изображения, прикрепленные к статье

Список статей:

`.clauses_list{}` – список статей

`.clauses_list .clauses{}` – статья в списке

`.clauses_list .clauses_name{}` – название статьи

`.clauses_list .clauses_img{}` – изображение, прикрепленное к статье

`.clauses_list .clauses_date{}` – дата статьи

`.clauses_list .clauses_anons{}` – анонс статьи

`.clauses_list .clauses_cat_anons{}` – краткое описание категории статей

`.clauses_list .clauses_cat_text{}` – полное описание категории статей

`.clauses_list .clauses_cat_img{}` – изображение, прикрепленное к категории статей

`.clauses_list .clauses_cat_all_img{}` – все изображения, прикрепленные к категории статей

`.clauses_list .clauses_cat_link{}` – ссылка на вложенную категорию статей

`.clauses_list .show_all{}` – ссылка на все статьи в категории

`.clauses_list .previous_next_links{}` – ссылки на предыдущую, следующую категории

`.clauses_list .previous_link{}` – ссылка на предыдущую категорию

`.clauses_list .next_link{}` – ссылка на следующую категорию

Страница статьи:

`.clauses_id{}` – страница статьи

`.clauses_id .clauses_date{}` – дата статьи

`.clauses_id .clauses_anons{}` – анонс статьи

`.clauses_id .clauses_text{}` – текст статьи

`.clauses_id .clauses_all_img{}` – все изображения, прикрепленные к статье  
`.clauses_id .show_all{}` – ссылка на все статьи  
`.clauses_id .previous_next_links{}` – ссылки на предыдущую, следующую статью  
`.clauses_id .previous_link{}` – ссылка на предыдущую статью  
`.clauses_id .next_link{}` – ссылка на следующую статью

Шаблонная функция `show_block`:

`.clauses_block{}` – блок статей  
`.clauses_block h2{}` – заголовок  
`.clauses_block .clauses{}` – статья в блоке  
`.clauses_block .clauses_name{}` – название статьи  
`.clauses_block .clauses_img{}` – изображение статьи  
`.clauses_block .clauses_date{}` – дата статьи  
`.clauses_block .clauses_anons{}` – анонс статьи  
`.clauses_block .show_all{}` – ссылка на все статьи

## Файлы

1. `modules/clauses/clauses.php` – контроллер модуля;
2. `modules/clauses/clauses.model.php` – формирование данных для страницы модуля;
3. `modules/clauses/clauses.view.php` – шаблон модуля;
4. `modules/clauses/clauses.install.php` – установка модуля;
5. `modules/clauses/clauses.sitemap.php` – карта модуля для файла `sitemap.xml`;
6. `modules/clauses/admin/clauses.admin.php` – редактирование статей;
7. `modules/clauses/admin/clauses.admin.category.php` – редактирование категорий статей;
8. `modules/clauses/admin/clauses.admin.config.php` – конфигурация модуля;
9. `modules/clauses/admin/clauses.admin.menu.php` – формирование списка статей и категорий статей в административной части модуля «Меню».

# Файловый архив

- [Категории](#)
- [Файлы в модуле](#)
- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль состоит из 2х частей: категории файлов и файлы.

Модуль «Файловый архив» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка файлов, категорий и настроек) по принадлежности к странице.

## Категории

Категории файлов могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в конфигурации модуля.

Если перед названием категории имеется знак +, то этой категории принадлежат подкатегории. Добавить подкатеорию к категории можно, нажав на знак  рядом с названием категории.

Порядок отображения категорий можно редактировать с помощью кнопок  .

Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот,  блокирования). При удалении категории удаляются все вложенные подкатегории и файлы.

Категории имеют следующие характеристики.

Основные:

- **Название;**
- **Показывать на сайте;**
- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Анонс** – краткое описание категории, для публикации в списке;
- **Описание** – описание категории.

Дополнительные:

- **Номер** – номер категории;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет автоматически сформирован из Названия категории - Названия страницы сайта - Названия сайта;
- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords»,

используется для поисковой оптимизации;

- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу категории будет генерироваться следующим образом: *http://www.site.ru/псевдоссылка/*. Если псевдоссылка не указана адрес страницы будет: *http://www.site.ru/псевдоссылка\_страницы/catНомер\_категории/*. При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы. При редактировании можно это исправить;
- **Вложенность: принадлежит** – категория верхнего уровня;
- **Раздел сайта** – принадлежность к странице сайта;
- **Сортировка: установить перед** – редактирование порядка отображения пункта. Поле доступно для редактирования только для элементов, отображаемых на сайте;
- **Не показывать на карте сайта** – скрывает отображение ссылки на категорию в файле sitemap.xml и [модуле «Карта сайта»](#).

Если в конфигурации модуля включен параметр «Показывать комментарии к категориям», то при редактировании категории будет отображена активная ссылка на комментарии к категории или надпись «Комментариев нет» при их отсутствии.

## Файлы

Порядок отображения файлов можно редактировать с помощью кнопок  . Если подключены категории файлов, то сортировать файлы можно только в пределах категории (в фильтре «Категория» нужно выбрать категорию файлов).

Кнопка  служит для удаления файлов. Кнопка  служит для быстрой публикации файлов (или, наоборот,  блокирования).

Файлы имеют следующие характеристики.

Основные:

- **Название;**
- **Файл** – загружаемый файл. Доступные типы файлов изменяются в настройках модуля (параметр «Доступные типы файлов»). Можно указать ссылку для загрузки файла из удаленного источника или ссылку на файл для скачивания;
- **Показывать на сайте** – публикация на сайте, активность;
- **Изображения** – возможность загрузки нескольких изображений, если в конфигурации модуля включен параметр «Использовать изображения». Изображение загружаются в двух вариантах: маленьком (предпросмотр) и большом (увеличенная версия). Также для отражения в административной части сайта генерируются дополнительные изображения (50x50, качество 70). Качество и размер загружаемых изображений для пользовательской части сайта указываются в настройках модуля;
- **Категория** – помимо основной можно указать дополнительные категории;
- **Анонс** – краткое описание файла;
- **Описание** – полное описание файла.

Дополнительные:

- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Номер** – номер файла;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет автоматически сформирован из Названия файла - Названия страницы сайта - Названия сайта;
- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords», используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу статьи будет генерироваться следующим образом: *http://www.site.ru/псевдоссылка/*. Если псевдоссылка не указана адрес страницы будет: *http://www.site.ru/псевдоссылка\_страницы/catНомер\_категорию/showНомер\_файла/*. При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы и категории файла. При редактировании можно это исправить;
- **Раздел сайта** – принадлежность к странице сайта;
- **Сортировка: установить перед** – редактирование порядка отображения пункта. Поле доступно для редактирования только для элементов, отображаемых на сайте;
- **Не показывать на карте сайта** – скрывает отображение ссылки на файл в файле *sitemap.xml*.

Если в конфигурации модуля включен параметр «Показывать комментарии к файлам», то при редактировании файла будет отображена активная ссылка на комментарии к файлам или надпись «Комментариев нет» при их отсутствии.

Если в конфигурации модуля включен параметр «Показывать рейтинг файлов», то при редактировании файлов будет отображена активная ссылка на рейтинг файла с указанием среднего балла и количества голосов или надпись «Рейтинг: нет голосов», если голосов за файл нет.

Если в конфигурации модуля включен параметр «Подключить теги», то при редактировании файла появится возможность редактирования тегов, прикрепленных к файлу.

Шаблонные функции: [show\\_block](#).

## Настройки

**Использовать категории** – параметр позволяет включить/отключить категории файлов.

**Количество файлов на странице** – количество одновременно выводимых файлов в списке файлов.

**Количество файлов в списке категорий** – количество файлов, выводимых в списке категорий на главной странице модуля.

**Показывать файлы подкатегорий** – позволяет показывать в категории верхнего уровня не только файлы, принадлежащие текущей категории, но и файлы подкатегорий, принадлежащих данной категории.

**Использовать изображения** – параметр позволяет включить/отключить загрузку изображений к файлам.

**Размер большого изображения** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер маленького изображения** – размер изображения предпросмотра.

**Отображение изображений в списке файлов** – параметр принимает значения:

- нет (отключает отображение изображений в списке);
- показывать одно изображение;
- показывать все изображения.

**Использовать изображения для категории** – параметр позволяет включить/отключить загрузку изображений для категорий.

**Размер большого изображения для категории** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер маленького изображения для категории** – размер изображения предпросмотра.

**Отображение изображений категорий в списке категорий** – параметр принимает значения:

- нет (отключает отображение изображений в списке);
- показывать одно изображение;
- показывать все изображения.

**Использовать анимацию при увеличении изображений** – параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений.

**Максимальный размер загружаемых файлов** – параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.

**Накладывать водяной знак** – параметр позволяет включить/отключить наложение водяного знака на изображение.

**Расположение водяного знака** – расположение водяного знака в пикселях от левого края или от центра и от нижнего края или от центра.

**Использовать водяной знак только для большого изображения** – отключает наложение водяного знака на изображение предпросмотра.

**Применить настройки ко всем ранее загруженным изображениям** – позволяет перекодировать размер уже загруженных изображений. Размер вновь созданных изображений не может быть больше размера большой копии уже загруженного изображения.

**Показывать комментарии к файлам** – подключение модуля «Комментарии» к файлам. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [«Модуль Комментарии»](#).

**Показывать комментарии к категориям файлов** – подключение модуля «Комментарии» к категориям файлов. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [«Модуль Комментарии»](#).

**Показывать теги файлов** – подключение модуля «Теги» к файлам. Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [«Модуль Теги»](#).

**Показывать рейтинг файлов** – подключение модуля «Рейтинг» к файлам. Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [«Модуль Рейтинг»](#).

### Доступные типы файлов (через запятую)

Пример: *doc, gif, jpg, mpg, pdf, png, txt, zip*

## CSS-форматирование

Общие стили:

[.files\\_name](#){ } – название файлов

[.files\\_anons](#){ } – краткое описание файла

[.files\\_text](#){ } – полное описание файла

[.files\\_img](#){ } – изображение, прикрепленное к файлу

[.files\\_all\\_img](#){ } – все изображения, прикрепленные к файлу

Список файлов:

[.files\\_list](#){ } – список файлов

[.files\\_list .files](#){ } – файл в списке

[.files\\_list .files\\_name](#){ } – название файла в списке

[.files\\_list .files\\_img](#){ } – изображения файла в списке

[.files\\_list .files\\_cat\\_anons](#){ } – краткое описание категории файлов

[.files\\_list .files\\_cat\\_text](#){ } – полное описание категории файлов

[.files\\_list .files\\_cat\\_img](#){ } – изображение, прикрепленное к категории файлов

[.files\\_list .files\\_cat\\_all\\_img](#){ } – все изображения, прикрепленные к категории файлов

[.files\\_list .files\\_cat\\_link](#){ } – ссылка на вложенную категорию файлов

[.files\\_list .show\\_all](#){ } – ссылка на все файлы в категории

[.files\\_list .previous\\_next\\_links](#){ } – ссылки на предыдущую, следующую категории

[.files\\_list .previous\\_link](#){ } – ссылка на предыдущую категорию

[.files\\_list .next\\_link](#){ } – ссылка на следующую категорию

Страница файла:

[.files\\_id](#){ } – страница файла

[.files\\_id .files\\_size](#){ } – размер прикрепленного файла

[.files\\_id .files\\_download](#){ } – ссылка на скачивание прикрепленного файла

[.files\\_id .files\\_anons](#){ } – краткое описание файла

`.files_id .files_text{}` – полное описание файла

`.files_id .files_all_img{}` – все изображения

`.files_id .previous_next_links{}` – ссылки на предыдущий, следующий файлы

`.files_id .previous_link{}` – ссылка на предыдущий файл

`.files_id .next_link{}` – ссылка на следующий файл

Шаблонная функция [show\\_block](#):

`.files_block{}` – шаблонная функция `show_block`

`.files_block h2{}` – заголовок

`.files_block .files{}` – файл в блоке

`.files_block .files_img{}` – изображение файла

`.files_block .files_name{}` – название файла

`.files_block .files_anons{}` – краткое описание файла

`.files_block .show_all{}` – ссылка на все файлы

## Файлы

1. `modules/files/files.php` – контроллер модуля;
2. `modules/files/files.model.php` – вывод файлов и разделов каталога в пользовательской части сайта;
3. `modules/files/files.view.php` – шаблон вывода в пользовательской части сайта;
4. `modules/files/files.sitemap.php` – карта сайта модуля для файла `sitemap.xml`;
5. `modules/files/files.install.php` – установка модуля;
6. `modules/files/admin/files.admin.php` – редактирование файлов;
7. `modules/files/admin/files.admin.menu.php` – формирование списка файлов и категорий файлов в административной части модуля «Меню».
8. `modules/files/admin/files.admin.category.php` – редактирование категорий файлов;
9. `modules/files/admin/files.admin.config.php` – конфигурация модуля.

## Теги

- [Подключение](#)
- [Как добавлять теги к модулю](#)
- [Административная часть](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль относится к части CMS «Контент» и может быть установлен опционально.

Модуль «Теги» подключается к другим модулям. В стандартной комплектации diafan.CMS «Теги» подключены к следующим модулям: Фотогалерея, Новости, Статьи, Вопрос-Ответ, Магазин, Файловый архив. Включение тегов в рамках модуля настраивается в конфигурации конкретного модуля.

## Подключение

Подключаемая часть – файл *modules/tags/tags.inc.php*. В нем описан класс *Tags\_inc*. В модуле к объекту классам можно обратиться через переменную `$this->diafan->_tags`. Экземпляр класса создается при первом вызове переменной.

## Методы

string **get**([integer \$element\_id = 0], [string \$module\_name = "]) – выводит подключенные к элементу теги.

- access: public
- integer \$element\_id: номер элемента модуля
- string \$module\_name: название модуля

## Пример

```
// получаем теги статьи в файле modules/clauses/clauses.model.php
$tags = $this->diafan->_tags->get($id, 'clauses');
// выводим теги в шаблоне модуля в файле modules/clauses/clauses.
view.php
echo $tags;
```

## Как добавлять теги к модулю

В конфигурации модуля (*modules/модуль/admin/модуль.admin.config.php*) и в файле редактирования элементов модуля (*modules/модуль/admin/модуль.admin.php*) необходимо добавить параметр **'tags'**:

```
$this->variable_table = array(..., 'tags' => 'module', ...);
```

Страница тега в пользовательской части выводит элементы список элементов других модулей, к которым тег подключен.

Данные о подключаемом элементе генерируются либо функцией *tags*, либо, если *tags* не определена, функцией *get\_elements* из модели подключаемого модуля.

### Пример

В классе *Shop\_model* файл *modules/shop/shop.model.php* определена функция *tags*. Она используется модулем «Теги» для формирования данных о товаре для списка тегов.

А вот для новостей функция *tags* не определена, поэтому данные о новости, к которой прикреплен тег, формируются с помощью функции *get\_elements* класса *News\_model* из файла *modules/news/news.model.php*.

Похожая ситуация с шаблонами вывода. Если в шаблоне модуля определена функция *list\_tags*, то используется она, иначе берется стандартная функция вывода списка элементов модуля – *list\_*.

### Пример

В классе *Photo\_view* (файл *modules/photo/photo.view.php*) определена функция *list\_tags*. Это шаблон вывода фотографий для списка элементов, прикрепленных к тегу.

В классе *Clauses\_view* (файл *modules/clauses/clauses.view.php*) функция *list\_tags* не определена, поэтому используется стандартная функция из этого класса *list\_*.

## Административная часть

В административной части модуля собираются все теги в одну таблицу, но при редактировании конкретного элемента модуля, к которому подключены теги, можно редактировать теги к нему. Теги можно сортировать по принадлежности к отдельному элементу.

Кнопка  служит для удаления тега. Порядок отображения тегов можно редактировать с помощью кнопок  .

При удалении какого-либо элемента, удаляются все теги, прикрепленные к нему.

Теги имеют следующие характеристики.

Основные:

- **Название тега;**
- **Псевдоссылка** – ссылка на страницу тега будет генерироваться следующим образом: *http://www.site.ru/псевдоссылка/*. Если псевдоссылка не указана адрес страницы будет: *http://www.site.ru/псевдоссылка\_страницы/showНомер\_тега/*. При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы. При редактировании можно это исправить.

Дополнительные:

- **Сортировка: установить перед** – редактирование порядка отображения пункта;
- **Не показывать на карте сайта** – скрывает отображение ссылки на страницу тега в файле *sitemap.xml*.

## CSS-форматирование

Теги, прикрепленные к элементу:

`.tags{}` – теги, прикрепленные к элементу

`.tags .tags_header{}` – заголовок «Теги»

Страница тега:

`.tags_list{}` – список элементов, к которым прикреплен тег (если не используется шаблон модуля)

`.tags_list .tag_name{}` – название элемента

`.tags_list .tag_text{}` – краткий анонс элемента

Шаблонная функция `show_block`:

`.tags_block{}` – список всех тегов

`.tags_block h2{}` – заголовок «Теги»

`.tags_block a{}` – ссылки на теги

`.tags_block span{}` – текущий/активный тег

## Файлы

1. `modules/tags/tags.inc.php` – формирует вывод тегов к элементам модулей;
2. `modules/tags/tags.php` – контроллер модуля;
3. `modules/tags/tags.model.php` – страница «Теги» выводит элементы модулей, соответствующих тегу;
4. `modules/tags/tags.view.php` – шаблон вывода;
5. `modules/tags/tags.sitemap.php` – карта сайта модуля для файла `sitemap.xml`;
6. `modules/tags/tags.install.php` – установка модуля;
7. `modules/модуль/модуль.model.php` – подключение тегов к модулям;
8. `modules/tags/admin/tags.admin.php` – редактирование тегов;
9. `modules/tags/admin/tags.admin.js` – редактирование тегов, JS-сценарий;
10. `modules/tags/admin/tags.admin.view.php` – шаблон вывода тегов в административной части;
11. `modules/tags/admin/tags.admin.edit.php` – вывод тегов в административной части модулей;
12. `modules/tags/admin/tags.admin.save.php` – сохранение тегов в административной части модулей;
13. `modules/tags/admin/tags.admin.act.php` – активация/блокирование тегов при активации/блокировании элементов модулей;
14. `modules/tags/admin/tags.admin.del.php` – удаление тегов при удалении элементов модулей;
15. `modules/tags/admin/tags.admin.ajax.php` – обработка Ajax-запросов в административной части модуля.

## Вопрос-Ответ

- [Категории](#)
- [Вопросы - ответы](#)
- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль позволяет пользователям задавать вопросы на сайте, а администратору публиковать ответы на эти вопросы с возможностью отправки ответа на e-mail пользователя, задавшего вопрос.

Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Модуль «Вопрос-Ответ» можно прикреплять к различным страницам сайта. Если модуль прикреплен к нескольким страницам, в административной части модуля появляется специальный фильтр «Раздел сайта» с возможностью фильтрации вывода (списка вопросов, категорий и настроек) по принадлежности к странице.

Модуль состоит из 2х частей: категории вопросов и вопросы-ответы.

## Категории

Категории вопросов могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в конфигурации модуля.

Если перед названием категории имеется знак +, то этой категории принадлежат подкатегории. Добавить подкатеорию к категории можно, нажав на знак  рядом с названием категории.

Порядок отображения категорий можно редактировать с помощью кнопок  .

Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот,  блокирования). При удалении категории удаляются все вложенные подкатегории и вопросы.

Категории имеют следующие характеристики.

Основные:

- **Название;**
- **Показывать на сайте;**
- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Анонс** – краткое описание категории, для публикации в списке;
- **Описание** – описание категории.

Дополнительные:

- **Номер** – номер категории;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет

автоматически сформирован из Названия категории - Названия категории - Названия сайта;

- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords», используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу категории будет генерироваться следующим образом: *http://www.site.ru/псевдоссылка/*. Если псевдоссылка не указана адрес страницы будет:  
*http://www.site.ru/псевдоссылка\_страницы/catНомер\_категории/*. При добавлении псевдоссылки к ней автоматически прикрепляется псевдоссылка страницы. При редактировании можно это исправить;
- **Вложенность: принадлежит** – категория верхнего уровня;
- **Раздел сайта** – принадлежность к странице сайта;
- **Сортировка: установить перед** – редактирование порядка отображения пункта. Поле доступно для редактирования только для элементов, отображаемых на сайте;
- **Не показывать на карте сайта** – скрывает отображение ссылки на категорию в файле sitemap.xml и [модуле «Карта сайта»](#).

Если в конфигурации модуля включен параметр «Показывать комментарии к категориям», то при редактировании категории будет отображена активная ссылка на комментарии к категории или надпись «Комментариев нет» при их отсутствии.

## Вопросы - ответы

В списке вопросы выводятся с приоритетом новизны.

Кнопка  служит для удаления вопроса. Кнопка  служит для быстрой публикации вопроса (или, наоборот,  блокирования).

Вопросы имеют следующие характеристики.

Основные:

- **Имя** – ФИО пользователя, задавшего вопрос;
- **Дата** – дата поступления вопроса;
- **Показывать на сайте** – публикация на сайте, активность;
- **Часто задаваемый вопрос** – используется в шаблонной функции [show\\_block](#);
- **Отправить ответ** – отправляет ответ пользователю. Содержание письма, а также e-mail, указываемый в обратном адресе можно редактировать в конфигурации модуля. Письмо не может быть отправлено, если не заполнено текстовое поле для ответа, поле с вопросом или e-mail получателя;
- **E-mail** – электронный ящик получателя ответа;
- **Прикрепленные файлы**;
- **Вопрос** – вопрос, который задал пользователь в форме в пользовательской части сайта;

- **Ответ** – текстовое поле для ответа.

Дополнительные:

- **Показывать в меню** – добавляет ссылку в меню согласно структуре сайта. Подробнее смотрите в разделе [модуль «Меню на сайте»](#);
- **Номер** – номер вопроса;
- **Время последнего изменения** – выводится в заголовке *Last modify*;
- **Заголовок окна в браузере, тэг Title** – если не заполнен, тег title будет автоматически сформирован из Названия страницы - Названия раздела - Названия сайта;
- **Ключевые слова, тэг Keywords** – ключевые слова для мета-тега «keywords», используется для поисковой оптимизации;
- **Описание, тэг Description** – описание для мета-тега «description», используется для поисковой оптимизации;
- **Псевдоссылка** – ссылка на страницу вопроса будет генерироваться следующим образом: *http://www.site.ru/псевдоссылка/*. Если псевдоссылка не указана адрес страницы будет:  
*http://www.site.ru/псевдоссылка\_страницы/catНомер\_категорию/showНомер\_в\_опроса/*. При добавлении псевдоссылки автоматически прикрепляется псевдоссылка страницы и категории вопроса. При редактировании можно это исправить;
- **Раздел сайта** – принадлежность к странице сайта;
- **Категория** – помимо основной можно указать дополнительные категории;
- **Не показывать на карте сайта** – скрывает отображение ссылки на вопрос в файле *sitemap.xml*.

Если в конфигурации модуля включен параметр «Показывать комментарии», то при редактировании вопроса в административной части сайта будет отображена активная ссылка на комментарии к вопросу или надпись «Комментариев нет» при их отсутствии.

При поступлении вопроса из формы в пользовательской части сайта может быть отправлено уведомительное письмо администратору, при соответствующей настройке в конфигурации модуля. Тема письма, содержание и e-mail настраиваются также в конфигурации модуля.

Если в конфигурации модуля включен параметр «Подключить теги», то при редактировании вопроса появится возможность редактирования тегов, прикрепленных к вопросу.

Если в конфигурации модуля включен параметр «Показывать рейтинг вопросов», то при редактировании вопроса будет отображена активная ссылка на рейтинг вопроса с указанием среднего балла и количества голосов или надпись «Рейтинг: нет голосов», если голосов за новость нет.

Если в конфигурации модуля разрешено прикрепление файлов («Разрешить добавление файлов»), то необходимо указать «Максимальное количество добавляемых файлов» (если поле пустое, то нет возможности прикрепить файлы), «Доступные типы файлов (через запятую)» (пример: doc, gif, jpg, mpg, pdf, png, txt, zip). Файлы загружаются в таблицу базы данных *{attachment}*.

Шаблонная функция: [show\\_block](#), [show\\_form](#).

## Настройки

Можно сохранять разные конфигурации модуля для разных страниц, к которым прикреплен модуль.

**Использовать категории** – позволяет включить/отключить категории вопросов.

**Количество вопросов на странице** – количество одновременно выводимых вопросов в списке вопросов.

**Количество вопросов в списке категорий** – количество вопросов, выводимых в списке категорий на главной странице модуля.

**Показывать вопросы подкатегорий** – позволяет показывать в категории верхнего уровня не только вопросы, принадлежащие текущей категории, но и вопросы подкатегорий, принадлежащих данной категории.

**Защита от спама** – возможные варианты:

- нет – не используется никаких средств для защиты от спама;
- только для зарегистрированных пользователей – вопросы могут добавлять только зарегистрированные пользователи;
- использовать защитный код – для добавления вопроса пользователь должен ввести защитный код.

**Формат даты** – позволяет настроить отображение даты в модуле.

**Сообщение после отправки** – сообщение, получаемое пользователем при удачной загрузке вопроса, допускаются HTML-теги для оформления сообщения.

Пример: `<div align="center"><b>Спасибо за ваше сообщение!</b><br>Наш консультант подберет необходимую информацию, после чего ваш вопрос и ответ на него будут опубликованы на этой странице.</div>`

**Ваше сообщение уже имеется в базе** – сообщение, получаемое пользователем при повторной попытке отправить вопрос.

**Разрешить добавление файлов** – позволяет пользователям прикреплять файлы к вопросам.

**Максимальное количество добавляемых файлов** – количество добавляемых файлов. Если значение равно нулю, то форма добавления файлов не выводится.

**Доступные типы файлов (через запятую)**

Пример: `doc, gif, jpg, mpg, pdf, png, txt, zip`

**Распознавать изображения** – позволяет прикрепленные к вопросу файлы в формате JPEG, GIF, PNG отображать как изображения.

**Размер для большого изображения** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер для маленького изображения** – размер изображения предпросмотра.

**Максимальный размер загружаемых файлов** – параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.

**Тема письма для ответа** – ответ отправляется пользователю, если при редактировании вопроса заполнены поля «E-mail», «Вопрос», «Ответ» и «Отправить ответ».

Пример: *%title (%url). Вопрос-Ответ.*

Где:

- *%title* – название сайта,
- *%url* – адрес сайта (например, site.ru).

### Сообщение для ответа

Пример: *Здравствуйте, %name!<br>Вы задали вопрос на сайте %title (%url).<br><b>Вопрос:</b> %question <br><b>Ответ:</b> %answer.*

Где:

- *%name* – имя пользователя,
- *%title* – название сайта,
- *%url* – адрес сайта (например, site.ru),
- *%question* – вопрос,
- *%answer* – ответ.

**E-mail, указываемый в обратном адресе пользователю** – возможные значения:

- e-mail, указанный в параметрах сайта;
- другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).

**Уведомлять о поступлении новых вопросов на e-mail** – возможность уведомления администратора о поступлении новых вопросов из формы в пользовательской части сайта.

**E-mail для уведомлений администратора** – возможные значения:

- e-mail, указанный в параметрах сайта;
- другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).

### Тема письма для уведомлений

Пример: *%title (%url). Новый вопрос в рубрике Вопрос-Ответ.* Где:

- *%title* – название сайта,
- *%url* – адрес сайта (например, site.ru).

### Сообщение для уведомлений

Пример: *Здравствуйте, администратор сайта %title (%url)!<br>В рубрике Вопрос-Ответ появился новый вопрос:<br>%question.<br>%name<br>%email.* Если включен параметр «Разрешить добавление файлов», то добавляется строка:  
*<br>Прикрепленные файлы: %files.*

Где:

- *%title* – название сайта,

- `%url` – адрес сайта (например, `site.ru`),
- `%question` – вопрос,
- `%name` – имя пользователя,
- `%email` – e-mail пользователя,
- `%files` – название прикрепляемых файлов.

**Показывать комментарии к вопросам** – подключение модуля «Комментарии». Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [«Модуль Комментарии»](#).

**Показывать комментарии к категориям** – подключение модуля «Комментарии» к категориям вопросов. Параметр не будет включен, если модуль «Комментарии» не установлен. Подробности см. в разделе [«Модуль Комментарии»](#).

**Подключить теги** – подключение модуля «Теги». Параметр не будет включен, если модуль «Теги» не установлен. Подробности см. в разделе [«Модуль Теги»](#).

**Показывать рейтинг вопросов** – подключение модуля «Рейтинг». Параметр не будет включен, если модуль «Рейтинг» не установлен. Подробности см. в разделе [«Модуль Рейтинг»](#).

## CSS-форматирование

Общие стили:

`.faq_date` – дата

`.faq_question` – вопрос

`.faq_answer` – ответ

Список вопросов:

`.faq_list` – список вопросов

`.faq_list .faq` – вопрос в списке

`.faq_list .faq_date` – дата вопроса в списке

`.faq_list .faq_question` – вопрос в списке

`.faq_list .faq_answer` – ответ в списке

`.faq_list .faq_cat_anons` – краткое описание категории вопросов

`.faq_list .faq_cat_text` – полное описание категории вопросов

`.faq_list .news_cat_link` – ссылка на вложенную категорию вопросов

`.faq_list .show_all` – ссылка на все вопросы в категории

`.faq_list .previous_next_links` – ссылки на предыдущую, следующую категории

`.faq_list .previous_link` – ссылка на предыдущую категорию

`.faq_list .next_link` – ссылка на следующую категорию

Страница вопроса:

`.faq_id` – страница вопроса

`.faq_id .faq_date` – дата на странице вопроса

`.faq_id .faq_question` – вопрос на странице вопроса  
`.faq_id .faq_answer` – ответ на странице вопроса  
`.faq_id .show_all` – ссылка на все вопросы на странице вопроса  
`.faq_id .previous_next_links` – ссылки на предыдущий, следующий вопросы  
`.faq_id .previous_link` – ссылка на предыдущий вопрос  
`.faq_id .next_link` – ссылка на следующий вопрос

Форма «Задать вопрос»:

`.faq_form` – форма «Задать вопрос»  
`.faq_form .inptext` – поля «Ваше имя», «Ваш вопрос», «E-mail для ответа»  
`.faq_form .inpfiler` – поле «Прикрепляемые файлы»  
`.faq_form .infofield` – подпись к полю в форме  
`.faq_form .button` – кнопка «Отправить» в форме

Шаблонная функция [show\\_block](#):

`.faq_block` – шаблонная функция `show_block`  
`.faq_block h2` – заголовок  
`.faq_block .faq` – вопрос в блоке  
`.faq_block .faq_date` – дата  
`.faq_block .faq_question` – вопрос  
`.faq_block .faq_answer` – ответ  
`.faq_block .show_all` – ссылка на все вопросы

## Файлы

1. `modules/faq/faq.php` – контроллер модуля;
2. `modules/faq/faq.model.php` – формирование данных для страницы модуля;
3. `modules/faq/faq.view.php` – шаблон модуля;
4. `modules/faq/faq.ajax.php` – обработка запроса на добавление вопроса;
5. `modules/faq/faq.install.php` – установка модуля;
6. `modules/faq/faq.sitemap.php` – карта модуля для файла `sitemap.xml`;
7. `modules/faq/admin/faq.admin.php` – редактирование вопросов;
8. `modules/faq/admin/faq.admin.category.php` – редактирование категорий вопросов;
9. `modules/faq/admin/faq.admin.config.php` – конфигурация модуля;
10. `modules/faq/admin/faq.admin.menu.php` – формирование списка вопросов и категорий вопросов в административной части модуля «Меню».

## Обратная связь

- [Сообщения](#)
- [Конструктор формы](#)
- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль позволяет пользователям оставлять сообщения на сайте и отвечать отправителям в административной части модуля. Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Модуль «Обратная связь» можно присоединить к различным страницам сайта.

Если модуль прикреплен к нескольким страницам, в административной части модуля выводится фильтр «Раздел сайта», позволяющий отобразить сообщения и настройку модуля для одной страницы.

## Сообщения

В списке сообщения сортируются в порядке добавления пользователями.

Кнопка  служит для удаления сообщения.

Модуль имеет следующие характеристики.

Основные:

- **Дата** – дата поступления вопроса;
- **Раздел сайта** – принадлежность к странице сайта;
- **Язык интерфейса** – языковая версия сайта, на которой находился пользователь, когда писал сообщение;
- **Прикрепленные файлы**;
- **Дополнительные поля** – поля, добавленные в конструкторе формы;
- **Отправить ответ** – возможность отправить ответ на e-mail, указанный пользователем. Содержание письма, а также e-mail, указываемый в обратном адресе можно редактировать в конфигурации модуля. Письмо не может быть отправлено, если не заполнено текстовое поле для ответа, поле с вопросом или e-mail получателя. Поле «Отправить ответ» появляется, если в конструкторе формы существует поле с типом «электронный ящик»;
- **Ответ** – текстовое поле для ответа.

При поступлении вопроса из формы в пользовательской части сайта может быть отправлено уведомительное письмо администратору. Тема письма, содержание и e-mail настраиваются в конфигурации модуля.

Если есть дополнительно поле с типом электронный ящик, то в административной части можно ответить на сообщение, отправив письмо на указанный ящик. Тема письма, содержание и обратный адрес настраиваются в конфигурации модуля.

Если в конфигурации модуля разрешено прикрепление файлов («Разрешить добавление файлов»), то необходимо указать «Максимальное количество

добавляемых файлов» (если поле пустое, то нет возможности прикрепить файлы), «Доступные типы файлов (через запятую)» (пример: doc, gif, jpg, mpg, pdf, png, txt, zip). Файлы загружаются в таблицу базы данных *{attachment}* и доступны для просмотра только администратору.

## Конструктор формы

На разных страницах сайта структура формы различная. Это обеспечивает конструктор формы.

Порядок отображения полей можно редактировать с помощью кнопок  . Кнопка  служит для удаления полей.

Конструктор – это дополнительные поля формы обратной связи. Можно добавлять поля общие (для всех разделов) или присваивать поле отдельной форме на отдельной странице.

Дополнительные поля имеют следующие параметры.

Основные:

- **Название;**
- **Тип** – строка, число, дата, текстовое поле, галочка, выпадающий список, список с выбором нескольких значений, электронный ящик;
- **Раздел сайта** – принадлежность к странице сайта;
- **Обязательно для заполнения;**
- **Описание;**

Дополнительные;

- **Сортировка: установить перед** – редактирование порядка отображения поля.

Если выбран тип «выпадающий список» или «список с выбором нескольких значений», то появятся дополнительные поля со значениями. Добавить значение можно, нажав  внизу последнего поля, сортировать значения можно с помощью кнопок  . Кнопка  служит для удаления значений.

Шаблонная функция: [show\\_form](#).

## Настройки

Можно сохранять разные конфигурации модуля для разных страниц, к которым прикреплен модуль.

**Защита от спама** – возможные варианты:

- нет – не используется никаких средств для защиты от спама;
- только для зарегистрированных пользователей – вопросы могут добавлять только зарегистрированные пользователи;
- использовать защитный код – для добавления вопроса пользователь должен ввести защитный код.

**Сообщение после отправки** – сообщение, получаемое пользователем при удачной загрузке вопроса, допускаются HTML-теги для оформления сообщения.

Пример: `<div align="center"><b>Спасибо за ваше сообщение!</b></div>`

**Сообщение об ошибке отправки** – сообщение, получаемое пользователем при неудачной загрузке вопроса в базу данных, допускаются HTML-теги для оформления сообщения.

Пример: *Невозможно добавить ваше сообщение<br>Проверьте правильность ввода.*

**Не заполнены обязательные поля** – сообщение при попытке отправить сообщение с незаполненными обязательными полями.

Пример: *Пожалуйста, заполните обязательное поле:*

**Разрешить добавление файлов** – позволяет пользователям прикреплять файлы к сообщению.

**Максимальное количество добавляемых файлов** – количество добавляемых файлов. Если значение равно нулю, то форма добавления файлов не выводится.

**Доступные типы файлов (через запятую)**

Пример: *doc, gif, jpg, mpg, pdf, png, txt, zip*

**Распознавать изображения** – позволяет прикрепленные к сообщению файлы в формате JPEG, GIF, PNG отображать как изображения.

**Размер для большого изображения** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер для маленького изображения** – размер изображения предпросмотра.

**Максимальный размер загружаемых файлов** – параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.

**Тема письма для ответа** – ответ отправляется пользователю, если при редактировании вопроса заполнены поля «E-mail», «Сообщение», «Ответ» и «Отправить ответ».

Пример: *%title (%url). Обратная связь.*

Где:

- *%title* – название сайта,
- *%url* – адрес сайта (например, site.ru).

**Сообщение для ответа**

Пример: *Здравствуйте!<br>Вы оставили сообщение в форме обратной связи на сайте %title (%url).<br><b>Сообщение:</b> %message <br><b>Ответ:</b> %answer.*

Где:

- *%title* – название сайта,
- *%url* – адрес сайта (например, site.ru),
- *%message* – сообщение,

– *%answer* – ответ.

**Е-mail, указываемый в обратном адресе пользователю** – возможные значения:

- е-mail, указанный в параметрах сайта;
- другой (при выборе этого значения появляется дополнительное поле **впишите е-mail**).

**Уведомлять о поступлении новых сообщений на е-mail** – возможность уведомления администратора о поступлении новых сообщений из формы в пользовательской части сайта.

**Е-mail для уведомлений администратора** – возможные значения:

- е-mail, указанный в параметрах сайта;
- другой (при выборе этого значения появляется дополнительное поле **впишите е-mail**).

**Тема письма для уведомлений**

Пример: *%title (%url)*. Новое сообщение в рубрике Обратная связь.

Где:

- *%title* – название сайта,
- *%url* – адрес сайта (например, site.ru).

**Сообщение для уведомлений**

Пример: *Здравствуйте, администратор сайта %title (%url)!*  
*В рубрике Обратная связь появилось новое сообщение:*  
*%message*. Если включен параметр «Разрешить добавление файлов», то добавляется строка:  
*Прикрепленные файлы: %files*.

Где:

- *%title* – название сайта,
- *%url* – адрес сайта (например, site.ru),
- *%message* – вопрос,
- *%files* – название прикрепляемых файлов.

## CSS-форматирование

*.feedback\_form*{} – форма

*.feedback\_form .infofiled*{} – подпись к полю в форме

*.feedback\_form .inpnext*{} – текстовые поля

*.feedback\_form .inpnum*{} – цифровые поля

*.feedback\_form .inpselect*{} – поля «Список»

*.feedback\_form .inpcheckbox*{} – поля «Чекбокс»

*.feedback\_form .inpfile*{} – поле «Файл»

*.feedback\_form .button*{} – кнопка «Оправить»

`.feedback_form_param_text{}` – описание поля

`.feedback_form_paramNUM{}` – поле с номером NUM

`.feedback_form_paramNUM .infofield{}` – название параметра с номером NUM

`.feedback_form_paramNUM .inpertext{}` – текстовое поле для ввода текстового значения параметра с номером NUM

`.feedback_form_paramNUM .inpnum{}` – текстовое поле для ввода цифрового значения параметра с номером NUM

`.feedback_form_paramNUM .inpselect{}` – список для выбора значения параметра с номером NUM

`.feedback_form_paramNUM .inpcheckbox{}` – чекбокс для выбора значения параметра с номером NUM

## Файлы

1. `modules/feedback/feedback.php` – контроллер модуля;
2. `modules/feedback/feedback.model.php` – форма отправки сообщений в пользовательской части сайта;
3. `modules/feedback/feedback.view.php` – шаблон формы в пользовательской части сайта;
4. `modules/feedback/feedback.ajax.php` – обработка запроса на добавление сообщения;
5. `modules/feedback/feedback.install.php` – установка модуля;
6. `modules/feedback/admin/feedback.admin.php` – редактирование сообщений;
7. `modules/feedback/admin/feedback.admin.param.php` – редактирование списка дополнительных полей формы;
8. `modules/feedback/admin/feedback.admin.param.js` – редактирование списка дополнительных полей формы, JS-сценарий;
9. `modules/feedback/admin/feedback.admin.ajax.php` – обработка Ajax-запросов в административной части модуля;
10. `modules/feedback/admin/feedback.admin.config.php` – конфигурация модуля.

## Комментарии

- [Подключение](#)
- [Как добавлять комментарии к модулю](#)
- [Административная часть](#)
- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Модуль «Комментарии» подключается к другим модулям. В стандартной комплектации diafan.CMS «Комментарии» подключены к следующим модулям: Фотогалерея, Новости, Статьи, Вопрос-Ответ, Магазин, Файлы, Страницы сайта. Подключение комментариев в рамках модуля настраивается в конфигурации конкретного модуля.

В пользовательской части сайта комментарии добавляются с помощью технологии Ajax, то есть без перезагрузки всей страницы. Комментарии, содержащие слова длиннее 20 символов, не пропускаются.

## Подключение

Подключаемая часть – файл *modules/comments/comments.inc.php*. В нем описан класс *Comments\_inc*. В модуле к объекту класса можно обратиться через переменную *\$this->diafan->\_comments*. Экземпляр класса создается при первом вызове переменной.

## Методы

string **get()** – показывает комментарии, прикрепленные к элементу, и форму добавления комментария.

- access: public

## Пример

```
// получаем комментарии к новости и форму добавления комментария в
// файле modules/news/news.model.php
$comments = $this->diafan->_comments->get();
// выводим комментарии в шаблоне модуля в файле modules/news/news.
// view.php
echo $comments;
```

## Как добавлять комментарии к модулю

В конфигурации модуля (*modules/модуль/admin/модуль.admin.config.php*) и в файле редактирования элементов модуля (*modules/модуль/admin/модуль.admin.php*) необходимо добавить параметр **'comments'**:

```
$this->variable_table = array(..., 'comments' => 'module', ...);
```

## Административная часть

В административной части модуля собираются все комментарии в одну таблицу и отображаются в порядке поступления. Комментарии можно сортировать по принадлежности к отдельному элементу. Для этого предусмотрены ссылки «Комментарии» в административной части модулей, к которым подключаются комментарии.

К примеру, при редактировании фотографии появится ссылка «Комментарии» на комментарии к этой фотографии или надпись «Комментариев нет» при их отсутствии.

Кнопка  служит для удаления комментария. Кнопка  служит для быстрой публикации комментария (или, наоборот,  блокирования).

При удалении какого-либо элемента, удаляются все комментарии, прикрепленные к нему.

Комментарии имеют следующие характеристики.

Основные:

- **Комментарий**;
- **Дата добавления**;
- **Показывать на сайте** – публикация на сайте, активность;
- **Объект** – объект, к которому прикреплены комментарии, ссылка на все комментарии к этой странице;
- **Пользователь** – пользователь, добавивший комментарий (если комментарий добавлен зарегистрированным пользователем).

## Настройки

**Только для зарегистрированных пользователей** – параметр позволяет запретить незарегистрированным пользователям добавлять комментарии.

**Использовать защитный код** – для добавления комментария пользователь должен ввести защитный код, комментировать могут все.

**Модерация сообщений** – комментарий не появится на сайте пока администратор не сделает его активным.

**Ваше сообщение уже имеется в базе** – сообщение пользователю при попытке повторного добавления сообщения.

**Спасибо! Ваш комментарий будет проверен в ближайшее время и появится на сайте.** – сообщение пользователю при удачном добавлении комментария

**Ваш комментарий не может быть добавлен** – сообщение, получаемое пользователем при неудачной загрузки комментария в базу данных.

**Отображать имя пользователя, добавившего комментарий.**

**Количество комментариев на странице.**

**Формат даты** – позволяет настроить отображение даты в модуле.

## CSS-форматирование

Форма добавления комментария:

`.comments_form` – форма добавления комментариев  
`.comments_form h2` – заголовок «Оставьте комментарий»  
`.comments_form .button` – кнопка «Отправить»  
`.comments_form .inptext` – текстовое поле для комментария

Список комментариев:

`.comments` – список комментариев  
`.comments_date` – дата добавления комментариев  
`.comments_name` – имя автора комментария  
`.comments_name .avatar` – аватар автора комментария  
`.comments_text` – текст комментария

## Файлы

1. `modules/comments/comments.inc.php` – формирование вывода комментариев;
2. `modules/comments/comments.view.php` – шаблон вывода;
3. `modules/comments/comments.ajax.php` – обработка запроса на добавление комментария;
4. `modules/comments/comments.install.php` – установка модуля;
5. `modules/comments/comments.js` – JS-сценарий постраничной навигации комментариев;
6. `modules/модуль/модуль.model.php` – подключение комментариев к модулям;
7. `modules/comments/admin/comments.admin.php` – редактирование комментариев;
8. `modules/comments/admin/comments.admin.edit.php` – вывод комментариев в административной части модулей;
9. `modules/comments/admin/comments.admin.save.php` – сохранение комментариев в административной части модулей;
10. `modules/comments/admin/comments.admin.del.php` – удаление комментариев при удалении элементов модулей;
11. `modules/comments/admin/comments.admin.config.php` – конфигурация модуля.

## Форум

- [Категории](#)
- [Сообщения](#)
- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Для модуля «Форум» предусмотрен особый тип пользователей «Модераторы». Это обычные пользователи сайта, помеченные как «Модератор» в административной части сайта.

Основная часть для работы с форумом - пользовательская. Пользователи добавляют темы и сообщения (в соответствии с конфигурацией модуля). Модераторы могут удалять, блокировать и редактировать темы и сообщения.

Административная часть модуля состоит из 2х частей: категории и сообщения.

## Категории

Категории формируют структуру форума:

- первый уровень – логические блоки,
- второй уровень – разделы;
- третий уровень – темы.

Прежде чем запустить форум, администратор должен создать структуру форума.

Если пользователь нажимает на название категории, то он переходит к списку сообщений, принадлежащих данной категории. В открывшейся странице рядом с названием категории появится ссылка «изменить», которая ведет к странице редактирования категории.

Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот,  блокирования). При удалении категории удаляются все вложенные подкатегории и сообщения. Категории выводятся в порядке создания.

Категории имеют следующие характеристики.

- **Название;**
- **Показывать на сайте;**
- **Удалено модератором;**
- **Закрепить тему (всегда сверху);**
- **Закреть тему;**
- **Дата добавления;**
- **Вложенность: принадлежит** – категория верхнего уровня.

## Сообщения

Кнопка ✖️ служит для удаления сообщений. Кнопка 💡 служит для быстрой публикации сообщений (или, наоборот, 🚫 блокирования).

Сообщения имеют следующие характеристики.

- Показывать на сайте;
- Удалено модератором;
- Автор;
- Дата;
- Вложенность: принадлежит;
- Категория;
- Сообщение;
- Прикрепленные файлы.

## Настройки

**Добавление новых тем, сообщений** – ограничение добавления новых тем:

- все без ограничения;
- использовать защитный код;
- только зарегистрированные.

**Формат даты** – позволяет настроить отображение даты в модуле.

**Количество тем на странице** – количество одновременно выводимых тем в списке.

**Предмодерация темы для обсуждения** – добавленные темы отображаются на сайте только после одобрения модератором.

**Предмодерация сообщений** – добавленные сообщения отображаются на сайте только после одобрения модератором.

**Разрешить добавление файлов** – позволяет пользователям прикреплять файлы к сообщению.

**Максимальное количество добавляемых файлов** – количество добавляемых файлов. Если значение равно нулю, то форма добавления файлов не выводится.

**Доступные типы файлов (через запятую)**

Пример: *doc, gif, jpg, mpg, pdf, png, txt, zip*

**Распознавать изображения** – позволяет прикрепленные к сообщению файлы в формате JPEG, GIF, PNG отображать как изображения.

**Размер для большого изображения** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер для маленького изображения** – размер изображения предпросмотра.

**Использовать анимацию при увеличении изображений** – добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений.

**Максимальный размер загружаемых файлов** – параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.

## CSS-форматирование

`.forum_date` – дата

`.forum_author` – автор

`.forum_author .avatar` – аватар автора

`.forum_news` – обновленные темы, сообщения

`.forum_actions` – действия над темами, сообщениями

Список категорий, тем:

`.forum_add` – ссылка «Добавить тему»

`.forum_list` – список тем

`.forum_list th` – легенда списка категорий

`.forum_list .forum_title` – название заголовка категорий (категории 1го уровня)

`.forum_list .forum_category_name` – название категорий (категории 2го уровня)

`.forum_list .forum_news` – обновленная тема

`.forum_list .forum_count` – количество тем в категории

`.forum_new_messages` – ссылка на список новых сообщений

`.forum_list .forum_last_theme` – последняя тема в категории

`.forum_list .forum_theme_name` – название темы (категории 3го уровня)

`.forum_list .forum_blocked` – заблокированная тема

`.forum_list .forum_count_message` – количество сообщений в теме

`.forum_list .forum_author_date` – дата обновления и автор темы

`.forum_list .forum_last_user` – последнее сообщение в теме

`.forum_list .forum_author` – автор темы, сообщения в списке тем

`.forum_list .forum_date` – дата в списке тем

`.forum_list .forum_actions` – действия над темами (удалить, редактировать, заблокировать/разблокировать, открыть/закрыть, закрепить/открепить)

Поиск по форуму:

`.forum_search` – поиск по форуму

`.forum_search .inptext` – поле для ввода информации для поиска

`.forum_search .button` – кнопка Искать

`.forum_search_result` – общая статистика результатов поиска

`.forum_category` – найденная тема

`.forum_category .forum_author` – автор найденной темы

`.forum_category .forum_date` – дата создания найденной темы

`.forum_message` – найденное сообщение

`.forum_message .forum_author` – автор найденного сообщения

`.forum_message .forum_date` – дата создания найденного сообщения

Форма добавления/редактирования темы:

`.forum_form` – форма добавления темы/сообщения

`.forum_form .inptext` – поля формы

`.forum_form .infofield` – название поля формы

`.forum_form .button` – кнопка Сохранить

Список сообщений:

`.forum_message` – сообщение в списке

`.forum_message .forum_blocked` – заблокированное сообщение

`.forum_message .forum_actions` – действия над сообщениями (удалить, добавить, заблокировать, ссылка)

`.forum_message .forum_text` – текст сообщения

`.forum_message .quote` – цитата в сообщении

`.forum_message .code` – программный код в сообщении

`.forum_message .quote_header` – подпись «Цитата»

`.forum_message .code_header` – подпись «Код» в сообщении

`.forum_message .forum_author` – автор

`.forum_message .forum_date` – дата

`.forum_message_level` – вложенный уровень сообщений

Форма добавления/редактирования сообщения:

`.forum_dashed_link` – ссылки «Ответить», «Добавить сообщение»

`.forum_form` – форма добавления темы/сообщения

`.forum_form .infofield` – название поля формы

`.forum_form .inptext` – поля формы

`.forum_form .inpfile` – поле «Файл»

`.forum_form .attachment_extensions` – информация о расширениях файлов, доступных для добавления

`.forum_form .button` – кнопка Сохранить

`.forum_message_toolbar` – слой инструментов для редактирования сообщения

## Файлы

1. `modules/forum/forum.php` – контроллер модуля;
2. `modules/forum/forum.model.php` – формирование данных для страницы модуля;
3. `modules/forum/forum.view.php` – шаблон модуля;
4. `modules/forum/forum.ajax.php` – обработка запроса на добавление и редактирование темы;
5. `modules/forum/forum.sitemap.php` – карта сайта модуля для файла `sitemap.xml`;
6. `modules/forum/forum.install.php` – установка модуля;
7. `modules/forum/forum.js` – JS-сценарий модуля;
8. `modules/forum/message/message.inc.php` – работа с сообщениями;

9. *modules/forum/message/message.view.php* – шаблона сообщений;
10. *modules/forum/message/message.ajax.php* – обработка запроса на добавление и редактирование сообщений;
11. *modules/forum/admin/forum.admin.php* – администрирование модуля;
12. *modules/forum/admin/forum.admin.element.php* – редактирование сообщений;
13. *modules/forum/admin/forum.admin.category.php* – редактирование структуры форума;
14. *modules/forum/admin/forum.admin.config.php* – конфигурация модуля.

## Опросы

- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль относится к части CMS «Интерактив» и может быть установлен по опционально.

Модуль «Опросы» выводятся с помощью шаблонной функции [show\\_votes](#).

Голосование проходит с помощью технологии Ajax, то есть без перезагрузки страницы.

Административная часть модуля состоит из 2х частей: вопросы и ответы. При открытии модуля выводится список вопросов. При клике на вопрос откроется список ответов на вопрос.

Для редактирования вопроса необходимо нажать на ссылку «изменить» рядом с названием вопроса.

Кнопка  служит для удаления вопросов и ответов. При удалении вопроса удаляются и ответы на него. Кнопка  служит для быстрой публикации вопроса (или, наоборот,  блокирования). Порядок отображения вопросов и ответов можно редактировать с помощью кнопок  .

Вопросы имеют следующие характеристики.

Основные:

- **Вопрос;**
- **Показывать на сайте** – публикация на сайте, активность;
- **Сортировка: установить перед** – редактирование порядка отображения пункта. Поле доступно для редактирования только для незаблокированных опросов;
- **Расположение** – сквозное (на всех страницах) или на конкретной странице.

Ответы имеют следующие характеристики.

Основные:

- **Ответ;**
- **Количество ответов;**
- **Сортировка: установить перед** – редактирование порядка отображения пункта;
- **Вопрос.**

## Настройки

**Только для зарегистрированных пользователей** – параметр позволяет запретить неавторизованным пользователям голосовать.

**Защита от накруток** – возможные значения:

- нет;

- использовать защитный код (captcha);
- вести лог голосовавших (все голоса записываются в таблицу `{log_note}`, с параметром `module=votes`, пользователь не может голосовать повторно в рамках одной сессии);
- запрещать голосовать повторно (пользователь не может голосовать повторно в рамках одной сессии).

**Порядок отображения** – возможные варианты:

- случайны выбор (выводить один опрос);
- выводить все (все опросы в порядке сортировки).

## CSS-форматирование

`.votes_form{}` – форма голосования

`.votes_question{}` – вопрос

`.votes_form_answer{}` – ответ в форме

`.votes_answer{}` – ответ – результат голосования

`.votes_line{}` – линия визуализации количества проголосовавших

`.votes_count{}` – количество проголосовавших

`.votes_form .button{}` – кнопка «Ответить»

## Файлы

1. `modules/votes/votes.inc.php` – формирует вывод формы для голосования;
2. `modules/votes/votes.view.php` – шаблон вывода;
3. `modules/votes/votes.ajax.php` – обработка запроса на добавление оценки и отображение результатов голосования;
4. `modules/votes/votes.install.php` – установка модуля;
5. `modules/votes/votes.js` – JS-сценарий работы с формой голосования;
6. `modules/votes/admin/votes.admin.php` – редактирование опросов;
7. `modules/votes/admin/votes.admin.element.php` – редактирование вопросов;
8. `modules/votes/admin/votes.admin.category.php` – редактирование ответов;
9. `modules/votes/admin/votes.admin.config.php` – конфигурация модуля;

# Рейтинг

- [Подключение](#)
- [Как добавлять рейтинг к модулю](#)
- [Административная часть](#)
- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль относится к части CMS «Интерактив» и может быть установлен опционально.

Модуль «Рейтинг» подключается к другим модулям. В стандартной комплектации diafan.CMS «Рейтинг» подключен к следующим модулям: Фотогалерея, Новости, Статьи, Магазин, Файловый архив, Вопрос-Ответ. Включение рейтинга в рамках модуля настраивается в конфигурации конкретного модуля.

В пользовательской части сайта голосование проходит с помощью технологии Ajax, то есть без перезагрузки всей страницы.

## Подключение

Подключаемая часть – файл `modules/rating/rating.inc.php`. В нем описан класс `Rating_inc`. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_rating`. Экземпляр класса создается при первом вызове переменной.

## Методы

`string get([integer $element_id = 0], [string $module_name = "])` – показывает рейтинг для элемента.

- access: public
- integer \$element\_id: номер элемента модуля
- string \$module\_name: название модуля

`string add([integer $element_id = 0], [string $module_name = "])` – показывает форму оценки элемента.

- access: public
- integer \$element\_id: номер элемента модуля
- string \$module\_name: название модуля

## Пример

```
// получаем рейтинг товара в файле modules/shop/shop.model.php
$rating = $this->diafan->_rating->get($id, 'shop');

// выводим рейтинг в шаблоне модуля в файле
modules/shop/shop.view.php
echo $rating;

// получаем форму оценки товара
в файле modules/shop/shop.model.php
$add_rating = $this->diafan->_rating->add($id, 'shop');
```

```
// выводим рейтинг в шаблоне модуля в файле
modules/shop/shop.view.php
echo $add_rating;
```

## Как добавлять рейтинг к модулю

В конфигурации модуля (*modules/модуль/модуль.admin.config.php*) и в файле редактирования элементов модуля (*modules/модуль/модуль.admin.php*) необходимо добавить параметр `'rating'`:

```
$this->variable_table = array(..., 'rating' => 'module', ...);
```

## Административная часть

В административной части модуля собираются средние оценки за все элементы разных модулей в одну таблицу и отображаются в порядке обновления. Рейтинг отдельного элемента модуля можно посмотреть не посещая раздел «Рейтинг» в административной части. Так как средняя оценка и число голосовавших за элемент отображается при редактировании элемента модуля, к которому подключен рейтинг.

Кнопка  служит для удаления оценки. При удалении какого-либо элемента, удаляется оценка этого элемента.

Рейтинг имеет следующие характеристики.

Основные:

- **Объект;**
- **Средняя оценка;**
- **Количество голосовавших;**
- **Дата последнего голосования.**

В списке оценок выводятся название и модуль оцениваемого элемента.

## Настройки

**Только для зарегистрированных пользователей** – параметр позволяет запретить неавторизованным пользователям голосовать.

**Защита от накруток** – возможные значения:

- нет;
- использовать защитный код (captcha);
- вести лог голосовавших (все голоса записываются в таблицу `{log_note}`, с параметром `module=rating`, пользователь не может голосовать повторно в рамках одной сессии);
- запрещать голосовать повторно (пользователь не может голосовать повторно в рамках одной сессии).

## CSS-форматирование

`.rating_form` – форма оценки элемента

`.rating_form .button` – кнопка «Оценить»

`.rating_votes` – рейтинг элемента

## Файлы

1. `modules/rating/rating.inc.php` – формирование вывода рейтинга и формы для голосования;
2. `modules/rating/rating.view.php` – шаблон вывода;
3. `modules/rating/rating.ajax.php` – обработка запроса на добавление оценки;
4. `modules/rating/rating.install.php` – установка модуля;
5. `modules/модуль/модуль.model.php` – подключение рейтинга к модулям;
6. `modules/rating/admin/rating.admin.php` – редактирование оценок;
7. `modules/rating/admin/rating.admin.edit.php` – вывод рейтинга в административной части модулей;
8. `modules/rating/admin/rating.admin.save.php` – сохранение рейтинга в административной части модулей;
9. `modules/rating/admin/rating.admin.del.php` – удаление рейтинга при удалении элементов модулей;
10. `modules/rating/admin/rating.admin.config.php` – конфигурация модуля.

## Рассылки

- [Категории](#)
- [Выпуски рассылки](#)
- [Настройки](#)
- [Файлы](#)

Модуль относится к части CMS «Интерактив» и может быть установлен опционально. Модуль имеет только административную часть, откуда отправляются выпуски рассылок.

Модуль состоит из 2х частей: категории рассылки и выпуски рассылки.

## Категории

Категории рассылки могут иметь неограниченное число подкатегорий. При необходимости категории можно не использовать. Для этого нужно отключить параметр «Использовать категории» в конфигурации модуля.

Выпуск рассылки можно отнести к нескольким категориям, чтобы охватить пользователей, подписанных на разные категории.

Добавить подкатеорию к категории можно, нажав на знак  рядом с названием категории. Порядок отображения категорий можно редактировать с помощью кнопок  . Кнопка  служит для удаления категории. Кнопка  служит для быстрой публикации категории (или, наоборот,  блокирования). При удалении категории удаляются все вложенные подкатегории и выпуски рассылок.

Категории имеют следующие характеристики.

Основные:

- **Название;**
- **Показывать на сайте;**
- **Описание.**

Дополнительные:

- **Вложенность: принадлежит** – категория верхнего уровня;
- **Сортировка: установить перед** – редактирование порядка отображения пункта. Поле доступно для редактирования только для незаблокированных категорий.

## Выпуски рассылки

Выпуски рассылки выводятся в порядке отправки. Кнопка  служит для удаления выпуска.

Выпуски имеют следующие характеристики.

Основные:

- **Название;**
- **Дата добавления/Дата отправления** – отображается для уже добавленной или отправленной рассылки;

- **Отправить/Отправлено** – рассылка отправляется один раз, если отмечено поле «Отправить»;
- **Не учитывать язык пользователя** – если у сайта несколько языковых версий, рассылка отправляется пользователям языка, которых соответствует текущему языку для редактирования;
- **Описание** – сообщение, отправляемое пользователям.

Дополнительные:

- **Категория.**

Рассылка отправляется только в том случае, если заполнены поля «Описание», «Отправить» и рассылка не была отправлена ранее.

Рассылка отправляется пользователям сайта. Если используются категории, то при редактировании пользователей появляется поле «Подписан на следующие рассылки», где можно выбрать категории рассылки. По умолчанию все пользователи сайта подписаны на все рассылки.

## Настройки

**Использовать категории** – параметр позволяет включить/отключить категории рассылки.

### Тема письма для рассылки

Пример: *Рассылка сайта %title (%url). %subject.*

Где:

- %title – название сайта;
- %url – адрес сайта (например, site.ru);
- %subject – тема рассылки.

**E-mail, указываемый в обратном адресе** – возможные значения:

- e-mail, указанный в параметрах сайта;
- другой (при выборе этого значения появляется дополнительное поле **впишите e-mail**).

## Файлы

1. *modules/subscribtion/subscribtion.install.php* – установка модуля;
2. *modules/subscribtion/admin/subscribtion.admin.php* – отправление рассылки, редактирование списка рассылок;
3. *modules/subscribtion/admin/subscribtion.admin.category.php* – редактирование разделов рассылки;
4. *modules/usersite/admin/usersite.admin.php* – редактирование списка подписчиков;
5. *modules/subscribtion/admin/subscribtion.admin.config.php* – конфигурация модуля.

## Пользователи сайта / Регистрация

- [Регистрация](#)
- [Восстановление пароля](#)
- [Пользователи сайта](#)
- [Конструктор формы регистрации](#)
- [Настройки](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль «Регистрация» позволяет регистрироваться на сайте пользователям для получения дополнительных возможностей.

Модуль относится к части CMS «Интерактив» и может быть установлен по необходимости.

### Регистрация

В пользовательской части сайта модуль представлен 2 частями: форма регистрации и редактирования данных уже зарегистрированными пользователями. Регистрация и редактирование осуществляются с помощью технологии Ajax, то есть без перезагрузки всей страницы.

Шаблонная функция: [show\\_login](#).

### Восстановление пароля

**Восстановление пароля** – это отдельный модуль в пользовательской части сайта (*reminding*). Он устанавливается вместе с модулем «Регистрация» и необходим для восстановления пароля пользователями. Восстановление пароля осуществляется с помощью технологии Ajax, то есть без перезагрузки всей страницы.

### Пользователи сайта

В административной части модуля можно просмотреть и редактировать данные всех зарегистрированных пользователей.

Кнопка ✖ служит для удаления пользователя. Кнопка 💡 служит для быстрой активации пользователя (или, наоборот, 🔒 блокирования).

Пользователи имеют следующие характеристики:

- **ФИО**;
- **Логин**;
- **Пароль**;
- **Email**;
- **Дата регистрации**;
- **Активен** – возможность блокирования пользователя;
- **Модератор форума** – дает право пользователю модерировать сообщения в

пользовательской части [модуля «Форум»](#);

- **Язык** – фиксируется язык интерфейса, который был при регистрации пользователя. При авторизации устанавливается язык интерфейса, указанный в настройках пользователя;
- **Аватар**;
- **Дополнительные поля** – группа полей, созданных в конструкторе формы регистрации;
- **Подписан на следующие рассылки** – выводит список категорий рассылок; по умолчанию пользователь подписан на все рассылки. Поле выводится, если подключен [модуль «Рассылки»](#) и конфигурации модуля подключено использование категорий.

## Конструктор формы регистрации

Форму регистрации / редактирования данных можно дополнить своими полями с помощью конструктора формы.

Порядок отображения полей можно редактировать с помощью кнопок  . Кнопка  служит для удаления полей.

Конструктор – это дополнительные поля формы регистрации. Можно добавлять поля только для формы регистрации или только для формы редактирования данных или вообще не выводить поля для пользователя.

Дополнительные поля имеют следующие параметры.

Основные:

- **Название**;
- **Обязательно для заполнения**;
- **Использовать в форме** – возможные значения:
  - нет;
  - во всех формах;
  - только при регистрации;
  - только при редактировании данных.
- **Тип** – строка, число, дата, текстовое поле, галочка, выпадающий список, список с выбором нескольких значений, электронный ящик.

Если выбран тип «выпадающий список» или «список с выбором нескольких значений», то появятся дополнительные поля со значениями. Добавить значение можно, нажав  внизу последнего поля, сортировать значения можно с помощью кнопок  . Кнопка  служит для удаления значений.

## Настройки

**Использовать защитный код** – позволяет использовать защитный код (captcha) при регистрации.

**Сообщение пользователю по окончании регистрации**

Пример:

*Вы успешно зарегистрированы. Для активации аккаунта перейдите по ссылке, высланной на ваш e-mail.*

**Использовать аватар** – подключает аватар к пользователям сайта. Аватар загружается при регистрации, редактировании данных пользователем, редактировании данных о пользователе администратором. Используется аватар на форуме, в комментариях и выводится вместе с приветствием пользователю.

**Размер** – аватар будет генерироваться строго заданного размера.

**Уведомлять администратора** – возможность уведомления администратора о регистрации новых пользователей.

### **Тема письма администратору**

Пример: *%title (%url). Новый пользователь*

Где:

- *%title* – название сайта;
- *%url* – адрес сайта (например, site.ru).

### **Текст письма администратору**

Пример:

*Здравствуйте, администратор сайта %title (%url)!<br>На сайте появился новый пользователь: %fio (%login), %email*

Где:

- *%title* – название сайта;
- *%url* – адрес сайта (например, site.ru);
- *%fio* – ФИО пользователя;
- *%login* – логин пользователя;
- *%email* – e-mail пользователя.

**E-mail для уведомлений администратора** – возможные значения:

- e-mail, указанные в параметрах сайта;
- другой (при выборе этого значения появляется дополнительное поле куда необходимо вписать **e-mail**).

**Порядок активации пользователя** – возможные варианты:

- при регистрации;
- по ссылке, высланной на e-mail;
- администратором.

### **Тема письма новому пользователю**

Пример: *Вы зарегистрированы на сайте %title (%url)*

Где:

- *%title* – название сайта;
- *%url* – адрес сайта (например, site.ru).

### Сообщение новому пользователю

Пример:

если активация происходит при регистрации:

*Здравствуйме, %fio!<br>*  
*Вы зарегистрированы на сайте %title (%url).<br>*  
*Логин: %login<br>*  
*Пароль: %password*

если активация происходит по ссылке:

*Здравствуйме, %fio!<br>*  
*Вы зарегистрированы на сайте %title (%url).<br>*  
*Логин: %login<br>*  
*Пароль: %password<br>*  
*Для активации аккаунта пройдите по ссылке %actlink<br>*  
*Ссылка действует в течении суток.*

если активацию производит администратор:

*Здравствуйме, %fio!<br>*  
*Вы зарегистрированы на сайте %title (%url).<br>*  
*Логин: %login<br>*  
*Пароль: %password<br>*  
*Ваш аккаунт будет активирован после проверки администратором.*

Где:

- *%fio* – ФИО пользователя;
- *%title* – название сайта;
- *%url* – адрес сайта (например, site.ru);
- *%login* – логин пользователя;
- *%password* – пароль пользователя;
- *%actlink* – ссылка для активации аккаунта.

**Е-mail, указываемый в обратном адресе новому пользователю** – возможные значения:

- е-mail, указанный в параметрах сайта;
- другой (при выборе этого значения появляется дополнительное поле куда необходимо вписать **е-mail**).

### Сообщение пользователю при восстановлении пароля

Пример: *Новый пароль выслан Вам на e-mail.*

### Тема письма с новым паролем

Пример: *Вы запросили новый пароль на сайте %title (%url)*

Где:

- %title – название сайта;
- %url – адрес сайта (например, site.ru).

### Текст письма с новым паролем

Пример:

*Здравствуйте, %fio!<br>*

*Вы запросили новый пароль на сайте %title (%url).<br>*

*Логин: %login<br>*

*Пароль: %password*

Где:

- %fio – ФИО пользователя;
- %title – название сайта;
- %url – адрес сайта (например, site.ru);
- %login – логин пользователя;
- %password – пароль пользователя.

## CSS-форматирование

### Регистрация

Форма регистрации:

`.registration_form` – форма регистрации

`.registration_form .inptext` – текстовые поля для ввода данных

`.registration_form .infofield` – подпись к полю в форме

`.registration_form .inpnum` – цифровые поля

`.registration_form .inpselect` – поля «Список»

`.registration_form .inpcheckbox` – поля «Чекбокс»

`.registration_form .inpfile` – поле «Файл»

`.registration_form .button` – кнопка «Регистрация»

`.registration_form_param_text` – описание поля

`.registration_form_paramNUM` – поле с номером NUM

`.registration_form_paramNUM .infofield` – название параметра с номером NUM

`.registration_form_paramNUM .inptext` – текстовое поле для ввода текстового значения параметра с номером NUM

`.registration_form_paramNUM .inpnum` – текстовое поле для ввода цифрового

значения параметра с номером NUM

`.registration_form_paramNUM .inpselect{}` – список для выбора значения параметра с номером NUM

`.registration_form_paramNUM .inpcheckbox{}` – чекбокс для выбора значения параметра с номером NUM

`.registration_avatar{}` – аватар пользователя

`.registration_avatar_delete{}` – кнопка удаления аватара пользователя

`.registration_avatar .avatar{}` – изображение аватара пользователя

Шаблонная функция [show\\_login](#):

`.login{}` – шаблонная функция `show_login` – авторизация

`.login .login_avatar{}` – аватар пользователя

`.login .infofield{}` – подпись к полю в форме

`.login .inpbutton{}` – слой с кнопкой «ОК»

`.login .button{}` – кнопка «ОК»

`.login .inptext{}` – текстовые поля для ввода логина и пароля

`.login_reminding{}` – слой с ссылкой «Забыли пароль?»

`.login_registration{}` – слой с ссылкой «Регистрация»

### Восстановление пароля

`.reminding_form{}` – форма восстановления пароля

`.reminding_form .infofield{}` – подпись к полю в форме

`.reminding_form .inptext{}` – текстовые поля для ввода данных

`.reminding_form .button{}` – кнопка «Регистрация»

### Файлы

1. `modules/registration/registration.php` – контроллер модуля «Регистрация / Редактирование данных о пользователе»;
2. `modules/registration/registration.model.php` – формирование вывода формы регистрации;
3. `modules/registration/registration.view.php` – шаблон формы регистрации;
4. `modules/registration/registration.ajax.php` – обработка запроса на регистрации или обновление данных о пользователе;
5. `modules/registration/registration.js` – JS-сценарий модуля «Регистрация / Редактирование данных о пользователе»;
6. `modules/reminding/reminding.php` – контроллер модуля «Восстановление пароля»;
7. `modules/reminding/reminding.model.php` – формирование вывода формы восстановления пароля;
8. `modules/reminding/reminding.ajax.php` – обработка запроса на восстановление пароля;
9. `modules/reminding/reminding.view.php` – шаблон формы восстановления

пароля;

10.*modules/usersite/usersite.install.php* – установка модуля;

11.*modules/usersite/admin/usersite.admin.php* – редактирование пользователей;

12.*modules/usersite/admin/usersite.admin.config.php* – конфигурация модуля;

13.*modules/usersite/admin/usersite.admin.param.php* – редактирование списка дополнительных полей формы регистрации / редактирования данных;

14.*modules/usersite/admin/usersite.admin.param.js* – редактирование списка дополнительных полей формы регистрации / редактирования данных, JS-сценарий.

## Автообновление

- [Установка/удаление модулей](#)
- [Список закрытых для обновления файлов](#)
- [Восстановление базы данных](#)
- [Файлы](#)

Модуль относится к части CMS «Конфигурация» и является обязательным для установки.

Модуль позволяет обновляться в пределах минорной версии diafan.CMS. Загрузка обновленных файлов производится по протоколу FTP, поэтому необходимо указать параметры подключения в [«Параметрах сайта»](#).

## Установка/удаление модулей

Пользователь может выбрать из списка модули, которые хочет установить или удалить. Если пользователь снимает галочку с уже установленного модуля и нажимает кнопку «Установить», то модуль деинсталлируется при этом вся информация, заполненная в этом модуле удаляется. Установить/удалить можно только те модули, которые имеют файл инсталляции:  
*modules/модуль/модуль.install.php*.

## Список закрытых для обновления файлов

Если Вы вносили изменения в файлы, доступные к обновлению, Вы можете поместить их в специальный список, чтобы при обновлении случайно их не перезаписать.

Файлы, занесенные в список будут помечены сообщением «Внимание! Вы изменили файл. Его обновление приведет к потере внесенных изменений». Такой файл можно обновить, игнорируя предупреждение.

## Восстановление базы данных

Если обновление на новую минорную версию произведено некорректно или внесены непоправимые изменения в структуру таблиц базы данных, можно воспользоваться инструментом «Восстановление базы данных».

Модуль сравнивает фактические таблицы в базе данных сайта и устанавливаемые при инсталляции. Если в фактических нет необходимых полей или таблиц, они устанавливаются. Лог изменений выводится на экран.

Восстановление базы данных может понадобиться после автообновления. Если релиз включает новую функциональность, затрагивающую базу данных, то после обновления на экран выведется сообщение «Требуется обновление базы данных».

Структура базы данных описана в файлах *installation/core.php* и *modules/модуль/модуль.install.php*.

## Файлы

1. *modules/update/admin/update.admin.php* – автообновление;
1. *modules/update/admin/update.admin.install.php* – установка / удаление модулей;
2. *modules/update/admin/update.admin.list.php* – список файлов, закрытых для

автообновления;

3. *modules/update/admin/update.admin.repair.php* – восстановление базы данных;

4. *modules/модуль/модуль.install.php* – установка модулей.

# Параметры сайта

- [Файлы](#)

Модуль «Параметры сайта» позволяет редактировать общие настройки сайта. Модуль относится к части CMS «Конфигурация» и является обязательным для установки.

Модуль имеет только административную часть.

Настроить сайт можно по следующим параметрам:

## Основные

- **Название сайта** – если на сайте используются несколько языковых версий, то поле «Название сайта» нужно редактировать для каждой версии.
- **Host для базы данных** – хост для подключения к базе данных. Например, localhost. Данные обычно предоставляются хостингом при регистрации.
- **База данных** – название базы данных. Данные обычно предоставляются хостингом при регистрации или создается база данных в панели управления хостингом.
- **Пользователь базы данных** – данные обычно предоставляются хостингом при регистрации или создается база данных в панели управления хостингом.
- **Пароль для базы данных** – данные обычно предоставляются хостингом при регистрации или создается база данных в панели управления хостингом.
- **Префикс (например, diafan\_)** – все таблицы начинаются с указанного префикса.
- **Кодировка базы данных** – diafan.CMS работает с базой данных в кодировке UTF8. Изменить параметр можно в случае индивидуальной настройки системы.
- **E-mail администратора** – адрес, на который по-умолчанию приходят уведомления и который указывается в обратном адресе всех писем. Здесь может быть указан только один адрес.
- **Использовать визуальный редактор** – возможность отключить визуальный редактор по всему сайту.
- **Папка для хранения пользовательских файлов** – по-умолчанию все пользовательские файлы хранятся в папке *userfiles*.
- **Язык сайта** – основной язык системы.
- **Папка административной части** – изменение параметра означает изменение URL-адреса панели администрирования.
- **Подключить редактирование из административной части** – параметр выводит панель редактирования в пользовательской части сайта для авторизованного администратора. Подробнее читайте в разделе [«Редактирование контента из пользовательской части»](#).
- **ЧПУ оканчивается на** – по-умолчанию, все ЧПУ оканчиваются на слэш (/). Вы можете задать свое окончание для ЧПУ. Например, .html, .php, .htm и пр. При этом ссылки без ЧПУ всегда будут генерироваться по старому алгоритму:

<http://site.ru/news/show13/>

- **Генерировать ЧПУ для модулей автоматически** – формирование ЧПУ для модулей (новостей, категорий новостей, товаров, статей и пр.) в автоматическом режиме из названий.

### Режим разработки

- **Режим разработки** – включает отображение PHP-ошибок.
- **Отключить кэширование** – отключение работы с кэшем.
- **Удалить кэш** – сохранение параметров сайта с подключенной опцией «Удалить кэш» означает единократное удаление всего кэша на сайте.
- **Профилирование запросов** – выводит внизу сайта все выполненные SQL-запросы и время их выполнения.

Редактируемые модулем настройки записываются в файл *config.php*. Для редактирования данных необходимо, чтобы этот файл был доступен для записи (права доступа 0777).

**Внимание!** Ошибочно введенные параметры доступа к базе данных приведут к остановке работы всего сайта. Если это произошло, необходимо вручную редактировать файл *config.php* через FTP.

Этот файл должен храниться в кодировке UTF-8 с параметром without BOM (не использовать служебный символ *Byte Order Mark*), для редактирования используйте редактор, позволяющий править этот параметр, например, Notepad++ .

### Файлы

1. *modules/config/admin/config.admin.php*.

## Администраторы

- [Администраторы](#)
- [Права доступа](#)
- [Файлы](#)

Модуль «Администраторы» позволяет редактировать пользователей административной части сайта и их права доступа.

Модуль относится к части CMS «Конфигурация» и является обязательным для установки. Он имеет только административную часть.

## Администраторы

Кнопка  служит для удаления пользователя. Кнопка  служит для быстрой активации пользователя (или, наоборот,  блокирования).

Пользователи имеют следующие характеристики:

- **логин;**
- **пароль;**
- **e-mail;**
- **дата регистрации;**
- **ФИО;**
- **тип пользователя;**
- **активен** (возможность блокирование пользователя).

## Права доступа

Часть модуля «Права доступа» позволяет редактировать возможности типов пользователей в административной части сайта.

Кнопка  служит для удаления типа доступа.

При добавлении/редактировании типа доступа открывается таблица с возможностью разрешения определенного набора действий для каждого модуля в административной части сайта. Некоторые действия являются зависимыми от других, поэтому при выборе, например, действия «удалить» автоматически выбирается «открыть».

## Файлы

1. *modules/users/admin/users.admin.php* – редактирование администраторов;
2. *modules/users/admin/users.role.admin.php* – редактирование прав доступа.

## Языки сайта

- [Языки сайта](#)
- [Редактирование перевода интерфейса](#)
- [Файлы](#)

Модуль относится к части CMS «Конфигурация» и является обязательным для установки.

Модуль состоит из 2х частей: языки сайта и редактирование перевода интерфейса.

## Языки сайта

При добавлении и удалении языка сайта меняется структура базы данных, структура сайта. Нельзя удалить основной язык сайта.

При изменении структуры базы данных учитывается содержание массива `$this->array_language`, определенного в файле `modules/language/admin/language.admin.list.php`. Он показывает поля в таблицах, подлежащие переводу.

## Редактирование перевода интерфейса

В данной части редактируется файл `language/язык.php`

Поля для перевода берутся из файла основного языка, например `language/ru.php`.

Перевод интерфейса можно редактировать также из пользовательской части с помощью [модуля «Редактирование контента из пользовательской части»](#).

## Файлы

1. `modules/language/language.inc.php` – отдает значение языковой константы;
2. `modules/language/admin/language.admin.php` – администрирование языков сайта;
3. `modules/language/admin/language.admin.list.php` – редактирование языков сайта;
4. `modules/language/admin/language.admin.file.php` – редактирование файлов перевода интерфейса.

## Корзина

- [Файлы](#)

Модуль относится к части CMS «Конфигурация» и является обязательным для установки.

Модуль собирает данные об удаленных элементах и дает возможность восстановить или удалить их.

## Файлы

1. *modules/trash/admin/trash.admin.php*

## Карта сайта

- [Файлы](#)

Модуль генерирует карту сайта. В карту сайта входят все активные страницы сайта, не отмеченные параметром «Не показывать на карте сайта», а также прикрепленные к страницам сайта активные категории модулей, не помеченные параметром «Не показывать на карте сайта».

Модуль имеет только пользовательскую часть и является обязательным для установки.

diafan.CMS генерирует также карту сайта для поисковых систем:

<http://site.ru/sitemap.xml>

Файл *sitemap.xml* генерируется налету, физически файл не создается. Он содержит все ссылки, создаваемые системой, кроме ссылок на элементы, помеченные параметром «Не показывать на карте сайта». Для генерирования ссылок модуля необходимо наличие файла *modules/модуль/модуль.sitemap.php*. Файл должен описывать класс *Модуль\_sitemap*, содержащий метод *get*:

**boolean get (integer \$site\_id, integer \$timeedit)** – генерирует карту модуля.

- access: public
- integer \$site\_id: номер страницы сайта
- integer \$timeedit: время редактирования страницы

## Файлы

1. *modules/map/map.php* – контроллер модуля;
2. *modules/map/map.model.php* – генерирует карту сайта;
3. *modules/map/map.view.php* – шаблон карты сайта;
4. *modules/модуль/модуль.sitemap.php* – карта сайта модуля.

## Поиск по сайту

- [CSS-форматирование](#)
- [Файлы](#)

Модуль выводит результаты поиска и форму для поиска. Модуль имеет только пользовательскую часть и является обязательным для установки.

С модулем связано шаблонная функция [show\\_search](#), которая выводит форму поиска на всех страницах кроме страницы результатов поиска.

## CSS-форматирование

Страница результатов поиска:

[.search\\_result](#){ } – результаты поиска – общая статистика

[.search\\_list](#){ } – результаты поиска – один элемент поисковой выдачи

Шаблонная функция [show\\_search](#):

[.search](#){ } – форма поиска

[.search .input\\_search](#){ } – поле для ввода выражения для поиска

[.search .submit\\_search](#){ } – кнопка «Найти»

## Файлы

1. *modules/search/search.php* – контроллер модуля;
2. *modules/search/search.model.php* – генерирует результаты поиска;
3. *modules/search/search.view.php* – шаблон поиска по сайту.

# Редактирование контента из пользовательской части

- [Подключение](#)
- [Файлы](#)

Модуль позволяет редактировать данные модулей и интерфейс из пользовательской части сайта. Модуль состоит из двух частей: подключаемая часть и обработка запроса на редактирование в административной части.

## Подключение

Подключаемая часть – файл `modules/useradmin/useradmin.inc.php`. В нем описан класс `Useradmin_inc`. В модуле к объекту классам можно обратиться через переменную `$this->diafan->_useradmin`. Экземпляр класса создается при первом вызове переменной.

## Методы

`string get (string $text, string $name, integer $element_id, string $module_name, [integer|string $lang_id = "], [string $type = "])` – генерирует ссылку на форму редактирования.

- access: public
- string \$text: значение переменной
- string \$name: название переменной
- integer \$element\_id: номер элемента
- string \$module\_name: модуль
- integer|string \$lang\_id: номер языка
- string \$type: тип данных

`string get_lang (string $name)` – генерирует ссылку на форму языковой константы.

- access: public
- string \$name: название константы

`boolean edit ()` – генерирует данные для формы редактирования.

- access: public

`string role (string $module_name)` – определяет есть ли права на сохранение для модуля у администратора.

- access: public
- string \$module\_name: название модуля

## Пример

```
// для администратора сделаем доступным редактирование текста
новости
$text = $this->diafan->_useradmin->get($text, 'text', $id, 'news',
_LANG);
```

Работа с модулем должна производиться вне зоны кэширования.

## Файлы

1. *modules/useradmin/useradmin.inc.php* – обеспечивает редактирование данных из пользовательской части;
2. *modules/useradmin/useradmin.view.php* – шаблон вывода модуля;
3. *modules/useradmin/useradmin.edit.php* – вывод формы редактирования данных;
4. *modules/useradmin/useradmin.js* – JS-сценарий открытия формы редактирования, функционирования панели редактирования;
5. *modules/useradmin/useradmin.css* – оформление панели редактирования, редактируемых областей;
6. *modules/useradmin/useradmin.edit.js* – JS-сценарий обработки формы редактирования данных;
7. *modules/useradmin/useradmin.edit.css* – оформление формы редактирования данных;
8. *modules/useradmin/admin/useradmin.admin.ajax.php* – обработка Ajax-запросов в административной части модуля.

# Прикрепленные изображения

- [Подключение](#)
- [Как добавить прикрепленные изображения к модулю](#)
- [Настройки](#)
- [Файлы](#)

Модуль работает с изображениями, прикрепленными к элементам других модулей. Модуль состоит из двух частей: подключаемая часть и работа модуля в административной части.

## Подключение

Подключаемая часть – файл `modules/images/images.inc.php`. В нем описан класс `Images_inc`. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_images`. Экземпляр класса создается при первом вызове переменной.

## Методы

array **get** (**integer \$id**, **string \$salt**, **string \$module\_name**, [**integer \$count = 0**], [**string|integer \$common\_link = 1**], [**boolean \$is\_category = FALSE**]) – получает изображения, прикрепленные к элементу модуля.

- access: public
- integer \$id: номер элемента, к которому прикреплены изображения
- string \$salt: альтернативный текст получаемых изображений
- string \$module\_name: название модуля, по умолчанию модуль, прикрепленный к текущей странице
- integer \$count: количество изображений
- string|integer \$common\_link: ссылка, на которую ведет изображение | 1 — ссылка ведет на увеличенную копию изображений
- boolean \$is\_category: изображения прикреплены к категории

array **get\_big** (**integer \$id**, **string \$salt**, **string \$module\_name**, [**integer \$count = 0**], [**boolean \$is\_category = FALSE**]) – получает большие копии изображений, прикрепленных к элементу модуля.

- access: public
- integer \$id: номер элемента, к которому прикреплены изображения
- string \$salt: альтернативный текст получаемых изображений
- string \$module\_name: название модуля, по умолчанию модуль, прикрепленный к текущей странице
- integer \$count: количество изображений
- boolean \$is\_category: изображения прикреплены к категории

## Пример

```
// получаем все изображения, прикрепленные к товару в модуле «Магазин» в файле modules/shop/shop.model.php
$images = $this->diafan->_images->get($id, "Название товара", "shop");
```

```

// выводим полученные файлы в шаблоне модуля в файле modules/shop/
shop.view.php
foreach($images as $img)
{
    switch($img["type"])
    {
        case 'animation':
            echo '<a href="'.BASE_PATH.$img["link"].'"
            rel="prettyPhoto[gallery' . $id . 'shop]'>';
            break;
        case 'big_image':
            echo '<a href="'.BASE_PATH.$img["link"].'"
            rel="big_image" width="'. $img["link_width"].'"
            height="'. $img["link_height"].'">';
            break;
        default:
            echo '<a href="'.BASE_PATH_HREF.$img["link"].'">';
            break;
    }
    echo ''
    . '</a> ';
}
}

```

## Как добавить прикрепленные изображения к модулю

В конфигурации модуля (*modules/модуль/admin/модуль.admin.config.php*) и в файле редактирования элементов модуля (*modules/модуль/admin/модуль.admin.php*) необходимо добавить параметр **'images'**:

```
$this->variable_table = array(..., 'images' => 'module', ...);
```

## Настройки

Модуль настраивается в конфигурации модуля, к которому прикрепляются изображения. Необходимо определить следующие параметры работы модуля:

**Использовать изображения** – параметр позволяет включить/отключить загрузку изображений.

**Размер большого изображения** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер маленького изображения** – размер изображения предпросмотра.

**Отображение изображений в списке** – параметр принимает значения:

- нет (отключает отображение изображений в списке);
- показывать одно изображение;
- показывать все изображения.

**Использовать изображения для категории** – параметр позволяет

включить/отключить загрузку изображений для категорий.

**Размер большого изображения для категории** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер маленького изображения для категории** – размер изображения предпросмотра.

**Отображение изображений категорий в списке категорий** – параметр принимает значения:

- нет (отключает отображение изображений в списке);
- показывать одно изображение;
- показывать все изображения.

**Использовать анимацию при увеличении изображений** – параметр добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений.

**Максимальный размер загружаемых файлов** – параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.

**Накладывать водяной знак** – параметр позволяет включить/отключить наложение водяного знака на изображение.

**Расположение водяного знака** – расположение водяного знака в пикселях от левого края или от центра и от нижнего края или от центра.

**Использовать водяной знак только для большого изображения** – отключает наложение водяного знака на изображение предпросмотра.

**Применить настройки ко всем ранее загруженным изображениям** – позволяет перекодировать размер уже загруженных изображений. Размер вновь созданных изображений не может быть больше размера большой копии уже загруженного изображения.

## Файлы

1. *modules/images/images.inc.php* – подключение для работы с прикрепленными файлами;
2. *modules/images/admin/images.admin.view.php* – шаблон вывода прикрепленных изображений в административной части;
3. *modules/images/admin/images.admin.edit.php* – вывод прикрепленных изображений в административной части модулей;
4. *modules/images/admin/images.admin.save.php* – сохранение прикрепленных изображений в административной части модулей;
5. *modules/images/admin/images.admin.del.php* – удаление прикрепленных изображений при удалении элементов модуля;
6. *modules/images/admin/images.admin.ajax.php* – обработка Ajax-запросов в административной части модуля;
7. *modules/images/admin/images.admin.js* – редактирование прикрепленных изображений, JS-сценарий.

# Прикрепленные файлы

- [Подключение](#)
- [Как добавить прикрепленные файлы к модулю](#)
- [Настройки](#)
- [Файлы](#)

Модуль работает с файлами, прикрепленными к элементам других модулей. Модуль состоит из трех частей: подключаемая часть, работа модуля в административной части и файл, отдающий прикрепленные файлы.

## Подключение

Подключаемая часть – файл `modules/attachments/attachments.inc.php`. В нем описан класс `Attachments_inc`. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_attachments`. Экземпляр класса создается при первом вызове переменной.

## Методы

`array get(integer $element_id, string $module_name, [integer] boolean $attachment_id = FALSE)` – отдает информацию о прикрепленных файлах/файле.

- access: public
- integer \$element\_id: номер элемента, к которому прикрепляется файл
- string \$module\_name: название модуля
- integer|boolean \$attachment\_id: номер файла

`string save(integer $element_id, string $module_name, [integer] $site_id = 0)` – сохраняет добавленные файлы.

- access: public
- integer \$element\_id: номер элемента, к которому прикрепляется файл
- string \$module\_name: название модуля
- integer \$site\_id: страница, к которой прикреплен модуль

`boolean|string upload(array $file, string $module_name, integer $element_id, [integer|boolean] $n = FALSE)` – загружает файлы.

- access: public
- array \$file: загружаемый файл/файлы
- string \$module\_name: название модуля
- integer \$element\_id: номер элемента, к которому прикрепляется файл
- integer|boolean \$n: false \$n номер файла в массиве файлов, если предан массив

`boolean delete(integer $element_id, string $module_name, [integer] boolean $attachment_id = FALSE)` – удаляет прикрепленные файлы/файл.

- access: public
- integer \$element\_id: номер элемента, к которому прикрепляется файл
- string \$module\_name: название модуля
- integer|boolean \$attachment\_id: номер файла

## Пример

```
// получаем файлы, прикрепленные к вопросу в модуле «Вопрос-Ответ»
// в файле modules/faq/faq.model.php
$attachments = $this->diafan->_attachments->get($id, 'faq');

// выводим полученные файлы в шаблоне модуля в файле
modules/faq/faq.view.php
foreach($attachments as $att)
{
    if($att["is_image"])
    {
        if($row["use_animation"])
        {
            $a_href = '<a href="'.att["link"]
                .'" rel="prettyPhoto[gallery]'.att["element_id"]
                .att["module_name"].'_1">';
            $a_href2 = '<a href="'.att["link"]
                .'" rel="prettyPhoto[gallery]'.att["element_id"]
                .att["module_name"].'_2">';
        }
        else
        {
            $a_href .= '<a href="'.att["link"]
                .'" rel="big_image" width="'.att["width"]
                .'" height="'.att["height"].'">';
            $a_href2 = $a_href;
        }
        echo '<p id="attachment'.att["id"].'">'
            . $a_href.att["name"].'</a> ('.att["size"].')'
            . ' '. $a_href2
            . '</a></p>';
    }
    else
    {
        echo '<p id="attachment'.att["id"].'"><a href="'
            . att["link"].'">'.att["name"].'</a> ('
            . att["size"].')</p>';
    }
}

// загружаем файлы, прикрепленные к задаваемому вопросу в файле mo
dules/faq/faq.ajax.php
$result = $this->diafan->_attachments->save($id, "faq", $site_id);
```

## Как добавить прикрепленные файлы к модулю

В конфигурации модуля (*modules/модуль/admin/модуль.admin.config.php*) и в файле редактирования элементов модуля (*modules/модуль/admin/модуль.admin.php*) необходимо добавить параметр **'attachments'**:

```
$this->variable_table = array(..., 'attachments' => 'module', ...);
```

## Настройки

Модуль настраивается в конфигурации модуля, к которому прикрепляются файлы. Все настройки являются необязательными для правильной работы модуля «Прикрепленные файлы». Можно определить следующие параметры работы модуля:

**Разрешить добавление файлов** – позволяет пользователям прикреплять файлы к элементу модуля.

**Максимальное количество добавляемых файлов** – количество добавляемых файлов.

### Доступные типы файлов (через запятую)

Пример: *doc, gif, jpg, mpg, pdf, png, txt, zip*

**Распознавать изображения** – позволяет прикрепленные к сообщению файлы в формате JPEG, GIF, PNG отображать как изображения.

**Размер для большого изображения** – размер изображения, отображаемый в пользовательской части сайта при увеличении изображения предпросмотра.

**Размер для маленького изображения** – размер изображения предпросмотра.

**Использовать анимацию при увеличении изображений** – добавляет JavaScript код, позволяющий включить анимацию при увеличении изображений.

**Максимальный размер загружаемых файлов** – параметр показывает максимально допустимый размер загружаемых файлов, установленный в настройках хостинга.

Загруженные файлы открываются по адресу:

[http://www.site.ru/attachments/get/номер\\_файла/название\\_файла](http://www.site.ru/attachments/get/номер_файла/название_файла)

## Файлы

1. *modules/attachments/attachments.inc.php* – подключение для работы с прикрепленными файлами;
2. *modules/attachments/attachments.get.php* – отдает файл;
3. *modules/attachments/admin/attachments.admin.edit.php* – вывод прикрепленных файлов в административной части модулей;
4. *modules/attachments/admin/attachments.admin.save.php* – сохранение прикрепленных файлов в административной части модулей;
5. *modules/attachments/admin/attachments.admin.del.php* – удаление прикрепленных файлов при удалении элементов модуля.

## Постраничная навигация

- [Подключение](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль предназначен для формирования постраничной навигации контента. Модуль состоит только из подключаемой части и шаблона.

### Подключение

Подключаемая часть – файл *modules/paginator/paginator.inc.php*. В нем описан класс *Paginator\_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_paginator`. Экземпляр класса создается при первом вызове переменной.

### Свойства

string **\$get\_nav**

- var: GET переменные, участвующие в навигации для постраничного вывода
- access: public

integer **\$nastr = 10**

- var: количество элементов, показанных на странице
- access: public

string **\$navlink**

- var: ссылка на первую страницу
- access: public

string **\$navlink\_tpl**

- var: шаблон ссылки второй и последующих страниц, если не задан используется `navlink + urlpage`
- access: public

integer **\$navnastr = 10**

- var: количество ссылок постраничной навигации, показанных на одной странице
- access: public

integer **\$nen**

- var: количество элементов в списке
- access: public

integer **\$page**

- var: номер страницы
- access: public

integer **\$polog = 0**

- var: порядковый номер элемента, с которого начинается вывод элементов
- access: public

string **\$urlpage = 'page%d/'**

- var: шаблон части ссылки, отвечающей за передачу номера страницы
- access: public

## Методы

array **get()** – Формирует строку навигации

- access: public

## Пример

```
// 1. Определяем свойства класса:

// номер текущей страницы
$this->diafan->_paginator->page = $this->diafan->page;

// адрес первой страницы
// вычисляем с помощью подключения "Маршрутизатор"
$this->diafan->_paginator->navlink = $this->diafan->_route-
>current_link("page");
// или задаем вручную
$this->diafan->_paginator->navlink = "news.html";

// шаблон последующих страниц
// вычисляем с помощью подключения "Маршрутизатор"
$this->diafan->_paginator->navlink_tpl = $this->diafan->_route-
>current_link("", array("page" =>
"%d"));
// или задаем вручную
$this->diafan->_paginator->navlink_tpl = 'news/page%d/';

// вместо переменной navlink_tpl можно задать utlpage, тогда
navlink_tpl будет равна navlink + urlpage
$this->diafan->_paginator->urlpage = 'page%d/';

// количество новостей
$this->diafan->_paginator->nen = DB::query_result("SELECT COUNT(*)
FROM {news}");

// 2. Получаем список ссылок в постраничной навигации
$links = $this->diafan->_paginator->get();

// 3. Отправляем запрос к базе данных для получения списка
новостей, ограниченных данной страницей
$result = DB::query_range("SELECT * FROM {news}", $this->diafan-
->_paginator->polog, $this->diafan->_paginator->nastr);

// 4. Подставляем полученные данные в шаблон
// в модели модуля
$this->result["paginator"] = $this->diafan->_tpl->get('get', 'pagi
nator', $links);
```

```
// или в шаблоне модуля  
echo $this->get('get', 'paginator', $links);
```

В шаблоне описаны две функции: `get` и `get_admin`. Первая используется для пользовательской части, вторая для административной. Вы можете создать свой шаблон функцию и использовать ее в своем модуле.

## CSS-форматирование

`.paginator` – навигация «Постраничный вывод»

## Файлы

1. `modules/paginator/paginator.inc.php` – формирует постраничную навигацию;
2. `modules/paginator/paginator.view.php` – шаблон постраничной навигации.

## Защитный код

- [Подключение](#)
- [CSS-форматирование](#)
- [Файлы](#)

Модуль выводит защитный код (капчу) в формах других модулей. Модуль состоит из подключаемой части, шаблона вывода каптчи и файла, генерирующего каптчу.

## Подключение

Подключаемая часть – файл *modules/captcha/captcha.inc.php*. В нем описан класс *Captcha\_inc*. В модуле к объекту класса можно обратиться через переменную `$this->diafan->_captcha`. Экземпляр класса создается при первом вызове переменной.

## Методы

`string get([string $modules = "modules"], [ $error = ""], [boolean $is_update = FALSE])` – выводит каптчу.

- access: public
- string \$modules: метка каптчи
- string \$error: ошибка ввода кода, если запрос передан не через Ajax
- boolean \$is\_update: каптча генерируется для обновления

`string|boolean is_right([string $modules = "modules"])` – проверяет правильность ввода каптчи.

- return: false
- access: public
- string \$modules: метка каптчи

## Пример

Вывод защитного кода в форме модуля «Вопрос-Ответ».

```
// в модели modules/faq/faq.model.php запрашиваем каптчу
$captcha = $this->diafan->_captcha->get("faq", $this->get_error("faq", "code"));

//выводим защитный код в шаблоне modules/faq/faq.view.php
echo $captcha;

//запрашиваем правильно ли введен защитный код при обработке
запроса в файле modules/faq/faq.ajax.php
$error = $this->diafan->_captcha->is_right('faq');
if($error)
{
    // защитный код введен неправильно
}
```

## CSS-форматирование

- `.code{}` – форма ввода защитного кода
- `.code_enter{}` – надпись «Введите защитный код»
- `.code_img{}` – изображение с защитным кодом
- `.code_input{}` – поле для ввода защитного кода
- `.code_update{}` – ссылка «Обновить защитный код»

## Файлы

1. `modules/captcha/captcha.inc.php` – формирует код вывода каптчи;
2. `modules/captcha/captcha.view.php` – шаблон вывода каптчи;
3. `modules/captcha/captcha.get.php` – генерирует изображение каптчи.

## III. Подключения

**Подключение** – набор функций подключаемых для осуществления определенных целей (работа с файлами, работа с базой данных, работа с изображениями).

### Базы данных

Класс для работы с базой данных – *DB* определен в файле *includes/database.php*.

Для фильтрации аргументов SQL-запроса, вместо данных передаются маски, а данные передаются в качестве дополнительных аргументов для функций `DB::query()`, `DB::query_result()` и `DB::query_range()` в порядке, в котором они следуют в SQL-запросе. Существуют следующие маски:

- `%d` – число,
- `%s` – строка,
- `%h` – строка без HTML кода,
- `%%` - символ %,
- `%f` – число с плавающей точкой,
- `%b` – файлы

Название таблицы пишется без префикса в фигурных скобках.

Пример: `{attachments}`

Переводимые переменные в SQL-запросе пишутся в квадратных скобках.

Пример: `[name]`

Чтобы вывести на экран сгенерированный SQL-запрос, нужно перед запросом написать "DEV " и включить режим разработки.

Пример:

```
$this->result["text"] = DB::query_result("DEV SELECT [text] FROM {shop_category} WHERE id=%d", $this->diafan->cat);
```

### Методы

resource **connect(string \$url)** – пробует подключиться к базе данных.

- access: public
- string \$url: доступ к базе данных

array **fetch\_array(\$result)** – извлекает результирующий ряд как массив.

- access: public
- \$result

object **fetch\_object(\$result)** – извлекает результирующий ряд как объект.

- access: public
- \$result

integer **next\_id (string \$table)** – получает номер id для таблицы \$table.

- access: public
- string \$table: название таблицы без префикса

integer **num\_rows(\$result)** – получает количество рядов в результате.

- access: public
- \$result

resource **query(\$query)** – отправляет запрос к базе данных.

- access: public
- \$query

resource **query\_range(\$query)** – отправляет запрос к базе данных с лимитом на количество получаемых в результате рядов.

- access: public
- \$query

mixed **query\_result(\$query)** – получает результирующие данные из SQL-запроса.

- access: public
- \$query

mixed **result(\$result, [ \$row = 0])** – получает результирующие данные.

- access: public
- \$result
- \$row

mixed **title(string \$table, integer \$id, string \$param\_name)** – возвращает значение поля \$param\_name для таблицы \$table и строки с номером \$id.

- access: public
- string \$table: название таблицы без префикса
- integer \$id: идентификатор строки
- string \$param\_name: название поля

mixed **where(string \$table, string \$where, string \$param\_name)** – возвращает значение поля \$param\_name для таблицы \$table и одной строки, соответствующей запросу \$where.

- access: public
- string \$table: название таблицы без префикса
- string \$where: SQL-условие
- string \$param\_name: название поля

# Валидация данных

Методы для валидации данных объединены в класс *Validate*, описанный в файле *includes/validate.php*.

## Методы

`string|boolean login(string $name)` – проверяет логин на валидность.

- return: false
- access: public
- string \$name: логин

`string|boolean mail(string $mail)` – проверяет e-mail на валидность.

- return: false
- access: public
- string \$mail: электронный ящик

`string|boolean mail_user(string $mail)` – проверяет зарегистрирован ли пользователь с указанным электронным ящиком.

- return: false
- access: public
- string \$mail: электронный ящик

`string|boolean password(string $password)` – проверяет пароль на валидность.

- return: false
- access: public
- string \$password: пароль

`string|boolean text(string $text, [integer $max_length = 40])` – проверяет текст на наличие длинных слов.

- return: false
- access: public
- string \$text: текст
- integer \$max\_length: максимальная длина слова

`boolean url(string $url, [boolean $absolute = FALSE])` – проверяет URL на валидность.

- access: public
- string \$url: электронный адрес
- boolean \$absolute: абсолютный адрес

# Изображения

Для работы с изображениями в файле *includes/image.php* определен класс *Image*.

## Методы

**void `resize` (string `$src_image`, integer `$dest_width`, integer `$dest_height`, [integer `$quality` = 80], [boolean `$max` = FALSE])** – изменяет размеры изображения.

- access: public
- string `$src_image`: путь к файлу
- integer `$dest_width`: новая ширина изображения
- integer `$dest_height`: новая высота изображения
- integer `$quality`: качество изображения
- boolean `$max`: изменять по максимальной стороне

**boolean `watermark` (string `$original`, string `$watermark`, string `$placement`)** – добавляет водяной знак на изображение.

- access: public
- string `$original`: путь к файлу
- string `$watermark`: путь к водяному знаку
- string `$placement`: расположение знака

# Конфигурация сайта

Работа с файлом *config.php* – параметры сайта осуществляется с помощью подключения «Конфигурация сайта» (файл *includes/config.php*, класс *Config*).

## Методы

boolean true **save (array \$new\_values, array \$language\_all)** – сохраняет параметры сайта.

- access: public
- array \$new\_values: новые значения параметров сайта
- array \$language\_all: языки сайта

Пример:

```
//изменяем параметр «Режим разработки»
include_once(ABSOLUTE_PATH.'includes/config.php');

$new_values = array('MOD_DEVELOPER' => TRUE);
$language_all = array(
    array('', 'ru'),
    array(2, 'eng')
);
Config::save($new_values, $language_all);
```

## Кэширование

Кэширование обеспечивает класс *Cache*, описанный в файле *includes/cache.php*.

Время актуальности кэша не ограничено. Кэш чистится при редактировании контента модуля из административной части.

В [модуле «Параметры сайта»](#) предусмотрены две опции для управления кэшем: «Отключить кэширование» и «Удалить кэш».

### Методы

boolean **delete(string|array \$name, [string \$module = 'site'])** – удаляет кэш для модуля *\$module* с меткой *\$name*. Если функция вызвана с пустой меткой, то удаляется весь кэш для модуля *\$module*.

- return: true
- access: public
- string|array *\$name*: метка кэша
- string *\$module*: название модуля

mixed **get(string|array \$name, [string \$module = 'site'], [boolean \$ignore\_mod\_dev\_cache = FALSE])** – читает кэш модуля *\$module* с меткой *\$name*.

- access: public
- string|array *\$name*: метка кэша
- string *\$module*: название модуля
- boolean *\$ignore\_mod\_dev\_cache*: игнорировать отключение кэша в параметрах сайта

boolean **save(mixed \$data, string|array \$name, [string \$module = 'site'])** – сохраняет данные *\$data* для модуля *\$module* с меткой *\$name*.

- return: true
- access: public
- mixed *\$data*: данные, сохраняемые в кэше
- string|array *\$name*: метка кэша
- string *\$module*: название модуля

### Примеры

```
$cache_meta = array("name" => "block", "cat_id" => $cat_id,
"language" => _LANG, ...);
if(! $result = $this->diafan->_cache->get($cache_meta, "модуль"))
{
    // генерирование кэша
    // сохранение кэша
    $this->diafan->_cache->save($result, $cache_meta, "модуль");
}
```

# Маршрутизатор

*Route* – класс для работы с ЧПУ. Описан он файле `includes/route.php`. Объект класса создается в административной и пользовательской частях и в модулях доступен для обращения через переменную `$this->diafan->_route`.

## Методы

`string link(integer $site_id, [string $module_name = "], [integer $cat_id = 0], [integer $element_id = 0], [integer $param_id = 0])` – генерирует ссылку.

- access: public
- integer \$site\_id: номер страницы сайта
- string \$module\_name: модуль
- integer \$cat\_id: номер категории
- integer \$element\_id: номер элемента модуля
- integer \$param\_id: номер параметра

`string module(string $module_name)` – получает ЧПУ страницы сайта по названию модуля.

- access: public
- string \$module\_name: название модуля

`string current_link([string|array $exclude = "], [array $include = array()])` – выдает URL текущей страницы с включенными или исключенными переменными.

- access: public
- string|array \$exclude: исключенные переменные
- array \$include: включенные переменные

`string current_admin_link([string|array $exclude = "])` – выдает URL текущей страницы административной части с включенными переменными.

- access: public
- string|array \$exclude: исключенные переменные

`array|boolean false search(string $rewrite, [boolean $arguments_in_url = FALSE])` – ищет псевдоссылку в базе данных.

- access: public
- string \$rewrite: текущая псевдоссылка
- boolean \$arguments\_in\_url: в URL переданы аргументы

`boolean true get_rewrite(string $module_name, integer $id, [string $tag = ""])` – получает ЧПУ по тегу.

- access: private
- string \$module\_name: модуль
- integer \$id: номер элемента
- string \$tag: тег

## Примеры

```
// получаем ссылку на товар
$link = BASE_PATH_HREF
.$this->diafan->_route->link($row["site_id"], 'shop', $row["id"]);
```

## Общие функции

Общие функции принадлежат к абстрактному классу *Core*, который наследуются другими классами. Эти функции доступны через основной объект `$diafan`, создание которого инициирует генерирование страницы. В модулях функции доступны для вызова через объект `$this->diafan`.

```
Пример: $this->diafan->configmodules("list_img");
```

`void redirect([string $url = "], [integer $http_response_code = 302])` – редирект. Адрес для редиректа можно передавать как в абсолютном виде (`http://www.site.ru/`), так и в относительном (`news/`).

- access: public
- string \$url: URL для редиректа
- integer \$http\_response\_code: статус-код

`void redirect_js([string $url = "])` – редирект с помощью JavaScript. Используется в случае, если нельзя использовать редирект с помощью заголовков страниц.

- access: public
- string \$url: URL для редиректа

`integer unixdate(string $date)` – возвращает дату, переданную в формате dd.mm.yyyy hh:ii в виде даты в [формате UNIX](#).

- access: public
- string \$date: дата в формате dd.mm.yyyy hh:ii

`mixed get_param(array $array, string $name, [mixed $default = null], [integer $mask = 0])` – функция получает из массива переменную и приводя ее к типу в соответствии с маской (0 – вырезает все пробелы, 1 – вырезает тэги, 2 – оставляет только числа).

- access: public
- array \$array: массив с переменной
- string \$name: имя переменной в массиве
- mixed \$default: значение по-умолчанию
- integer \$mask: тип преобразования

```
Пример: $id = $this->diafan->get_param($_GET, 'cid', 0, 2);
```

`mixed configmodules (string $name, [string $module_name = 'site'], [integer $site_id = 0], [integer $lang_id = 0])` – функция возвращает значение переменной `$name` в конфигурации модуля `$module_name` для языковой версии `$lang_id` и страницы `$site_id`.

- access: public
- string \$name: имя переменной в конфигурации
- string \$module\_name: название модуля
- integer \$site\_id: раздел сайта
- integer \$lang\_id: номер языковой версии

`string path_module([integer $page_act = 0], [integer $parentid_act = 0], [integer $config_act = 0], [integer $cat_act = 0], [integer $catid_act = 0])` – возвращает url страницы с сохранением переменных `page`, `parent`, `config`, `cat`, `catid`. Можно

отказаться от сохранения значений этих переменных в адресе, передав необязательные параметры.

- access: public
- integer \$page\_act: убрать из адреса текущей страницы переменную page
- integer \$parent\_act: убрать из адреса текущей страницы переменную parent
- integer \$config\_act: убрать из адреса текущей страницы переменную config
- integer \$cat\_act: убрать из адреса текущей страницы переменную cat
- integer \$catid\_act: убрать из адреса текущей страницы переменную catid

Пример: `$this->diafan->path_module(1, 0, 0, 0, 1);`

выдаст адрес без сохранения переменной page (страница) и catid (раздел).

string **short\_text(string \$text, [integer \$count = 80])** – возвращает сокращенный текст.

- access: public
- string \$text: текст
- integer \$count: количество символов для сокращения

string **convert(integer \$size)** – конвертирует количество бит в байты, килобайты, мегабайты.

- access: public
- integer \$size: размер в байтах

string **search\_theme(intger \$id)** – выводит название шаблона для страницы с номером **id**.

- access: public
- integer \$size: размер в байтах

array **get\_children(integer \$id, string \$table)** – выдает массив номеров детей.

- access: public
- integer \$id: номер исходного элемента
- string \$table: таблица

array **get\_parents(integer \$id, string \$table)** – выдает массив номеров родителей.

- access: public
- integer \$id: номер исходного элемента
- string \$table: таблица

string **code\_password(string \$value)** – кодирует пароль.

- access: public
- string \$value: исходный пароль

string **translit(string \$text)** – переводит кириллицу в транслит для строки **text**.

- access: public
- string \$text: исходный текст

string **format\_price(integer \$price)** – формтирует цену **price** в соответствии с конфигурацией модуля «Магазин».

- access: public
- integer price: исходная цена

## Подключение шаблонов

Для работы с шаблонами сайта в файле `includes/template.php` описан класс `Template`. В модуле объект этого класса доступен через переменную `$this->diafan->_tpl`, в шаблоне модуля объект класса доступен через переменную `$this`.

### Методы

`mixed get(string $name, string $module, string $result)` – подключает шаблон.

- access: public
- string \$name: имя функции
- string \$module: название модуля
- mixed \$result: передаваемые в шаблон данные

`mixed _(string $name, [boolean $useradmin = TRUE])` – получает значение языковой константы.

- access: public
- string \$name: имя константы, без приставки `_LANG_`
- boolean \$useradmin: выдавать форму для редактирования

### Примеры

```
// подключаем шаблон первой страницы модуля «Магазин» в файле
modules/shop/shop.php
$this->diafan->_tpl->get('first_page', 'shop', $result);

// выводим языковую константу в шаблоне модуля «Магазин» в файле
modules/shop/shop.view.php
echo $this->_('PRICE');

// подключаем шаблон формы авторизации в шаблоне модуля
«Оформление заказа» в файле modules/order/order.view.php
echo $this->get('show_login', 'registration', $result);
```

# Пользователи

Работа с пользователями разделена на две части: пользовательскую (класс *User* файл *includes/user.php*) и административную (класс *User\_admin* файл *adm/includes/user.php*).

Объект класса доступен в модулях как `$this->diafan->_user`.

## User

Работа с пользователями в пользовательской части.

### Свойства

boolean **\$admin**

- var: пользователь является администратором
- access: public

array **\$admin\_roles**

- var: права доступа администратора
- access: public

boolean **\$checked**

- var: пользователь проверен по идентификационному хэшу
- access: public

integer **\$created**

- var: дата регистрации текущего пользователя
- access: public

string **\$errauth**

- var: ошибка авторизации
- access: public

string **\$fio**

- var: ФИО пользователя
- access: public

array **\$fields**

- var: характеристики пользователя
- access: public

integer **\$id = 0**

- var: идентификатор пользователя
- access: public

integer **\$language**

- var: язык интерфейса, который выбрал текущий пользователь
- access: public

string **\$mail**

- var: e-mail пользователя

- access: public

boolean **\$moderator**

- var: модератор
- access: public

string **\$name**

- var: логин пользователя
- access: public

string **\$session**

- var: данные, хранящиеся в сессии
- access: public

## User\_admin

Работа с пользователями в административной части.

### Свойства

boolean **\$checked**

- var: пользователь проверен по идентификационному хэшу
- access: public

integer **\$created**

- var: дата регистрации текущего пользователя
- access: public

string **\$errauth**

- var: ошибка авторизации
- access: public

string **\$fio**

- var: ФИО пользователя
- access: public

integer **\$id = 0**

- var: идентификатор пользователя
- access: public

string **\$mail**

- var: e-mail пользователя
- access: public

string **\$name**

- var: логин пользователя
- access: public

array **\$roles = array()**

- var: права доступа текущего пользователя
- access: public

string **\$session**

- var: данные, хранящиеся в сессии
- access: public

## Методы

(одинаковые для двух классов)

boolean **anonymous**([string \$session = ""]) – определяет, что текущий пользователь не авторизован

- return: true
- access: public
- string \$session: данные, записанные в сессии

boolean **auth**(\$form\_values) – проверяет авторизован ли пользователь.

- access: public
- \$form\_values

string **get\_hash**() – генерируем идентификационный пользовательский хэш.

- access: public

boolean **logout**() – очищает информацию о текущем пользователе.

- return: true
- access: public

boolean **set**(object \$user) – определяет текущего пользователя.

- return: true
- access: public
- object \$user: данные о текущем пользователе

## Почта

Работа с электронной почтой основана на плагине [PHPMailer](#). Соединение diafan.CMS и плагина обеспечивает функция `send_mail`, описанная в файле `includes/mail.php`.

**boolean `send_mail(string|array $recipient, string $subject, string $body, [string $from = "], [boolean $mode = TRUE])`** – отправляет электронное письмо.

- `string|array $recipient`: получатель/получатели
- `string $subject`: тема письма
- `string $body`: содержание письма
- `string $from`: адрес отправителя
- `boolean $mode`: формат письма (текстовый/HTML)

## Режим разработки

В diafan.CMS предусмотрены специальные инструменты для разработчика: режим разработки (отображает все PHP-ошибки) и профилирование запросов.

Они описаны в классе *Dev* файле *includes/dev.php*.

### Методы

boolean **get\_profiling()** – профилирование запросов.

- access: public

boolean **set\_error()** – разрешает/запрещает вывод ошибок.

- access: public

boolean **set\_profiling()** – активирует профилирование запросов, если это разрешено в параметрах.

- access: public

# Сессии

Работа с сессией разделена на две части: пользовательскую (класс *Session* файл *includes/session.php*) и административную (класс *Session\_admin* файл *adm/includes/session.php*).

## Методы

(одинаковые для двух классов)

boolean **close()** – закрывает сессию освобождает ресурсы.

- return: true
- access: public

boolean **destroy(string \$key)** – удаляет сессию.

- return: true
- access: public
- string \$key: идентификатор сессии

boolean **gc([integer \$lifetime = 1814400])** – чистит мусор - удаляет сессии старше \$lifetime.

- return: true
- access: public
- integer \$lifetime: время хранения сессии в секундах

boolean **init()** – стартует сессию.

- return: true
- access: public

boolean **open(string \$save\_path, string \$session\_name)** – открывает сессию.

- return: true
- access: public
- string \$save\_path
- string \$session\_name

string **read(string \$key)** – читает сессию.

- return: сериализованные данные сессии
- access: public
- string \$key: идентификатор сессии

boolean **write(string \$key, string \$value)** – записывает данные в сессию.

- return: true
- access: public
- string \$key: идентификатор сессии
- string \$value: сериализованные данные сессии

## Сжатие страниц

Для сжатия страницы в файле *includes/gzip.php* определен класс *Gzip*.

### Методы

boolean **do\_gzip**([ *\$rewrite* = ""]) – выдает сжатые данные.

- return: true
- string *\$rewrite*: текущая псевдоссылка

boolean **init\_gzip**() – инициализирует сжатие.

- return: true

## IV. Плагины

**Плагины** – функционал сторонних разработчиков, используемый для выполнения определенных задач.

diafan.CMS версии 4.3 использует следующие плагины:

- [PHPMailer](#) – работа с электронной почтой;
- [Kohana](#) – заимствована работа с кодировкой UTF8;
- [FTP от David Grudl](#) – доступ к FTP серверу;
- [JsHttpRequest](#) – заимствовано преобразование данных в формат JSON;
- [KCAPTCHA](#) – работа с защитным кодом;
- [jQuery](#) – JavaScript библиотека;
- [jQuery Form Plugin](#) – jQuery плагин для работы с формами;
- [prettyPhoto](#) – jQuery плагин для анимации открытия изображений;
- [TinyMCE](#) – визуальный редактор;
- [Image Manager](#) – плагин для TinyMCE, обеспечивающий загрузку файлов.

# V. Архитектура diafan.CMS

## Файловая структура

**adm** – административная часть сайта;

**adm/htmleditor** – визуальный редактор;

**adm/img** – изображения, используемые в шаблоне административной части;

**adm/includes** – файлы, реализующие прикладной уровень административной части;

**adm/themes** – шаблоны административной части;

**cache** – файлы кеша, используемые для ускорения выдачи страницы;

**css** – файлы стилей сайта;

**img** – изображения, используемые в шаблоне сайта;

**includes** – файлы, реализующие прикладной уровень;

**installation** – установка diafan.CMS;

**js** – JavaScript сценарии, используемые на сайте;

**language** – локализация сайта;

**modules** – модули сайта (пользовательская и административная части);

**modules/название\_модуля** – папка конкретного модуля;

**modules/название\_модуля/admin** – административная часть модуля;

**plugins** – функционал сторонних разработчиков;

**themes** – шаблоны сайта;

**userfiles** – файлы загружаемые пользователем.

# Пользовательская часть сайта

## Инициализация системы

Все запросы к системе попадают в первую очередь на файл *index.php*. Файл выполняет следующие функции:

- подключение всех «жизненно важных» файлов;
- перенаправление на установку системы;
- перенаправление на административную часть;
- определение путей сайта;
- инициация основного объекта системы.

Основной объект системы – экземпляр класса *Init*, описанный в файле *includes/init.php*.

При инициации основного объекта системы происходит:

- старт сессии;
- определение пользователя;
- определение запрошенной языковой версии сайта;
- определение запрошенной страницы сайта;
- подключение модуля;
- подключение шаблона сайта.

Экземпляр класса *Init* передается во все другие объекты и доступен из любого места системы в виде переменной `$this->diafan`.

Таким образом в любом месте системы можно получить глобальные данные:

- данные о текущей странице;
- переданные в URL переменные;
- общие функции;
- объекты классов-подключений.

### Данные о текущей странице

В область глобальной видимости попадают следующие данные о текущей странице и данные, сформированные модулей:

- `cid` – номер текущей страницы;
- `rewrite` – ЧПУ текущей страницы сайта;
- `rewrite_module` – ЧПУ текущей страницы, сгенерированный модулем;
- `name` – название текущей страницы;
- `title_meta` – заголовок текущей страницы для тега `title`;
- `titlemodule` – заголовок страницы, сформированный прикрепленным модулем;
- `edit_titlemodule` – данные для редактирования заголовка модуля;
- `titlemodule_meta` – заголовок текущей страницы для тега `title`, сформированный прикрепленным модулем;
- `keywords` – ключевые слова текущей страницы;

- descr – описание текущей страницы;
- theme – файл шаблона текущей страницы;
- text – контент текущей страницы;
- hide\_previous\_next – спрятать ссылки на предыдущую, последующую страницы
- parent\_id – номер страницы родителя;
- actm – показывать текущую страницу в меню;
- title\_no\_show – не показывать заголовок h1 текущей страницы;
- timeedit – время редактирования текущей страницы;
- module – модуль, прикрепленный к текущей странице;
- path – часть навигации «Хлебные крошки», сформированная прикрепленным модулем.

### Переданные в URL переменные

Список всех доступных переменных задан в массиве `$this->diafan->rewrite_variable_names_all`.

- page – номер страницы;
- show – номер элемента;
- add – добавить элемент;
- edit – номер элемента для редактирования;
- param – номер параметра;
- comments – номер страницы с комментариями;
- cat – номер категории;
- year – год;
- month – месяц;
- step – шаг;
- sort – сортировка.

Например, `http://site.ru/shop/cat1/page2/`. Переданы переменных `$cat = 1` (номер категории) и `$page = 2` (номер страницы). В файле модуля эти данные можно получить через переменные: `$this->diafan->cat`, `$this->diafan->page`.

### Общие функции

Класс *Init* наследует [абстрактный класс Core](#) (`includes/core.php`). Все функции этого класса можно вызвать в контроллере и модели модуля в виде:

```
$this->diafan->название_функции();
```

### Объекты классов-подключений

В основном объекте иницируются классы:

- [пользователи](#) (класс `User`, файл `includes/user.php`, вызывается `$this->diafan->_user`);
- [кеширование](#) (класс `Cache`, файл `includes/cache.php`, вызывается `$this->diafan->_cache`);
- [маршрутизатор](#) (класс `Route`, файл `includes/route.php`, вызывается `$this->diafan->_route`);
- [подключение шаблонов](#) (класс `Template`, файл `includes/template.php`, вызывается `$this->diafan->_tpl`).

Кроме того, через основной объект можно обратиться к любому модулю-подключению. Если в модуле есть файл `modules/модуль/модуль.inc.php`, где описан

класс *Модуль\_inc*, то при первом обращении к переменной `$this->diafan->_модуль`, создастся экземпляр класса.

## Структура модуля

diafan.CMS основана на [MVC](#) паттерне. Модуль, прикрепленный к странице должен обязательно иметь 3 файла:

- [контроллер](#) – файл `modules/модуль/модуль.php` класс `Модуль`;
- [модуль](#) – файл `modules/модуль/модуль.model.php` класс `Модуль_model`;
- [шаблон](#) – файл `modules/модуль/модуль.view.php` класс `Модуль_view`.

Если в модуле обрабатываются данных из формы, отправленные с помощью Ajax, то добавляется файл:

- [обработка](#) – файл `modules/модуль/модуль.ajax.php` класс `Модуль_ajax`.

Если модуль имеет часть «Подключение», то есть используется в других модулях, то добавляется еще один файл:

- [подключение](#) – файл `modules/модуль/модуль.inc.php` класс `Модуль_inc`.

Если модуль устанавливается опционально, то должен существовать файл инсталляции:

- [инсталляция](#) – файл `modules/модуль/модуль.install.php`.

Если страница модуля должна существовать независимо от структуры сайта, то можно обратиться к файлу модуля напрямую. Для этого файл должен иметь название `modules/модуль/модуль.функция.php`. Обратится к нему можно так: `http://site.ru/модуль/функция/`.

- [прямое обращение](#) – файл `modules/модуль/модуль.функция.php`.

Модуль может [взаимодействовать с другими модулями](#). Для этого нужно подготовить файлы в формате, предусмотренном сторонними модулями.

## Контроллер

Контроллер модуля – класс *Модуль* в файле *modules/модуль/модуль.php*.

В классе описаны следующие функции:

- `init` – инициализация модуля;
- `show_module` – вывод контента модуля;
- шаблонные функции.

**Функция инициализации модуля** вызывается, если модуль подключен к текущей странице. В ней происходит:

1. Определение переменных, передаваемых в URL, которые используются в модуле.

Пример:

```
$this->rewrite_variable_names = array('page', 'show', 'cat',  
'sort', 'param');  
$this->diafan->rewrite_variable_names = $this->  
>rewrite_variable_names;
```

2. Подключение модели и выбор функции модели модуля в зависимости от переданных параметров.

Пример:

```
// создаем экземпляр объекта модели модуля «Магазин»  
$model = new Shop_model($this->diafan);  
// если в URL передана переменная show, вызываем функцию id в  
модели – показываем страницу товара  
if($this->diafan->show)  
{  
    $this->result = $model->id();  
}  
// иначе вызываем функцию list_ в модели – выводим список товаров  
else  
{  
    $this->result = $model->list_();  
}
```

3. Определение глобальных переменных.

Глобальные переменные модуля – заголовок страницы, навигация «Хлебные крошки», время редактирования страницы, ключевые слова, описание страницы, спрятать ссылки на предыдущую, последующую страницы. Подробнее смотрите в [разделе «Инициализация системы»](#).

Пример: `$this->get_global_variables();`

В примере используется стандартная функция из класса *Controller*. Вместо этого можно использовать свой алгоритм.

**Функция вывода контента модуля** вызывается из общей [шаблонной функции `show\_body`](#). Её основное назначение – в зависимости от переданных параметров

вызвать функцию в шаблоне модуля.

Функции шаблона вызываются с помощью подключения шаблонов.

Пример:

```
// если в URL передана переменная show, вызываем функцию id в шаблоне – показываем страницу товара
if($this->diafan->show)
{
    $this->diafan->_tpl->get('id', 'shop', $this->result);
}
// иначе вызываем функцию list_ в шаблоне – выводим список товаров

else
{
    $this->diafan->_tpl->get('list_', 'shop', $this->result);
}
```

Данные, сгенерированные в модели передаем в шаблон с помощью свойства класса `$result`.

**Шаблонные функции модуля** – определяются условия, при которых функция не отображается, фильтруются полученные атрибуты, подключаются соответствующие функции в модели и представлении модуля. Подробнее смотрите в [разделе «Как добавить шаблонный тег»](#).

## Класс Controller

Контроллер модуля наследует класс *Controller* – это каркас для всех страниц.

### Свойства

array **\$rewrite\_variable\_names**

- var: переменные, передаваемые в URL страницы
- access: public

array **\$result**

- var: сгенерированные в модели данные, передаваемые в шаблон
- access: protected

object **\$diafan**

- var: основной объект системы
- access: protected

### Методы

void **init ()** – инициализация модуля.

- access: public

void **show\_module ()** – выводит шаблон модуля.

- access: public

boolean true **get\_global\_variables ()** – определяет свойства страницы, если они заданы в модуле.

- access: protected

array **get\_attributes (array \$attributes)** – задает неопределенным атрибутам шаблонного тега значение по умолчанию.

- access: protected
- array \$attributes: массив определенных атрибутов

## Модель

Модель модуля – класс *Модуль\_model* в файле *modules/модуль/модуль.model.php*.

Содержание файла может быть различным в зависимости от заложенного в модуле функционала. Как правило, класс модели описывает следующие функции:

- `list_()` – формирует список элементов;
- `list_category()` – формирует список элементов в категории;
- `first_page()` – первая страница модуля;
- `id()` – формирует данные для страницы элемента;
- `show_block()` – формирует данные для шаблонной функции.

Функции отдадут данные в виде массива, который затем передается в шаблон.

Для ускорения генерирования страницы данные кэшируются. Подробнее о кэшировании в diafan.CMS смотрите [раздел «Кэширование»](#).

## Класс Model

Модуль модуля наследует класс *Model*, описанный в файле *includes/model.php*.

### Методы

string **format\_date (integer \$date, [string \$module\_name = "], [integer \$site\_id = 0])** – форматирует дату в соответствии с конфигурацией модуля.

- access: protected
- integer \$date: дата в [формате UNIX](#)
- string \$module\_name: название модуля, по умолчанию модуль, прикрепленный к текущей странице
- integer \$site\_id: номер страницы сайта

array **get\_author (integer \$author)** – получает имя, никнейм и аватар пользователя сайта.

- access: protected
- integer \$author: номер пользователя сайта

string **get\_error ([string \$module = "], [string \$field = "])** – выдает ошибку о неправильном заполнении формы.

- access: protected
- string \$module: название модуля
- string \$field: название поля

array **get\_path(string \$module)** – генерирует данные для навигации «Хлебные крошки».

- access: protected
- string \$module: модуль

## Шаблон

Шаблон модуля – класс *Модуль\_view* в файле *modules/модуль/модуль.view.php*.

Шаблон представляет из себя набор функции, в которых подставляют сгенерированные в модели данные в HTML код.

Содержание файла может быть различным в зависимости от заложенного в модуле функционала. Как правило, класс шаблона описывает следующие функции:

- `list_()` – шаблон списка элементов;
- `list_level_HOMEP()` – шаблон списка элементов в категории уровня (номер);
- `first_page()` – шаблон первой страницы модуля;
- `id()` – страница элементе;
- `show_block()` – шаблонный тег.

В функциях шаблона можно вызывать шаблоны других модулей, а также выводить значения языковых констант.

Пример:

```
// выводим языковую константу в шаблоне модуля «Магазин» в файле
modules/shop/shop.view.php
echo $this->_('PRICE');
```

```
// подключаем шаблон формы авторизации в шаблоне модуля
«Оформление заказа» в файле modules/order/order.view.php
echo $this->get('show_login', 'registration', $result);
```

Шаблон модуля наследует класс *View*, описанный в файле *includes/view.php*.

В diafan.CMS предусмотрены стандартное CSS-форматирование данных. С его помощью можно описать стиль соответствующего элемента, не обращаясь к файлу шаблона модуля.

`.show_all` – ссылка на все элементы модуля

`.previous_next_links`{clear: both; padding-top: 10px; height: 40px;} – ссылки на предыдущую и следующую страницы

`.previous_link`{width: 40%; float: left;} – ссылка на предыдущую страницу

`.next_link`{width: 40%; float: right; text-align: right;} – ссылка на следующую страницу

`.inptext` – поля input, textarea в форме

`.infofield` – подписи к полю в форме

`.infoform` – заголовок к группе полей

`.inpnum` – цифровые поля

`.inpfile` – поля для добавления файлов

`.inpselect` – поля «Список»

`.inpcheckbox` – поля «Чекбокс»

`.inpdate` – поля «Дата»

`.button` – кнопка Submit

`.errors{color: red;}` – отображает ошибки пользователей при Ajax-запросе; слой становится видимым, если ошибки передаются

## Обработка Ajax-запросов

Все данные из формы обрабатываются классом *Модуль\_ajax*, описанным в файле *modules/модуль/модуль.ajax.php*.

В файле должна быть описана основная функция – *ajax\_request()*, вызывающая при передаче данных модулю. Функция подключается всегда, когда передана переменная `$_POST["module"] = 'модуль';`

То есть данные модулю могут быть переданы даже если он не подключен ни к одной странице сайта. Уже в функции *ajax\_request()* проверяется соответствие прикрепленного к странице модуля и вызванного модуля, если это необходимо.

Функция *ajax\_request()* выдает ответ *true* или *false*, означающий приняты ли данные и стоит ли прерывать загрузку страницы.

Структура класса может быть любой для того, чтобы реализовывать предусмотренный функционал.

Как правило, в классе *Модуль\_ajax* описана еще и функция *send\_errors()*. Она выдает результаты обработки в формате JSON обратно или, если данные переданы не с помощью Ajax, делает редирект на сайт, передавая результаты обработки в виде GET-переменных.

Класс *Модуль\_ajax* часто наследует класс *Model*, если нужно использовать при обработки данных функции каркаса модели.

## Подключение

Для того, чтобы использовать модуль в других модулях в файле *modules/модуль/модуль.inc.php* описывается класс *Модуль\_inc*.

Обращение к подключению модуля происходит через переменную `$this->diafan->_модуль`. При первом обращении к переменной, создается экземпляр класса *Модуль\_inc*.

Класс подключения наследует класс *Model*, описанный в файле *includes/model.php*. Его описание смотрите в [разделе «Модель»](#).

## Инсталляция

Для того, чтобы модуль мог быть установлен опционально создается файл инсталляции модуля. Это файл `modules/модуль/модуль.install.php`.

В файле определены следующие переменные и функции:

- `$title` – название модуля;
- `$sql` – массив SQL запросов для установки модуля;
- `$example` – массив SQL запросов для установки примеров заполнения модуля;
- `module_basic_shop()` – выполняет действия для основной установки модуля;
- `module_example_shop()` – выполняет действия для установки примеров заполнения модуля;
- `module_basic_uninstall_comments()` – деинсталлирует модуль.

Если у полей таблиц модуля предусмотрено различное содержание для различных языковых версий, то в SQL запросе поле обрабатывается функцией `multilang()`. Данная функция повторяет код для каждой языковой версии при этом заменяя код LANG на номер языка. Если функции переданы несколько аргументов, то для первого языка выводится первый аргумент, для второго – второй и т. д. Если число языков превышает количество аргументов, переданных функции, то для последующих языков сайта берется последний аргумент.

Пример:

В модуле «Магазин» название товаров переводится. Сайт переводится на 4 языка: русский, английский, немецкий, французский.

`// запрос на создание таблицы товаров, структура таблицы дана для примера упрощенная`

```
$sql[] = "CREATE TABLE {shop} (  
  `id` INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,  
  ".multilang("`nameLANG` TEXT NOT NULL DEFAULT '',")."  
  PRIMARY KEY (`id`)  
) CHARACTER SET utf8 COLLATE utf8_general_ci;";
```

`// запрос на создание примера заполнения таблицы товаров`

```
$example[] = "INSERT INTO {shop} (id ".multilang(", nameLANG")."  
VALUES (1".multilang(", 'Коршун'", ", 'Vulture'", ", ''").");";
```

В результате сформируются следующие SQL запросы к базе данных:

```
CREATE TABLE diafan_shop (  
  `id` INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `name` TEXT NOT NULL DEFAULT '',  
  `name2` TEXT NOT NULL DEFAULT '',  
  `name3` TEXT NOT NULL DEFAULT '',  
  `name4` TEXT NOT NULL DEFAULT '',  
  PRIMARY KEY (`id`)  
) CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
INSERT INTO diafan_shop (id, name, name2, name3, name4) VALUES  
( 1, 'Коршун', 'Vulture', '', '' );
```

Файл инсталляции используется при установки diafan.CMS, при установки / удалении модуля из административной части, при восстановлении базы данных из административной части.

## Прямое обращение к модулю

Для работы вне структуры сайта описывается файл `modules/модуль/модуль.функция.php`. Этот файл можно вызвать напрямую так: `http://site.ru/модуль/функция/`.

Прямое обращение к модулю может понадобиться в следствии следующих причин:

1. Страница модуля должна существовать независимо от того, создана ли соответствующая страница в структуре сайта. Тем самым нужно обезопасить функциональность от случайного удаления страницы.
2. Страница модуля имеет служебный характер и не должна отображаться на карте сайта, в файле `sitemap.xml`, в меню и пр.
3. При работе с частью модуля нужно избавиться от лишних запросов на определение страницы.

Файл `modules/модуль/модуль.функция.php` подключается в классе инициализации системы. Поэтому основной класс системы доступен через переменную `$this`.

Через эту переменную можно получить доступ к классам пользователей, кэширования, маршрутизатора и подключения шаблонов, а также к подключениям модулей.

Кроме того перед подключением файлов происходит старт сессии, определяется запрошенная языковая версия сайта, временная зона сайта и пользователь. Все эти данные можно использовать в файле модуля.

Если запрошен адрес `http://site.ru/модуль/функция/какие-то_данные/`, то данные, переданные в URL будут доступны в модуле в виде:

```
$_GET["rewrite"] = 'какие-то_данные';
```

Примеры реализации прямого обращения к модулю в diafan.CMS:

- `modules/captcha/captcha.get.php` – `http://site.ru/captcha/get/` – генерирует изображение защитного кода;
- `modules/news/news.rss.php` – `http://site.ru/news/rss/` – генерирует RSS-ленту новостей;
- `modules/order/oder.payment.php` – `http://site.ru/order/payment/` – работа с платежными системами;
- `modules/useradmin/useradmin.edit.php` – `http://site.ru/useradmin/edit/` – форма редактирования данных из пользовательской части;
- `modules/attachments/attachments.get.php` – `http://site.ru/attachments/get/` – отдает прикрепленный файл.

## Взаимодействие с другими модулями

Некоторые модули должны учитывать особенности других модулей в своей работе. Эти особенности должны быть описаны в файлах других модулей в нужном формате.

Например, формирование карты сайта (файл *sitemap.xml*) должно учитывать особенности построения ссылок во всех модулях. Поэтому в diafan.CMS есть файлы *modules/news/news.sitemap.php*, *modules/shop/shop.sitemap.php*, *modules/tags/tags.sitemap.php* и др.

Файл описывает класс *Модуль\_sitemap* и функцию *get()* в нем.

`boolean true get (integer $site_id, integer $timeedit)` – генерирует карту модуля.

- `access: public`
- `integer $site_id`: номер страницы сайта
- `integer $timeedit`: время редактирования страницы

Формат файла заложен в модуле «Карта сайта».

В текущей версии diafan.CMS 4.4 в пользовательской части заложена только одна связь модулей, заложенная модулем *sitemap*.

## Как добавить шаблонный тег

Шаблонные теги позволяют подключать исполняемый код в определенное место сайта.

Исполняемый код может быть общего характера или относиться к модулю.

Если нужно добавить код **общего характера**, то создаем общий шаблонный тег. Для этого добавляем функцию в класс *Theme\_functions* описанный в файле *includes/function.php*. Новая функция должна иметь доступ *public*. Теперь шаблонный тег с названием функции можно использовать.

Функции можно передавать атрибуты тега. Для этого функция должна принимать аргумент – массив атрибутов.

Пример:

```
public function show_text($attributes)
{
    if($attributes["type"] == 1)
    {
        echo "Текст первого типа";
    }
    else
    {
        echo "Текст второго типа";
    }
}
```

тег `<insert name="show_text">` выведет «Текст второго типа»

тег `<insert name="show_text" type="1">` выведет «Текст первого типа»

Если код **связан с модулем**, значит шаблонную функцию нужно добавлять в контроллер модуля. Так как модуль основан на паттерне MVC, шаблонный тег должен иметь свою функцию в модели модуля и в шаблоне, а основная шаблонная функция, определенная в контроллере.

Пример:

```
// в файле modules/news/news.php определена шаблонная функция show_date
public function show_date($attributes)
{
    include_once(ABSOLUTE_PATH.'modules/news/news.model.php');
    $model = new News_model($this->diafan);
    $result = $model->date($attributes["day"]);
    $this->diafan->_tpl->get('date', 'news', $result);
    return TRUE;
}
//в файле modules/news/news.model.php
public function date($day)
{
    if($day)
    {
```

```
        $date = $day.date(".m.Y");
    }
    else
    {
        $date = date("d.m.Y");
    }
    return $date;
}
//в файле modules/news/news.view.php
public function date($date)
{
    echo '<div class="news_date">'.$date.'</div>';
}
```

## **Подключение модуля**

Подключаемый модуль выбирается из списка модулей при редактировании страницы сайта.

Большинство модулей можно прикрепить к нескольким страницами, при этом контент для различных страниц будет различным. Также для разных страниц можно задать различную конфигурацию модуля.

# Административная часть сайта

## Инициализация системы

Если файл *index.php* определил, что запрошена административная часть, то подключается файл *adm/index.php*. Файл выполняет следующие функции:

- подключение всех «жизненно важных» файлов;
- инициация основного объекта системы.

Основной объект системы – экземпляр класса *Initadmin*, описанный в файле *adm/includes/init.php*.

При инициации основного объекта системы происходит:

- старт сессии;
- определение пользователя;
- определение запрошенной языковой версии для редактирования;
- определение запрошенного модуля;
- подключение модуля;
- подключение шаблона административной части.

Экземпляр класса *Initadmin* передается во все другие объекты и доступен из любого места системы в виде переменной `$this->diafan`.

Таким образом в любом месте системы можно получить глобальные данные:

- данные о текущей странице;
- переданные в URL переменные;
- общие функции;
- объекты классов-подключений.

### Данные о текущей странице

В область глобальной видимости попадают следующие данные о текущей странице и данные, сформированные модулей:

- `cid` – номер текущей страницы;
- `rewrite` – ЧПУ текущей страницы сайта;
- `name` – название текущей страницы;
- `theme` – файл шаблона текущей страницы;
- `text` – контент текущей страницы;
- `parent_id` – номер страницы родителя;
- `module` – модуль, прикрепленный к текущей странице.

### Переданные в URL переменные

Список всех доступных переменных задан в массиве `$this->diafan->rewrite_variable_names`.

- `page` – номер страницы;
- `addnew` – значение 1, добавление элемента;
- `edit` – номер элемента для редактирования;
- `save` – значение 1, сохранение нового элемента;
- `save` – номер элемента для сохранения;
- `cat` – номер категории;
- `site` – номер страницы сайта;
- `parent` – номер родителя;
- `error` – код ошибки, уведомления.

Пример:

Страница редактирования товара – <http://www.site.ru/admin/shop/edit13/>

### Общие функции

Класс *Initadmin* наследует [абстрактный класс Core](#) (*includes/core.php*). Все функции этого класса можно вызвать в виде:

```
$this->diafan->название_функции();
```

### Объекты классов-подключений

В основном объекте иницируются классы:

- [пользователи](#) (класс *User*, файл *includes/user.php*, вызывается `$this->diafan->_user`);
- [маршрутизатор](#) (класс *Route*, файл *includes/route.php*, вызывается `$this->diafan->_route`);
- [подключение шаблонов](#) (класс *Template*, файл *includes/template.php*, вызывается `$this->diafan->_tpl`).

Кроме того, через основной объект можно обратиться к любому модулю-подключению. Если в модуле есть файл *modules/модуль/модуль.inc.php*, где описан класс *Модуль\_inc*, то при первом обращении к переменной `$this->diafan->_модуль`, создастся экземпляр класса.

## Действия

Логика основных действий для редактирования данных заложена в ядре системы. В diafan.CMS предусмотрены следующие действия:

- формирование списка – файл `adm/includes/show.php`;
- редактирование, добавление – файл `adm/includes/edit.php`;
- сохранение – файл `adm/includes/save.php`;
- удаление, удаление в корзину – файл `adm/includes/del.php`;
- блокирование, разблокирование – файл `adm/includes/act.php`;
- сортировка – файл `adm/includes/order.php`;
- обработка Ajax запросов – файл `adm/includes/ajax.php`.

Система определяет какой из файлов подключить по данным переданным в URL. В файле модуля можно использовать стандартный функционал действия или расширить и даже переопределить на свои действия.

Рассмотрим как это можно сделать.

### Формирование списка

При открытии модуля вызывается функция `show_module()`, определенная в каждом классе администрирования модуля. В этой функции можно заложить свое представление страницы администрирования модуля, можно вызвать стандартную функцию, формирующую список элементов таблицы модуля:

```
$this->list_row();
```

Перед вызовом функции можно определить переменные класса `$this->name` и `$this->where`:

- `name` – название переменной, используемой в качестве ссылки на элемент;
- `where` – условие для SQL запроса.

Настройки вывода списка закладываются в переменную `$this->config_module`, определенную в файле модуля.

В список можно добавить свои колонки. Для этого определяется массив `$this->config_other_row` с названиями дополнительных полей для отображения.

Для каждого из полей можно определить пользовательскую функцию, которая будет выводить ячейку таблицы. Формат пользовательской функции:

`other_row_переменная($row)`, где `$row` – массив данных об элементе текущей строки таблицы.

### Редактирование, добавление

Если в URL страницы передана переменная `edit` или `addnew`, то подключается функция `edit_module()` из файла `adm/includes/edit.php`. Функция генерирует форму редактирования на основе массива `$this->variable_table`, определенного в классе модуля.

Функцию можно переопределить в классе модуля.

Также дополнить редактирование полей своим представлением вывода. Для этого в классе модуля определяется функция `edit_переменная_module()` или

`edit_переменная_module_config()` для файла настроек.

Пользовательская функция вызывается всегда, если она определена, независимо от типа переменной, указанного в массиве `$this->variable_table`. Исключение составляет тип *module*.

Некоторые пользовательские функции редактирования полей определены в файле `adm/includes/edit.php`. Например, редактирование ЧПУ, сортировки, раздела сайта и пр. Их также можно переопределить в классе модуля.

Некоторые модули, агрегируют информацию об элементах других модулей. И при редактировании последних, нужно чтобы связанная информация также редактировалась.

Например, при редактировании новости нужно редактировать прикрепленные изображения и теги.

Для этого в модулях-подключения нужно создать файл `modules/модуль/admin/модуль.admin.edit.php`. В файле должен быть определен класс `Модуль_admin_edit` со статичной функцией `edit` или `edit_config` для файла настроек.

`string edit (object $th)` – редактирует элементы модуля, связанные с другими элементами.

- access: protected
- object \$th: класс редактирование

`string edit_config (object $th)` – редактирует настройку модуля для другого модуля.

- access: protected
- object \$th: класс редактирование

Затем в основном модуле (в примере это модуль новости) в таблице `$this->variable_table` нужно указать переменную, названную также как модуль-подключение и задать ей тип *module*.

Пример:

```
$this->variable_table['images'] = 'module';
```

## Сохранение

Если в URL страницы передана переменная `save` или `save_new`, то подключается функция `save_module()` из файла `adm/includes/save.php`. Функция сохраняет данные из формы на основе массива `$this->variable_table`, определенного в классе модуля.

Функцию можно переопределить в классе модуля.

Также дополнить сохранение полей своей обработкой. Для этого в классе модуля определяется функция `save_переменная_module()` или `save_переменная_module_config()` для файла настроек.

Функция определяет:

```
$this->query[] = "название_поля='%f'";  
$this->value[] = "значение_поля";
```

Для определения маски обратитесь к [разделу «Базы данных»](#).

Пользовательская функция вызывается всегда, если она определена, независимо от

типа переменной, указанного в массиве `$this->variable_table`. Исключение составляет тип *module*.

Некоторые пользовательские функции редактирования полей определены в файле *adm/includes/save.php*. Например, время редактирования, сортировка, ЧПУ и пр. Их также можно переопределить в классе модуля.

Некоторые пользовательские функции редактирования полей определены в файле *adm/includes/edit.php*. Например, редактирование ЧПУ, сортировки, раздела сайта и пр. Их также можно переопределить в классе модуля.

Некоторые модули, агрегируют информацию об элементах других модулей. И при сохранении последних, нужно чтобы связанная информация также сохранялась.

Например, при сохранении новости нужно сохранять информацию о ссылках в меню на новость.

Для этого в модулях-подключения нужно создать файл *modules/модуль/admin/модуль.admin.save.php*. В файле должен быть определен класс *Модуль\_admin\_save* со статичной функцией *save* или *save\_config* для файла настроек.

string **save (object \$th)** – сохраняет элементы модуля, связанные с другими элементами.

- access: protected
- object \$th: класс редактирование

string **save\_config (object \$th)** – сохраняет настройку модуля для другого модуля.

- access: protected
- object \$th: класс редактирование

Затем в основном модуле (в примере это модуль новости) в таблице `$this->variable_table` нужно указать переменную, названную также как модуль-подключение и задать ей тип *module*.

Пример:

```
$this->variable_table['menu'] = 'module';
```

### Удаление, удаление в корзину

При удалении элемента или нескольких элементов модуля подключается функция *del\_module()* из файла *adm/includes/del.php*.

Эту функцию можно переопределить в классе модуля.

Кроме того, если в классе модуля определена функция `delete($del_id, $trash_id)`, то помимо заложенных в ядре действий она выполняется для каждого удаляемого элемента.

Надо учитывать, что если элементы удаляются в корзину, то сопутствующие им элементы тоже должны быть удалены в корзину. Поэтому вместо прямого SQL запроса на удаление лучше использовать функцию *del\_or\_trash\_where*, определенную в файле *adm/includes/del.php*.

Пример:

При удалении товара нужно удалять его цены. Для этого в классе *Shop\_admin* файла *modules/shop/shop.admin.php* определяем функцию:

```
protected function delete($del_id, $trash_id)
{
    $this->del_or_trash_where("shop_price", "good_id=".$del_id, $trash_id);
    return TRUE;
}
```

Некоторые модули, агрегируют информацию об элементах других модулей. И при удалении последних, нужно чтобы связанная информация также удалялась.

Например, при удалении товара нужно удалять ссылку в меню на него и комментарии о товаре.

Для этого в модулях-подключения нужно создать файл *modules/модуль/admin/модуль.admin.del.php*. В файле должен быть определен класс *Модуль\_admin\_del* со статичной функцией *delete*.

**string delete (object \$th, string \$module\_name, integer \$element\_id, [integer \$trash\_id = 0], [boolean \$is\_category = FALSE])** – удаляет элементы модуля, связанные с другими элементами.

- access: protected
- object \$th: класс удаление
- string \$module\_name: название модуля, элемент которого удаляется
- integer \$element\_id: номер удаляемого элемента
- integer \$trash\_id: номер записи в корзине, с которой связано удаление
- integer \$is\_category: удаляемый элемент - категория

### **Блокирование, разблокирование**

При удалении элемента или нескольких элементов модуля подключается функция *act\_module()* из файла *adm/includes/act.php*.

Эту функцию можно переопределить в классе модуля.

Некоторые модули, агрегируют информацию об элементах других модулей. И при блокировании последних, нужно чтобы связанная информация также блокировалась.

Например, при блокировании страницы сайта нужно блокировать пункт меню, связанный с ней.

Для этого в модулях-подключения нужно создать файл *modules/модуль/admin/модуль.admin.act.php*. В файле должен быть определен класс *Модуль\_admin\_act* со статичной функцией *act*.

**string delete (object \$th, string \$stable, array \$element\_ids, integer \$act)** – блокирует / разблокирует элементы модуля, связанные с другими элементами.

- access: protected
- object \$th: класс блокирование / разблокирование
- string string \$stable таблица
- array \$element\_ids: номера блокируемого / разблокируемого элементов
- integer \$act: блокировать/разблокировать

## Сортировка

При сортировке элементов подключается функция `sort_module()` из файла `adm/includes/sort.php`.

Эту функцию можно переопределить в классе модуля.

## Обработка Ajax запросов

Некоторые запросы требуют специальной обработки. Например, подгрузка списка дерева сайта при редактировании родителя страницы. Эта обработка происходит в файле `adm/includes/ajax.php`.

Если нужно производить обработку Ajax запросов в модуле, то создается файл `modules/модуль/admin/модуль.admin.ajax.php`. В файле описывается класс `Модуль_admin_ajax`. Обработка данных производится в конструкторе модуля. Для того, чтобы система подключила файл, нужно отправить переменную `$_POST["module"] = 'модуль';`

# Структура модуля

## Файл администрирования

Файл администрирования модуля – это файл `modules/модуль/admin/модуль.admin.php`, в котором определен класс `Модуль_admin`.

Если модуль имеет несколько частей, то административная часть модуля может содержать и другие файлы `modules/модуль/admin/модуль.admin.часть.php`, где описан класс `Модуль_admin_часть`.

Класс администрирования модуля должен содержать как минимум две функции:

- `__construct` – определяет переменные-настройки;
- `show_module` – выводит главную страницу администрирования модуля.

В конструкторе модуля могут быть определены переменные-настройки:

### 1. Название таблицы:

`$this->table`

---

### 2. Переменные для редактирования:

`$this->variable_table` – массив с названием переменных (название столбцов таблицы) и типов переменных. Определены следующие типы:

- `text` – строка;
- `password` – пароль;
- `numtext` – число;
- `floattext` – число с плавающей точкой;
- `checkbox` – переменная с двумя значениями (1,0);
- `select` – список (для использования типа необходимо определить массив `$this->select["переменная"]` или `$this->select_arr["переменная"]`);
- `date` – дата (значение выводится в формате dd.mm.yyyy, сохраняется в [формате UNIX](#));
- `datetime` – дата и время (значение выводится в формате dd.mm.yyyy hh:mm, сохраняется в [формате UNIX](#));
- `textarea` – текстовое поле типа `textarea`;
- `editor` – текстовое поле с визуальным редактором;
- `none` – при редактировании не выводится, но значение сохраняется (можно использовать с пользовательской функцией);
- `function` – обозначаются поля, к которым подключены пользовательские функции (подключаются также к любому другому типу кроме `module`);
- `module` – обозначаются поля, к которым подключены другие модули;
- `hr` – выводит горизонтальную полосу на странице редактирования, при сохранении не учитывается;
- `disabled` – текстовое поле, которое невозможно редактировать.

Формат:

```
$this->variable_table = array(
    'переменная' => 'тип',
    ...
);
```

Массив можно делить на вкладки для этого надо определить переменную:

```
$this->config_module["tab_card"] = 1;
```

Тогда формат массива `$this->variable_table` должен быть следующим:

```
$this->variable_table = array(
    "вкладка" => array(
        "переменная" => "тип",
        ...
    ),
    ...
);
```

Названия переменных для отображения пользователю должны быть определены в файле локализации (папка *language*) в виде констант в формате:

`__ADMIN_МОДУЛЬ_ПЕРЕМЕННАЯ` или `__ADMIN_ПЕРЕМЕННАЯ` (для использования в нескольких модулях).

Для конфигурации модуля:

`__ADMIN_МОДУЛЬ_CONFIG_ПЕРЕМЕННАЯ` или `__ADMIN_ПЕРЕМЕННАЯ_CONFIG` (для использования в нескольких модулях).

Можно использовать подсказки:

`__ADMIN_МОДУЛЬ_ПЕРЕМЕННАЯ_HELP` или `__ADMIN_ПЕРЕМЕННАЯ_HELP` (для использования в нескольких модулях).

Для конфигурации модуля:

`__ADMIN_МОДУЛЬ_CONFIG_ПЕРЕМЕННАЯ_HELP` или `__ADMIN_ПЕРЕМЕННАЯ_CONFIG_HELP` (для использования в нескольких модулях).

Для полей с типом editor (визуальный редактор) можно настроить высоту поля с помощью переменной `$this->editor_height`.

Формат: `$this->editor_height = array('поле' => 'высота в px');`

---

### 3. Справочники:

Если в массиве `$this->variable_table` определены переменные с типом select, то для них необходимо определить массив `$this->select["переменная"]` или `$this->select_arr["переменная"]`. Формат:

1. Использование значений из таблицы:

```
$this->select["название_переменной"] = array(
    "название таблицы",
    "название поля значения",
    "название поля для отображения",
    "значение списка при добавлении элемента",
    "Отображение для пустого поля (если значение не задано
, пустого элемента в списке не существует",
```

```
        "условие при sql запросе",
        "поле для организации вложенности (если вложенности не
существует, значение пустое)"
    );
```

2. Массив содержит значения для списка:

```
$this->select_arr["название_переменной"] = array(
    "значение" => "отображение",
    ...
);
```

---

#### 4. Переводимые поля:

`$this->language_names` — массив переводимых полей

Формат:

```
$this->language_names = array(
    "название_переменной",
    ...
);
```

---

#### 5. Настройки отображения:

`$this->config_module` – настройка отображения списка значений:

- `act` – отображение кнопки быстрой публикации (при наличии поля `act`);
- `del` – отображение кнопки удаления;
- `date` – отображение даты в формате `dd.mm.yyyy` (при наличии поля `data`), сортировка по дате;
- `datetime` – отображение даты и времени в формате `dd.mm.yyyy hh:ii` (при наличии поля `data`), сортировка по дате;
- `prior` – приоритетная сортировка по полю `prior`;
- `parent` – использовать вложенность элементов (при наличии поля `parent_id`);
- `order` – возможность сортировки элементов (при наличии поля `sort`);
- `view` – просмотр страницы на сайте;
- `menu` – показывать страницу в меню;
- `porage` – показывать все элементы (без разбивки на страницы);
- `image` – показывать изображение элемента;
- `element` – использование категорий в модуле (отдельная таблица `{таблица_category}`); часть с элементами;
- `element_multiple` – элемент может быть прикреплен к нескольким категориям;
- `element_site` – использование разделов (если модуль может быть подключен к нескольким страницам);
- `category` – использование категорий в модуле (отдельная таблица `{таблица_category}`); часть с категориями;
- `category_flat` – категории элементов модуля не содержат вложенностей;
- `category_rel` – работают вместе с таблицей `{модуль_category_rel}`;
- `link_to_element` – основная ссылка ведет к списку элементов, принадлежащих категории;

- trash – использование корзины для удаления элементов;
- no\_out – не отображать кнопку «Сохранить и выйти»;
- tab\_card – использование вкладок;
- search\_name – форма поиска по названию;
- config – файл настроек модуля.

Формат:

```
$this->config_module = array(
    'переменная' => 1,
    ...
);
```

---

## 6. Дополнительные поля для формирования списка элементов:

`$this->config_other_row` – массив названий дополнительных полей для отображения.

Формат:

```
$this->config_other_row = array(
    'переменная',
    ...
);
```

Для отображения дополнительных полей возможно определение пользовательских функций формата: `other_row_переменная($row)`, где `$row` – массив значений для текущего элемента. Пример использования модули `modules/rating/rating.admin.php`, `modules/users/users.admin.php`.

---

## 7. Дополнительные поля при редактировании элемента:

Для определения некоторых переменных как дополнительных при редактировании элемента используется массив `$this->variable_other`.

Формат:

```
$this->variable_other = array(
    'переменная' => 1,
    ...
);
```

---

Если отображение одного поля зависит от значения другого, то используются массив `$this->show_tr_click_checkbox`.

Формат:

```
$this->show_tr_click_checkbox = array(
    "название_главной_переменной" => array(
        'название_зависимой_переменной',
        ...
    ),
    ...
);
```

Главная переменная должна иметь тип checkbox. Пример использования файл *modules/faq/faq.admin.config.php*.

## Настройки модуля

Настройки модуля – это интерфейс для конфигурирования модуля.

Настройки всех модулей хранятся в таблице `{config}`.

Для использования настроек модуля как в пользовательской части, так и в административной существует функция [configmodules](#).

Пример:

```
if($this->diafan->configmodules("переменная", "модуль") ==
"значение")
{
    //выполнение действия
}
```

## Административная часть

Файл настроек модуля – `modules/модуль/admin/модуль.admin.config.php`. Файл строится по тому же принципу, что и любой файл административной части модуля.

### Разделы сайта

Если модуль прикреплен к нескольким страницам сайта, то для каждой страницы можно задать свои настройки для модуля.

Это обеспечивает следующая запись в классе настроек модуля:

```
$this->config_module = array(
    'element_site' => 1,
    ...
);
```

### Переводимые настройки

Некоторые поля имеют разные значения для разных языковых интерфейсов.

Переводимые значения задаются в массиве

```
$this->language_names = array('название_поля', ...);
```

Пример: параметр `'add_message'` - «Сообщение после отправки» в конфигурации модуля «Обратная связь» должен иметь разные значения для английской и русской версии сайта. Поэтому в файле `modules/feedback/admin/feedback.admin.config.php` делается следующая запись:

```
$this->language_names = array(..., 'add_message', ...);
```

В пользовательской части (в файле `modules/feedback/feedback.ajax.php`) обращаемся к переменной обычным способом:

```
$this->diafan->configmodules("add_message");
```

## Подключение модуля

Для того, чтобы модуль можно было редактировать, необходимо создать страницу в административной части, ссылающуюся на него.

Для этого существует специальный интерфейс, доступный по адресу:  
<http://site.ru/admin/adminsite/>.

Важно знать, что для подключения модуля, нужно, чтобы псевдоссылка страницы совпадала с названием модуля. Если подключается одна из неосновных частей модуля (файл `modules/модуль/admin/модуль.admin.часть.php`), то псевдоссылка должна быть `модуль/часть`.

Новые страницы заносятся в таблицы `{adminsite}` и, если это страница второго уровня, `{adminsite_parents}`.

## VI. Справочник тегов

### В

#### **show\_basket module="basket"**

Выводит информацию о заказанных товарах, т. н. корзину.

Пример: `<insert name="show_basket" module="basket">`

выведет информацию о корзине

Шаблон: функция *show\_basket* в файле *modules/basket/basket.view.php*.

CSS-форматирование:

`.basket_block`{} – шаблонная функция *show\_basket*

`.basket_block .basket_count`{} – количество купленных товаров

`.basket_block .basket_summ`{} – стоимость товаров

#### **show\_body**

Выводит основной контент страницы: заголовка (если не запрещен его вывод в настройке странице «Не показывать заголовков»), текста страницы и прикрепленного модуля.

Пример: `<insert name="show_body">`

Шаблон: функция *show\_body* в файле *includes/function.php*.

#### **show\_block**

Выводит содержимое страницы сайта, номер которой передан в виде атрибута **id**. Страница должна быть помечена опцией «Блок на сайте» и быть активной.

Пример: `<insert name="show_block" id="7">`

выведет в ячейке таблицы содержимое страницы с `id=7`, редактируемой в модуле «Страницы сайта».

Шаблон: функция *show\_body* в файле *includes/function.php*.

#### **show\_block module="clauses"**

Выводит последние статьи на всех страницах, кроме страницы статей, когда выводится список тех же статей, что и в функции.

Атрибуты:

- **count** – количество выводимых статей (по умолчанию 3);
- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** – категория статей (`id` категории, по умолчанию выводятся все статьи без учета категории), если в конфигурации модуля отмечено «Использовать категории»;

- **images** – количество фотографий, прикрепленных к статьям;
- **sort** – сортировка статей (**rand** – в случайном порядке, **date** – по дате). По умолчанию статьи сортируются по дате;
- **template** – шаблон тега (функция в файле *modules/clauses/clauses.view.php* должна называться *show\_block\_template*; по умолчанию шаблон *show\_block*).

Формат даты задается в конфигурации модуля.

Пример:

```
<insert name="show_block" module="clauses">
```

выведет 3 последние статьи сайта

```
<insert name="show_block" module="clauses" count="5">
```

выведет 5 последних статей сайта

```
<insert name="show_block" module="clauses" count="1" site_id="35">
```

выведет одну последнюю статью сайта, из раздела статей с id=35)

Шаблон: функция *show\_block* в файле *modules/clauses/clauses.view.php*.

CSS-форматирование:

`.clauses_block{}` – блок статей

`.clauses_block h2{}` – заголовок

`.clauses_block .clauses{}` – статья в блоке

`.clauses_block .clauses_name{}` – название статьи

`.clauses_block .clauses_img{}` – изображение статьи

`.clauses_block .clauses_date{}` – дата статьи

`.clauses_block .clauses_anons{}` – анонс статьи

`.clauses_block .show_all{}` – ссылка на все статьи

### **show\_block module="faq"**

Выводит последние вопросы. На странице вопросов, когда выводится список аналогичных вопросов, блок не выводится.

Атрибуты:

- **count** – количество выводимых вопросов (по умолчанию 3);
- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** – категория вопросов (id категории, по умолчанию выводятся все вопросы без учета категории), если в конфигурации модуля отмечено «Использовать категории»;
- **often** – часто задаваемые вопросы (если `often=1`, выводятся вопросы с

пометкой «Часто задаваемый вопрос»);

— **template** – шаблон тега (функция в файле *modules/faq/faq.view.php* должна называться *show\_block\_template*; по умолчанию шаблон *show\_block*).

Формат даты задается в конфигурации модуля.

```
<insert name="show_block" module="faq">
```

выведет 3 последних активных вопроса сайта с ответами

```
<insert name="show_block" module="faq" count="5">
```

выведет в ячейке таблицы 5 последних активных вопросов сайта с ответами

```
<insert name="show_block" module="faq" often="1">
```

выведет часто задаваемые вопросы

Шаблон: функция *show\_block* в файле *modules/faq/faq.view.php*.

CSS-форматирование:

*.faq\_block*{ } – шаблонная функция *show\_block*

*.faq\_block h2*{ } – заголовок

*.faq\_block .faq*{ } – вопрос в блоке

*.faq\_block .faq\_date*{ } – дата

*.faq\_block .faq\_question*{ } – вопрос

*.faq\_block .faq\_answer*{ } – ответ

*.faq\_block .show\_all*{ } – ссылка на все вопросы

### **show\_block module="files"**

Выводит несколько файлов из файлового архива.

Атрибуты:

— **count** – количество выводимых файлов (по умолчанию 3);

— **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;

— **cat\_id** – категория файлов (id категории, по умолчанию выводятся все файлы без учета категории), если в конфигурации модуля отмечено «Использовать категории»;

— **images** – количество фотографий, прикрепленных к файлам;

— **sort** – сортировка файлов (**rand** – в случайном порядке, **date** – по дате). По умолчанию файлы сортируются по дате;

— **template** – шаблон тега (функция в файле *modules/files/files.view.php* должна называться *show\_block\_template*; по умолчанию шаблон *show\_block*).

Пример: `<insert name="show_block" module="files">`

выведет 3 последние файла из файлового архива

Шаблон: функция `show_block` в файле `modules/files/files.view.php`.

CSS-форматирование:

`.files_block{}` – шаблонная функция `show_block`

`.files_block h2{}` – заголовок

`.files_block .files{}` – файл в блоке

`.files_block .files_img{}` – изображение файла

`.files_block .files_name{}` – название файла

`.files_block .files_anons{}` – краткое описание файла

`.files_block .show_all{}` – ссылка на все файлы

### **show\_block module="menu"**

Выводит меню.

Без атрибутов выводит просто ссылки вида `<a href="...">имя пункта меню</a>`, одну за одной.

Атрибуты:

- **id** – идентификатор категории меню;
- **template** – шаблон тега (функция в файле `modules/menu/menu.view.php` должна называться `show_block_template`; по умолчанию значение атрибута `default` шаблон `show_block`). Если атрибут не задан, меню оформляется с помощью атрибутов оформления и выводится с помощью функции в шаблоне `show_menu`.

Атрибуты оформления:

- **tag\_start\_номер** – текст, выводимый перед каждой ссылкой меню (может содержать слово `Increment` – при выводе автоматически заменяемое на арифметическую прогрессию, 1, 2, 3, ... и `Level` – заменяемое на номер уровня меню);
- **tag\_end\_номер** – текст, выводимый после каждой ссылки пункта меню (может содержать слово `Increment` - заменяемое на увеличивающийся номер и `Level` – заменяемое на номер уровня меню);
- **tag\_level\_start\_номер** – текст, выводимый перед уровнем (номер) меню;
- **tag\_level\_end\_номер** – текст, выводимый после уровня (номер) меню;
- **tag\_active\_start\_номер** – текст, выводимый перед активным пунктом меню уровня (может содержать слово `Increment` - заменяемое на увеличивающийся номер и `Level` – заменяемое на номер уровня меню);
- **tag\_active\_end\_номер** – текст, выводимый после активного пункта меню уровня (может содержать слово `Increment` - заменяемое на увеличивающийся номер и `Level` – заменяемое на номер уровня меню);
- **tag\_active\_child\_start\_номер** – текст, выводимый перед пунктом меню уровня (номер) с активным дочерним пунктом (может содержать слово `Increment` - заменяемое на увеличивающийся номер и `Level` – заменяемое на номер уровня меню);

- **tag\_active\_child\_end\_номер** – текст, выводимый после пункта меню уровня (номер) с активным дочерним пунктом (может содержать слово Increment - заменяемое на увеличивающийся номер и Level – заменяемое на номер уровня меню);
- **separator\_номер** – текст, разделяющий пункты меню.

Номер в названии этих атрибутов обозначает номер уровня. Если атрибуты не указаны для текущего уровня, то они наследуются у верхнего уровня.

Пример: `<insert name="show_block" module="menu">`

выведет просто список ссылок основного меню

```
<table>
  <insert name="show_block" module="menu" id="2"
tag_start_1="[tr][td]" tag_end_1="[/td][/tr]">
</table>
```

сформирует таблицу, в отдельной ячейке которой будут элементы списка меню из второй категории

```
<insert name="show_block" module="menu"
tag_start_1="[div class=`menu`]" tag_end_1="[/div]"
tag_start_2="[div class=`submenu`]" tag_end_2="[/div]">
```

Выведет:

```
<div class="menu"><a href="...">0 нас</a></div>
  <div class="submenu"><a href="...">Животные</a></div>
  <div class="submenu"><a href="...">Миссия</a></div>
  <div class="submenu"><a href="...">История</a></div>
<div class="menu"><a href="...">Фотогалерея</a></div>
```

```
<insert name="show_block" module="menu"
tag_start_1="[div class=`menuLevel`]" tag_end_1="[/div]">
```

Выведет:

```
<div class="menu1"><a href="...">0 нас</a></div>
  <div class="menu2"><a href="...">Животные</a></div>
  <div class="menu2"><a href="...">Миссия</a></div>
  <div class="menu2"><a href="...">История</a></div>
<div class="menu1"><a href="...">Фотогалерея</a></div>
```

Шаблон: функция *show\_block*, *show\_level* (второй уровень) – при оформлении шаблоном, *show\_menu* – при оформлении тегами в файле *modules/menu/menu.view.php*.

CSS-форматирование:

`.menu_block{}` – блок меню, если используется шаблон по-умолчанию

`.menu_block h2` – заголовок блока меню  
`.menu_item` – пункт меню  
`.menu_active` – текущий пункт меню  
`.menu_active_child` – пункт меню с текущим дочерним пунктом  
`.menu_level_HOME` – уровень меню HOME  
`.menu_item_HOME` – пункт меню уровня HOME

### **show\_block module="news"**

Выводит последние новости на всех страницах, кроме страницы новостей, когда выводится список тех же новостей, что и в функции.

Атрибуты:

- **count** – количество выводимых новостей (по умолчанию 3);
- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** – категория новостей (id категории, по умолчанию выводятся все новости без учета категории), если в конфигурации модуля отмечено «Использовать категории»;
- **images** – количество фотографий, прикрепленных к новости;
- **template** – шаблон тега (функция в файле `modules/news/news.view.php` должна называться `show_block_template`; по умолчанию шаблон `show_block`).

Формат даты задается в конфигурации модуля.

```
<insert name="show_block" module="news">
```

выведет 3 последние новости сайта

```
<insert name="show_block" module="news" count="5">
```

выведет 5 последних новостей сайта

```
<insert name="show_block" module="news" count="1" site_id="12">
```

выведет последнюю новость, прикрепленную к странице 12

Шаблон: функция `show_block` в файле `modules/news/news.view.php`.

CSS-форматирование:

`.news_block` – шаблонная функция `show_block`  
`.news_block h2` – заголовок  
`.news_block .news` – новость в блоке  
`.news_block .news_name` – название новости  
`.news_block .news_img` – изображение новости

- `.news_block .news_date{}` – дата новости
- `.news_block .news_anons{}` – анонс
- `.news_block .show_all{}` – ссылка на все новости

### **show\_block module="photo"**

Выводит несколько фотографий.

Атрибуты:

- **count** – количество выводимых фотографий (по умолчанию 3);
- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** – альбом фотографий (id альбома, по умолчанию выводятся все фотографии без учета альбома), если в конфигурации модуля отмечено «Использовать альбомы»;
- **sort** – сортировка фотографий (**rand** – в случайном порядке, **date** – по дате). По умолчанию фотографии сортируются по дате;
- **template** – шаблон тега (функция в файле *modules/photo/photo.view.php* должна называться *show\_block\_template*; по умолчанию шаблон *show\_block*).

Пример: `<insert name="show_block" module="photo">`

выведет 3 последние фотографии из фотогалереи

Шаблон: функция *show\_block* в файле *modules/photo/photo.view.php*.

CSS-форматирование:

- `.photo_block{}` – шаблонная функция *show\_block*
- `.photo_block h2{}` – заголовок
- `.photo_block .photo{}` – фотография в блоке
- `.photo_block .photo_name{}` – название фотографии в блоке
- `.photo_block .photo_img{}` – изображение фотографии в блоке
- `.photo_block .photo_size{}` – размер фотографии в блоке
- `.photo_block .photo_anons{}` – краткое описание фотографии в блоке
- `.photo_block .show_all{}` – ссылка на все фотографии

### **show\_block module="shop"**

Выводит несколько товаров из каталога.

Атрибуты:

- **count** – количество выводимых товаров (по умолчанию 3);
- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** – категория товаров (id категории, по умолчанию выводятся все товары без учета категории), если в конфигурации модуля отмечено «Использовать категории»;
- **images** – количество фотографий, прикрепленных к товарам;

- **sort** – сортировка товаров (**rand** – в случайном порядке, **date** – по дате, **price** – по цене). По умолчанию товары сортируются по дате;
- **param** – значения дополнительных характеристик;

Пример:

Товары обладают следующими характеристиками:

- цвет – выпадающий список, номер 3;
- высота – текстовое поле, номер 10;
- наличие аналогов – галочка, номер 16.

Значит значение атрибута `param3=5&param3=6&param10=12м&param16=0` расшифровывается как товары красного и синего цвета (5 и 6 номер), высотой 12м, не имеющие аналогов.

- **template** – шаблон тега (функция в файле `modules/shop/shop.view.php` должна называться `show_block_template`; по умолчанию шаблон `show_block`).

Пример: `<insert name="show_block" module="shop">`

выведет 3 последних товара из магазина

`<insert name="show_block" module="shop" count="5" sort="rand">`

выведет 5 случайных товаров из магазина

`<insert name="show_block" module="shop" sort="price" count="4" cat_id="12">`

выведет 4 самых дешевых товаров из рубрики №12 магазина

Шаблон: функция `show_block` в файле `modules/shop/shop.view.php`.

CSS-форматирование:

`.shop_block{}` – шаблонная функция `show_block`

`.shop_block h2{}` – заголовок

`.shop_block .shop{}` – товар в блоке

`.shop_block .shop_name{}` – название товара

`.shop_block .shop_img{}` – изображение товара

`.shop_block .shop_price{}` – цена товара

`.shop_block .shop_old_price{}` – цена товара без скидки

`.shop_block .shop_discount{}` – скидка товара

`.shop_block .shop_anons{}` – краткое описание товара

`.shop_block .shop_text{}` – полное описание товара

`.shop_block .shop_form{}` – форма покупки товара

`.shop_block .show_all{}` – ссылка на все товары

### **show\_block module="tags"**

Выводит список тегов.

Пример: `<insert name="show_block" module="tags">`  
выведет облако тегов

Шаблон: функция `show_block` в файле `modules/tags/tags.view.php`.

CSS-форматирование:

`.tags_block{}` – список всех тегов

`.tags_block h2{}` – заголовок «Теги»

`.tags_block a{}` – ссылки на теги

`.tags_block span{}` – текущий/активный тег

## **C**

### **show\_calendar module="news"**

Выводит календарь со ссылками на новости за период. Периоды отображаются в виде ссылок на месяцы, только если имеются новости, соответствующие этим периодам. Функция действует только при открытии страницы, к которой прикреплен модуль «Новости».

Атрибуты:

— **month\_hidden** – если необязательный атрибут равен 1, то выводятся только годы без месяцев;

— **template** – шаблон тега (функция в файле `modules/news/news.view.php` должна называться `show_calendar_`**template**; по умолчанию шаблон `show_calendar`).

Пример: `<insert name="show_calendar" module="news">`  
выведет календарь новостей

Шаблон: функция `show_calendar` в файле `modules/news/news.view.php`.

CSS-форматирование:

`.news_calendar{}` – шаблонная функция `show_calendar`

`.news_calendar h2{}` – заголовок

`.news_calendar .news_year{}` – год

`.news_calendar .news_year_current{}` – текущий (открытый) год

`.news_calendar .news_month{}` – месяц

`.news_calendar .news_month_current{}` – текущий (открытый) месяц

`.news_calendar .show_all{}` – ссылка на все новости

### **show\_comments**

Выводит комментарии, прикрепленные к странице сайта, если в конфигурации

модуля «Страницы сайты» подключены комментарии.

Пример: `<insert name="show_comments">`

выведет форму для отправки комментариев и добавленные комментарии посетителей

Шаблон: функция `show_comments` в файле `includes/function.php`.

## D

### `show_description`

Выводит мета-тег `description` страницы.

Пример: `<description><insert name="show_description"></description>`

выведет описание текущей страницы

Шаблон: функция `show_description` в файле `includes/function.php`.

## F

### `show_form module="faq"`

Выводит форму добавления вопроса. Для правильной работы тега должна существовать и быть активной страница, к которой прикреплен модуль Вопрос-Ответ.

Атрибуты:

- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** – категория вопросов (id категории, по умолчанию выбирается одна категория), если в конфигурации модуля отмечено «Использовать категории»;
- **template** – шаблон тега (функция в файле `modules/faq/faq.view.php` должна называться `form_template`; по умолчанию шаблон `form`).

Пример: `<insert name="show_form" module="faq">`

выведет форму добавления вопроса

Шаблон: функция `form` в файле `modules/faq/faq.view.php`.

CSS-форматирование:

`.faq_form{}` – форма «Задать вопрос»

`.faq_form .inptext{}` – поля «Ваше имя», «Ваш вопрос», «E-mail для ответа»

`.faq_form .inpfiler{}` – поле «Прикрепляемые файлы»

`.faq_form .infofield{}` – подпись к полю в форме

`.faq_form .button{}` – кнопка «Отправить» в форме

### `show_form module="feedback"`

Выводит форму добавления сообщения. Для правильной работы тега должна существовать и быть активной страница, к которой прикреплен модуль Обратная связь.

Атрибуты:

- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **template** – шаблон тега (функция в файле *modules/faq/faq.view.php* должна называться *form\_template*; по умолчанию шаблон *form*).

Пример: `<insert name="show_form" module="feedback">`

выведет форму добавления сообщения

Шаблон: функция *form* в файле *modules/feedback/feedback.view.php*.

CSS-форматирование:

`.feedback_form{}` – форма

`.feedback_form .infofiled{}` – подпись к полю в форме

`.feedback_form .inptext{}` – текстовые поля

`.feedback_form .inpnum{}` – цифровые поля

`.feedback_form .inpselect{}` – поля «Список»

`.feedback_form .inpcheckbox{}` – поля «Чекбокс»

`.feedback_form .inpfile{}` – поле «Файл»

`.feedback_form .button{}` – кнопка «Оправить»

`.feedback_form_param_text{}` – описание поля

`.feedback_form_paramNUM{}` – поле с номером NUM

`.feedback_form_paramNUM .infofield{}` – название параметра с номером NUM

`.feedback_form_paramNUM .inptext{}` – текстовое поле для ввода текстового значения параметра с номером NUM

`.feedback_form_paramNUM .inpnum{}` – текстовое поле для ввода цифрового значения параметра с номером NUM

`.feedback_form_paramNUM .inpselect{}` – список для выбора значения параметра с номером NUM

`.feedback_form_paramNUM .inpcheckbox{}` – чекбокс для выбора значения параметра с номером NUM

## Н

### show\_href

Выводит ссылку на страницу сайта. Если текущая страница соответствует адресу, на которую ведет ссылка, то ссылка становится неактивной. Шаблонная функция аналогична стандартной HTML-конструкции `<a href=""></a>`, но ликвидирует ссылки, которые никуда не ведут.

Атрибуты:

- **rewrite** — псевдоссылка страницы, на которую ведет ссылка, например: *news*;
- **img** — адрес изображения, использующегося в качестве ссылки, например: *img/logo.gif*;
- **img\_act** — адрес изображения на текущей странице, например:

- img/home\_act.gif*;
- **width** — ширина изображения;
  - **height** — высота изображения;
  - **alt, alt1, ...** — альтернативный тег для изображения или текст ссылки. Номер необходимо заменить номером языка, если на сайте используется мультиязычность.

Шаблон: функция *show\_href* в файле *includes/function.php*.

## I

### **show\_images**

Выводит изображения, прикрепленные к странице сайта (если в конфигурации модуля «Страницы сайта» включен параметр «Использовать изображения»).

Пример: `<insert name="show_images">`

выведет в ячейке таблицы группу фотографий, прикрепленных к текущей странице

Шаблон: функция *show\_images* в файле *includes/function.php*.

## J

### **show\_js**

Подключает JS файлы, обеспечивающие работу diafan.CMS.

Пример: `<insert name="show_js">`

Шаблон: функция *show\_js* в файле *includes/function.php*.

## K

### **show\_keywords**

Выводит ключевые слова страницы. Используется для мета-тега `keywords`.

Пример: `<keywords><insert name="show_keywords"></keywords>`

выведет ключевые слова текущей страницы

Шаблон: функция *show\_keywords* в файле *includes/function.php*.

## L

### **show\_language**

Выводит ссылки на альтернативные языковые версии сайта (если используется мультиязычность);

Пример: `<insert name="show_language">`

выведет список ссылок на языковые версии сайта, «ru/eng»

Шаблон: функция *show\_language* в файле *includes/function.php*.

CSS-форматирование:

`.language{}`

### **show\_links**

Выводит ссылки на страницы нижнего уровня, принадлежащие текущей странице.

Пример: `<insert name="show_links">`

Шаблон: функция `show_links` в файле `includes/function.php`.

CSS-форматирование:

`.show_links{}`

`.show_links li{}`

### **show\_login module="registration"**

Выводит форму авторизации или приветствие и ссылки на редактирование данных и выход, если пользователь авторизован.

Пример: `<insert name="show_login" module="registration">`

выведет форму авторизации

Шаблон: функция `show_login` в файле `modules/registration/registration.view.php`.

CSS-форматирование:

`.login{}` – шаблонная функция `show_login` – авторизация

`.login .login_avatar{}` – аватар пользователя

`.login .infofield{}` – подпись к полю в форме

`.login .inpbutton{}` – слой с кнопкой «ОК»

`.login .button{}` – кнопка «ОК»

`.login .inptext{}` – текстовые поля для ввода логина и пароля

`.login_reminding{}` – слой с ссылкой «Забыли пароль?»

`.login_registration{}` – слой с ссылкой «Регистрация»

## **P**

### **show\_path**

Выводит навигацию по сайту «Хлебные крошки». Необязательный атрибут `separator` — разделитель ссылок в навигации.

Пример: `<insert name="show_path">`

выведет ссылки навигации по сайту, «Главная – О компании – Услуги»

Шаблон: функция `show_path` в файле `includes/function.php`.

CSS-форматирование:

`.path{}`

## show\_previous\_next

Выводит ссылки на предыдущую и последующую страницы.

Пример: `<insert name="show_previous_next">`

Шаблон: функция `show_previous_next` в файле `includes/function.php`.

CSS-форматирование:

`.previous_next_links`{clear: both; padding-top: 10px; height: 40px;} – ссылки на предыдущую и следующую страницы

`.previous_link`{width: 40%; float: left;} – ссылка на предыдущую страницу

`.next_link`{width: 40%; float: right; text-align: right;} – ссылка на следующую страницу

## show\_protect

Выводит JS код запрещающий копирование контента на сайте.

Пример: `<body<insert name="show_protect">>`

Шаблон: функция `show_protect` в файле `includes/function.php`.

## S

### show\_search module="search"

Выводит форму поиска по сайту. Атрибуты: **button**, **button2**, ... – переводимое значение кнопки «Найти».

Пример: `<insert name="show_search" module="search" button="найти" button2="search">`

выведет форму поиска по сайту

Шаблон: функция `show_search` в файле `modules/search/search.view.php`.

CSS-форматирование:

`.search`{ } – форма поиска

`.search .input_search`{ } – поле для ввода выражения для поиска

`.search .submit_search`{ } – кнопка «Найти»

### show\_search module="shop"

Выводит форму поиска товаров. Если для категорий прикреплены дополнительные характеристики, то поиск по ним производится только на странице категории.

Атрибуты:

- **only\_shop** — выводить форму поиска только на странице магазина;
- **site\_id** – страница, к которой прикреплен модуль, по умолчанию выбирается одна страница;
- **cat\_id** – категория товаров (*id* категории, по умолчанию поиск осуществляется по всем категориям магазина), если в конфигурации модуля отмечено «Использовать категории»; можно указать значение **current**, тогда

поиск будет осуществляться по текущей (открытой) категории магазина или по всем категориям, если ни одна категория не открыта;

— **template** – шаблон тега (функция в файле *modules/shop/shop.view.php* должна называться *show\_search\_template*; по умолчанию шаблон *show\_search*).

Пример: `<insert name="show_search" module="shop">`

выведет форму поиска по каталогу товаров

Шаблон: функция *show\_search* в файле *modules/shop/shop.view.php*.

CSS-форматирование:

`.shop_search{}` – шаблонная функция *show\_search*

`.shop_search .infoform{}` – заголовок группы полей

`.shop_search .infofield{}` – подпись к полю в форме

`.shop_search .inptext{}` – текстовые поля

`.shop_search .inpshort{}` – цифровые поля

`.shop_search .inpcheckbox{}` – поля «Чекбокс»

`.shop_search .button{}` – кнопка «Оправить»

`.shop_search_name{}` – поиск по названию

`.shop_search_name .infofield{}` – заголовок «Название:»

`.shop_search_name .inptext{}` – текстовое поле для ввода названия товара

`.shop_search_description{}` – поиск по описанию

`.shop_search_description .infofield{}` – заголовок «Описание:»

`.shop_search_description .inptext{}` – текстовое поле для ввода описания товара

`.shop_search_price{}` – поиск по цене

`.shop_search_price .infofield{}` – заголовок «Цена:»

`.shop_search_price .inpnum{}` – текстовое поле для ввода цены товара

`.shop_search_paramNUM{}` – поиск по параметру с номером NUM

`.shop_search_paramNUM .infofield{}` – название параметра с номером NUM

`.shop_search_paramNUM .infoform{}` – заголовок – параметр с номером NUM

`.shop_search_paramNUM .inptext{}` – текстовое поле для ввода значения параметра с номером NUM

`.shop_search_paramNUM .inpnum{}` – текстовое поле для ввода цифрового значения параметра с номером NUM

`.shop_search_paramNUM .inpcheckbox{}` – чекбокс для выбора значения параметра с номером NUM

### **show\_social\_links**

Выводит ссылки на социальные сети.

## **T**

## show\_tags

Выводит теги (слова-якори), прикрепленные к странице сайта, если в конфигурации модуля «Страницы сайты» подключены теги.

Пример: `<insert name="show_tags">`

выведет список ссылок на теги, прикрепленные к текущей странице

Шаблон: функция `show_tags` в файле `includes/function.php`.

## show\_title

Выводит заголовок. Используется между тегами `<title></title>` в шапке сайта.

Пример: `<title><insert name="show_title"></title>`

выведет заголовок текущей страницы

Шаблон: функция `show_title` в файле `includes/function.php`.

## U

### show\_users

Выводит статистику пользователей на сайте.

Пример: `<insert name="show_users">`

выведет «Сейчас на сайте: \*\* гостей, \*\* пользователей.»

Шаблон: функция `show_users` в файле `includes/function.php`.

CSS-форматирование:

```
.show_users{}
```

## V

### show\_votes

Выводит голосования. Если не задан номер вопроса, то выводится один или все вопросы (в зависимости от конфигурацией модуля), соответствующие открытой странице, в случайном порядке.

Атрибуты:

— **id** – задает номер вопроса.

Пример: `<insert name="show_votes">`

выведет вопрос, варианты ответа и кнопки «Ответить» и «Результаты», согласно настройкам и заполнению модуля «Опросы» в административной части сайта

Шаблон: функция `show_votes` в файле `includes/function.php`.

## Y

### show\_year

Выводит период функционирования сайта в годах. Начало отсчета от атрибута **year** (по умолчанию текущий год).

Пример: `<insert name="show_year">`

в 2009 году выведет «2009 г.», в 2010 году выведет «2010 г.»

---

`<insert name="show_year" year="2010">`

в 2010 году выведет «2010 г.», в 2011 году выведет «2010-2011 гг.»

Шаблон: функция `show_year` в файле `includes/function.php`.